NeuroImage

www.elsevier.com/locate/ynimg
NeuroImage 36 (2007) 346–360

Dendritic spine detection using curvilinear structure detector and LDA classifier

Yong Zhang,[a,b] Xiaobo Zhou,[a,c,*] Rochelle M. Witt,[d] Bernardo L. Sabatini,[d] Donald Adjeroh,[b] and Stephen T.C. Wong[a,c]

[a]*Center for Bioinformatics, Harvard Center for Neurodegeneration and Repair, Harvard Medical School, Boston, MA 02215, USA*
[b]*Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506, USA*
[c]*Functional and Molecular Imaging Center, Department of Radiology, Brigham and Women's Hospital and Harvard Medical School, Boston, MA 02115, USA*
[d]*Department of Neurobiology, Harvard Medical School, 220 Longwood Avenue, Boston, MA 02115, USA*

Received 25 October 2006; revised 24 February 2007; accepted 27 February 2007
Available online 13 March 2007

Dendritic spines are small, bulbous cellular compartments that carry synapses. Biologists have been studying the biochemical pathways by examining the morphological and statistical changes of the dendritic spines at the intracellular level. In this paper a novel approach is presented for automated detection of dendritic spines in neuron images. The dendritic spines are recognized as small objects of variable shape attached or detached to multiple dendritic backbones in the 2D projection of the image stack along the optical direction. We extend the curvilinear structure detector to extract the boundaries as well as the centerlines for the dendritic backbones and spines. We further build a classifier using Linear Discriminate Analysis (LDA) to classify the attached spines into valid and invalid types to improve the accuracy of the spine detection. We evaluate the proposed approach by comparing with the manual results in terms of backbone length, spine number, spine length, and spine density.

Keywords: Neuron image processing; Dendritic spine; Curvilinear structures; Neurite outgrowth; Fluorescence microscopy

Introduction

Dendritic spines are defined as tiny membranous compartments consisting of a head (volume ~0.01–1 μm^3) connected to the parent dendrite by a thin (diameter ~0.1 μm) spine neck (Harris, 1999a). The morphological and statistical changes of small cellular compartments, such as the dendritic spines, are of great interest when studying neurons (Tavazoie et al., 2005; Sarbassov et al., 2005; Carter and Sabatini, 2004; Zhou et al., 2006). As an example, Tavazoie et al. (2005) and Zhou et al. (2006) show that the TSC pathway regulates soma size, the density and size of dendritic spines, and the properties of excitatory synapses in hippocampal pyramidal neurons. Spine Ca^{2+} has a crucial role in the induction of most forms of synaptic long-term potentiation (LTP) and long-term depression (LTD)—the putative cellular mechanisms of learning and memory (Sabatini et al., 2001). The inability to examine and measure fluorescent signals from individual spines or synapses has been a major obstacle for such study. Recently the 2-photon laser scanning microscopy (2PLSM) (Lendvai et al., 2000; Maletic-Savatic et al., 1999; Engert and Bonhoeffer, 1999) has been widely used to image very small subcellular compartments within brain slices or in vivo, making it possible to study biochemical signaling that occurs at the intracellular level.

Despite previous work on automating the detection of dendritic backbones and spines (Al-Kofahi et al., 2002; Herzog et al., 1997; Koh et al., 2002; Weaver et al., 2004), the problem still remains largely unsolved, partially due to the irregular and diverse shapes of the objects. Herzog et al. (1997) proposed a method for 3D reconstruction of dendritic trees using a parametric model of cylinders. This method, however, missed most of the short spines or spines with thin neck. Al-Kofahi et al. (2002) use a generalized cylinder model to estimate the local directions and the dendritic backbones are directly traced from a set of pre-estimated seed points. The extraction of backbones is not complete due to the incomplete estimation of the seed points. Koh et al. (2002) construct a modified medial axis to locate dendritic backbones and detect spines as geometric surface protrusions relative to the backbones. Since the segmentation is achieved by a simple global threshold method, and very limited geometric information is considered for spine detection, this method detects a lot of pseudo spines. Considering other 3D line extraction and axes generation algorithms, either they are not able to provide accurate, smooth, and complete segmentation of the objects, or they are not well-suited to our specific problem of accurately segmenting the spine

* Corresponding author. Center for Bioinformatics, Harvard Center for Neurodegeneration and Repair, Harvard Medical School, Boston, MA 02215, USA.
E-mail address: xiaobo_zhou@hms.harvard.edu (X. Zhou).
Available online on ScienceDirect (www.sciencedirect.com).

1053-8119/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.neuroimage.2007.02.044

and backbone structures. As an example, the airway finding methods have been studied for MRI/CT images in the application of virtual bronchoscopic and dynamic navigation (Sauret et al., 1999; Kiraly et al., 2002, 2004; Swift et al., 2002). These methods are mainly focusing on the definition of 3D tree structures and the modeling, visualization, and reconstruction of the 3D tubular objects. Swift et al. (2002) proposed a method for automatic axis generation using Discrete Model and Generalized Cylinder Model. Their method for segmenting airways is basically threshold based contour boundary finding. Sauret et al. (1999) proposed a semi-automated method to obtain full 3D topology and dimensions of branching networks in CT images. Their method is based on 3D skeletonization, which may give wrong segmentation results around image boundaries and need extra processing to remove side branches and smooth medial axis.

In this paper, we propose a novel method for automated segmentation and quantification of dendritic backbones and spines suitable for 3D neuron images in thin slices. Usually neurobiologists are interested in spine length, spine size, and spine density (defined as number of spines per backbone length). Since the focus of this paper is to propose a general image analysis methodology to extract different morphological features, we do not focus on any one particular feature, but the spine size, spine root, spine boundaries, spine and backbone length, number of spines, and spine density. To quantify those spine features, our method is able to accurately segment each individual spine as well as dendritic backbones. Then the biologists can make decisions based on that. Fig. 1 shows a sample (original and denoised) neuron image with dimension 512*512*11. The 3D image stack is projected onto the *xy*, *yz*, and *zx* planes, respectively. Since the slices along the optical direction (*z*) provide very limited information, it is desired to consider only the 2D projection onto the *xy* plane. The 2D image used for analysis is a maximum intensity projection of the original 3D stack. It is obtained by projecting in the *xy* plane the voxels with maximum intensity values that fall in the way of parallel rays traced from the viewpoint to the plane of projection. Our method treats the dendritic backbones and spines as curvilinear structures and extracts their centerlines and boundaries by estimating the second-order directional derivatives. A classifier is built from a pre-selected training set and the detected spines are further classified to remove pseudo spines. Once the image stack is processed on its 2D projection, the very small number of missing spines can be detected by simply looking through the image stack. This way, the process is significantly simplified and very fast. Although the volume information about spines is missing, most of the geometric and statistical information is reserved and adequate for further biological analysis.

Among the photosensitive devices in use today, the photomultiplier tube (PMT) is a versatile device that provides extremely high sensitivity and ultra-fast response. However, the electronic signal converted by PMT detectors is usually noisy. Thus, image denoising is a necessary step before any further processing. Some methods for image denoising and restoration are discussed in Kempen et al. (1996). In our case, the noise is mainly "salt and pepper" type. So we apply 2D median filtering (Lim, 1989) to remove the noise and preserve the boundary information for dendrites and spines. We use the original noisy image and the denoised image to estimate the signal to noise ratio (SNR) and the peak SNR (PSNR). A typical image has a SNR value of 13 dB and PSNR of 46 dB. We treat the dendritic backbones and spines as curvilinear structures and extract their centerlines using the curvilinear structure detector (Steger, 1998). The backbones and spines can be easily distinguished by examining the length of the detected centerlines, based on the observation that the length of a spine is much smaller than that of a backbone. The boundaries of the backbones and spines are extracted using information provided by the curvilinear structure detector for a fast process. Once the spines are detected, they are grouped as either attached or detached based on the distance between the spine root and the backbone. Since imaging artifacts and noise may cause irregular shapes in the image, there is no doubt that not all the small objects extended from backbones are spines (see examples in Fig. 8). In this paper, we use "*protrusion*" to indicate those that are not spines. In our proposed method, the user (usually an expert in the field of neurobiology) can pick spines and protrusions to train the classifier such that the protrusions can be removed from the detected objects to improve accuracy. In the final stage, the dendritic backbones and spines are quantified in terms of the spine number, backbone and

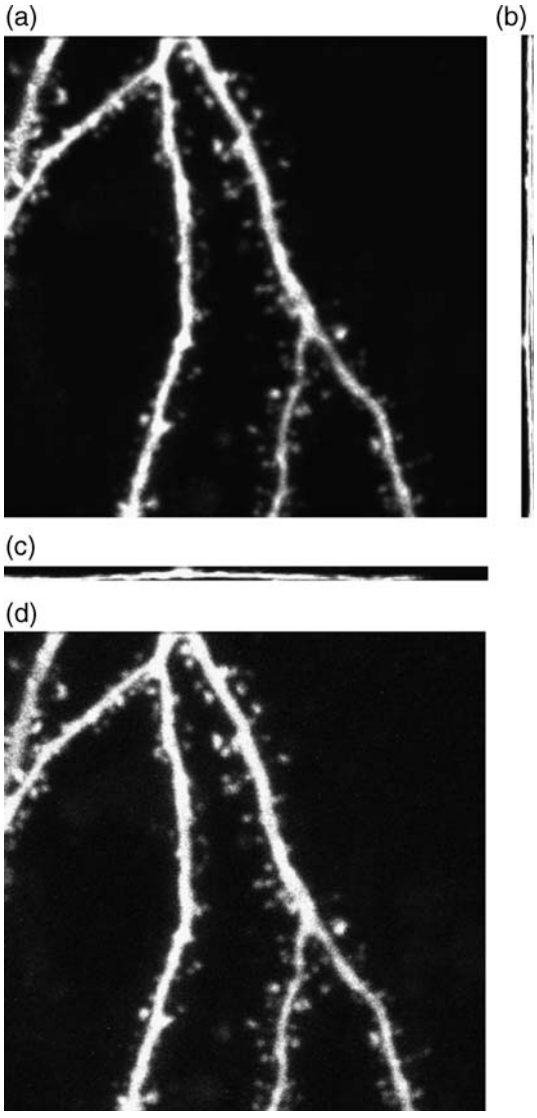

Fig. 1. A sample neuron image stack projected along (a) *z*-direction; (b) *x*-direction; and (c) *y*-direction. (d) Original noisy image projected along *z*-direction.

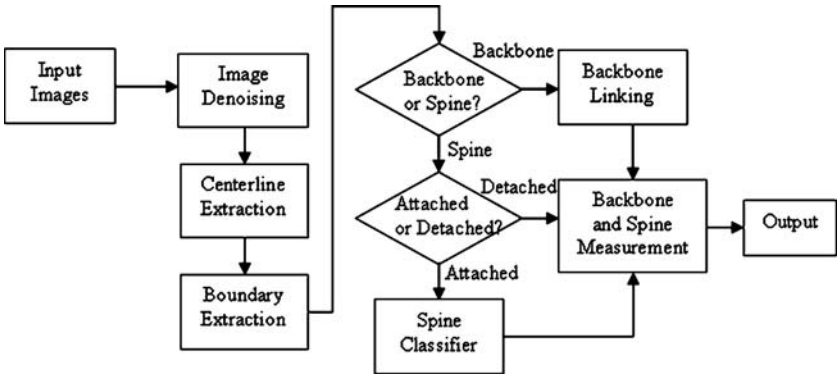

Fig. 2. The block diagram for the automated backbone and spine detection.

spine length, and the spine density. Fig. 2 shows the block diagram for our proposed approach.

The paper is organized as follows. The Materials and methods section presents the image acquisition method, and methods for dendritic backbone extraction and spine detection and classification. The experimental results and validation are presented in the Experimental results and validation section. We conclude our work in the Conclusion and discussion section and necessary future work is discussed.

Materials and methods

Neuron image acquisition

The test images are provided by Bernardo L. Sabatini's Laboratory, one of our collaborators at the Department of Neurobiology, Harvard Medical School. The neuron images are acquired with two-photon laser scanning microscopes with an excitation wavelength of 910 nm. The pyramidal neurons are transfected with the Green Fluorescent Protein (GFP) and are imaged at 0.8× zoom (image field, 300×270 μm), whereas spiny regions of basal and apical dendrites are imaged at 5× magnification (image field, 42×42 μm). Optical sections are taken at 1.0 μm spacing. In this paper, we use the magnified images of spiny regions of basal and apical dendrites, a portion of the pyramidal neuron, as input images for testing our method. We use "neuron image" to represent those magnified images in the following sections.

Automated extraction of dendritic backbones

Centerline extraction for dendritic backbones

A backbone is defined as the centerline of a dendrite segment. All the centerlines of the dendrite segments, if connected, form the dendritic backbone structure. The detection of the dendritic backbone involves extracting and connecting the centerlines and boundaries for all the dendrite segments. In this work, we apply the curvilinear structure detector (Steger, 1998; Xiong et al., 2006) to extract the centerlines of the dendritic backbone structures. Curvilinear structure detector is a robust line extraction method that is able to rapidly, accurately, and completely extract centerlines of line structures in 2D images. Basically, this method examines each pixel in the image and detects the line pixels that are located on or close to the line structures. These line pixels are then linked to form the centerlines of the line structures. A 2D line can be modeled as a curve that exhibits the characteristic 1D line profile in the direction perpendicular to the line, as shown in Fig. 3. In our algorithm, there are five steps to extract the dendritic backbones:

- Step 1: local line direction estimation;
- Step 2: line point detection;
- Step 3: line point linking;
- Step 4: boundary extraction;
- Step 5: backbone linking.

The local direction of the line is determined using the second-order directional derivatives of the image. However, since an image contains noise, the calculation of its derivatives is not straightforward. It is appropriate to estimate the derivatives of an image by convoluting it with the derivatives of the Gaussian kernel (Florack et al., 1992). Thus, the derivatives of a noisy image I can be estimated as follows:

$$r' = (I*G)' = I*G'$$
$$r'' = (I*G)'' = I*G'' \quad (1)$$

where "G" represents the Gaussian kernel and "*" is the convolution operator. The estimation of the derivatives for a 2D image can be achieved by convolving the image with the derivatives of the Gaussian kernel along one dimension and then convolving along the other dimension. After convoluting with the Gaussian kernel, the criterion for a bright salient line on a dark background is that the second-order derivatives (r'') take minimum value (or maximum negative value) at locations where the first-order derivatives (r') are equal to zero.

Since Gaussian kernel acts as a low-pass filter, it will blur the boundary information, or even smooth away weak line structures.

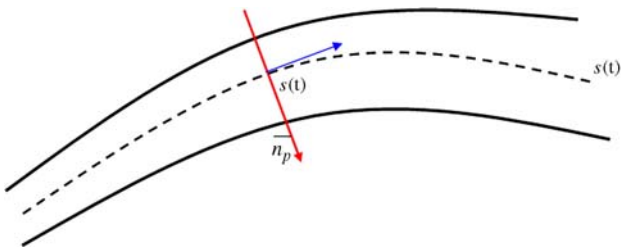

Fig. 3. The 2D line model represented by a 1D line profile along the direction n_p perpendicular to the local line direction $s(t)$.

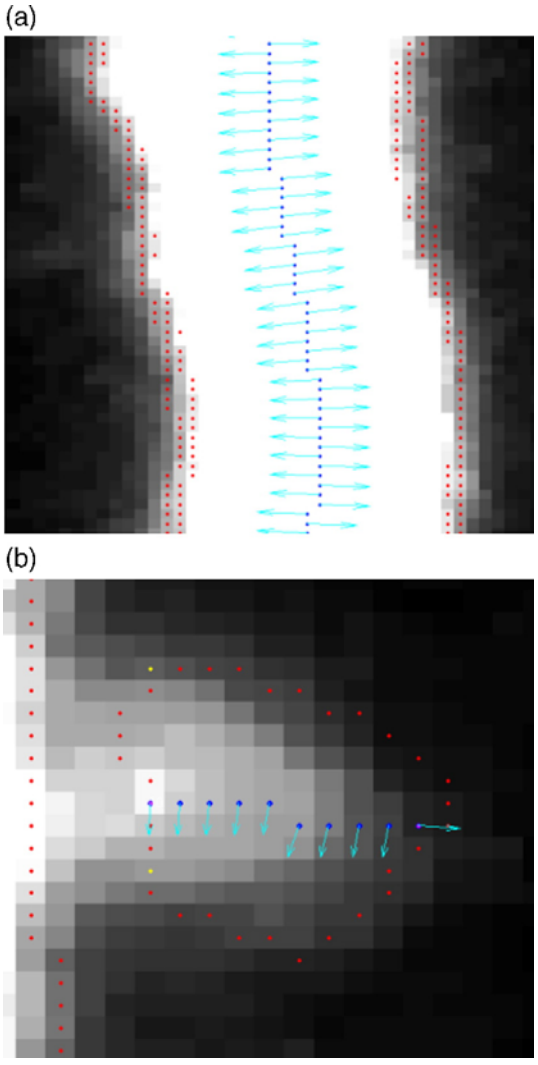

Fig. 4. Example results for boundary direction. The vectors in cyan color are one of the two directions perpendicular to the local line direction. The blue points are the detected center line points of the curvilinear structures. The red points are detected boundary points by searching along the vector directions and the opposite directions. Some boundary points are interpolated from the neighboring points. (a) Backbone boundary detection results; (b) spine boundary detection results.

Thus, the selection of the standard deviation (σ) for the Gaussian kernel is restricted by two conditions:

(1) σ should be large enough so that r'' exhibits a clearly defined minimum at $r'=0$;

(2) σ should not be too large to avoid smoothing away most of the line structures.

In order to preserve most of the line structures (including weak lines) during the convolution, we adopt the method in Xiong et al. (2006) to select multiple σ values for different pixels in a neuronal image. Simply, we consider the Hessian matrix for each pixel and calculate different set of eigenvalues when using different σ values for the convolution. We pick the value such that its eigenvalue has the maximum absolute value among all the eigenvalues.

The eigenvector corresponding to the maximum absolute eigenvalue is the normalized direction perpendicular to the line. Denote this normalized direction as $\vec{n}_p$ where subscript p represents the current pixel location. We call the second-order derivative the *strength value* since it indicates how strong the line is. A noticeable line pixel must have a second-order derivative that is larger than a threshold value, which is called *strength threshold*. $\vec{n}_p$ can also be represented as $(\vec{n}_x, \vec{n}_y)$ a unit vector along x and y directions. Once the local line direction is determined, the location of the candidate line point (x, y) is given by:

$$(x,y) = (s\vec{n}_x, s\vec{n}_y) \tag{2}$$

where s is calculated using the local line direction vector (Xiong et al., 2006).

After the line points are detected, they are linked to form the centerline. The linking starts from the line points whose second directional derivatives are larger than the user-selectable threshold value, called *upper threshold*. It ends at the line points whose second directional derivatives are lower than another user-selectable threshold value, called *lower threshold*. When adding an appropriate neighbor to the current line, only three neighboring pixels that are compatible with the current line direction are examined. The appropriate neighbor is given by:

$$p^{i+1} = \arg\min_{k=1,2,3} \left\{ \|p^i - p_k^{i+1}\|_2 + |\alpha^i - \alpha_k^{i+1}| \right\} \tag{3}$$

where p^i is the current line point and p^{i+1} is the point being added to the line, p_k^{i+1} ($k=1, 2, 3$) are the three neighborhoods, α is the angle value given by: $(\vec{n}_x, \vec{n}_y) = (\cos\alpha, \sin\alpha)$.

Boundary extraction for dendritic backbones

In Steger (1998) the boundaries of the curvilinear structures are extracted using a facet model line detector. In this paper we proposed a new method to accelerate the process. We consider only the bright lines on a dark background. The detection of the boundaries for such lines is based on the observation that the *strength values* of the line pixels change abruptly at the boundary locations. We perform boundary detection in three steps. In the first step, we examine the strength value for each pixel in the

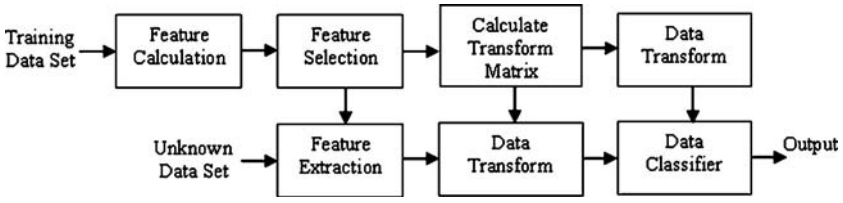

Fig. 5. The block diagram for the spine classification using LDA.

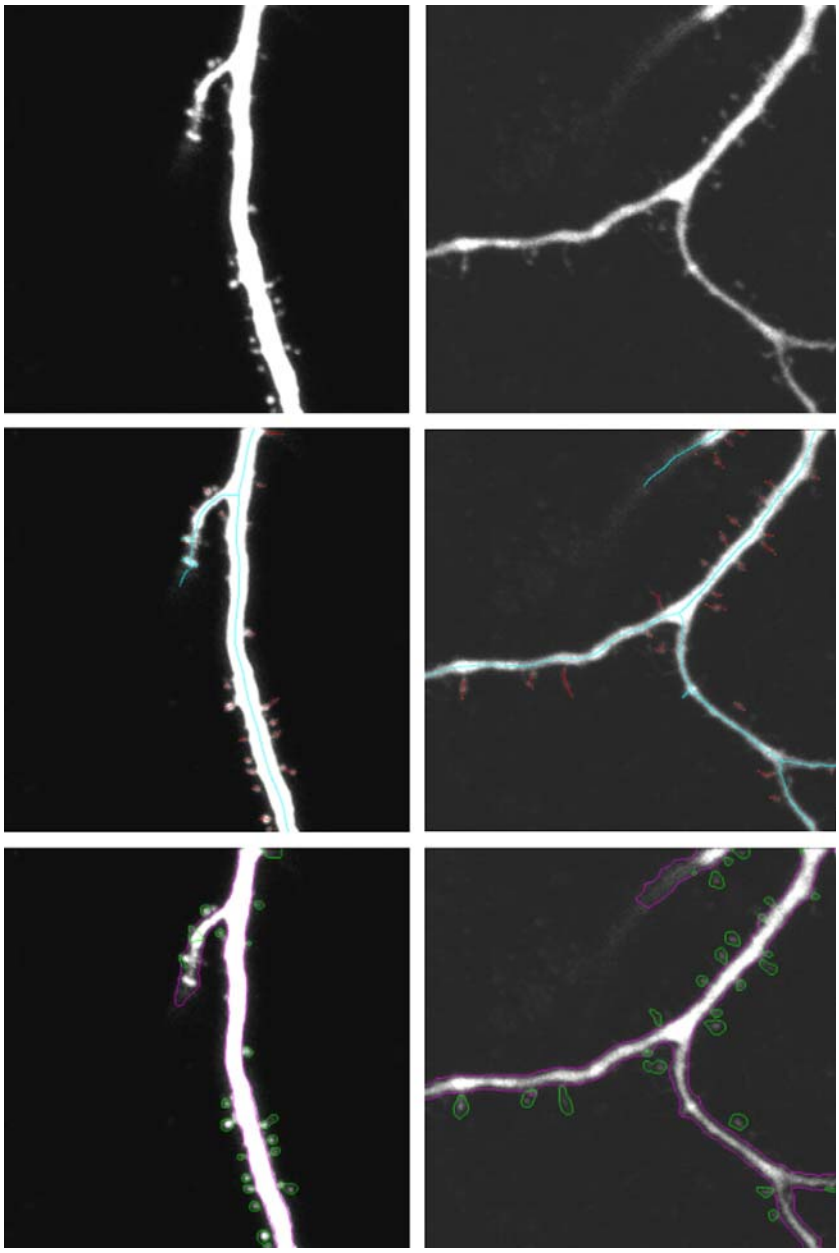

Fig. 6. Results for dendritic backbone and spine extraction. The images on the first row are the original neuron images, images on the second row are the results with the extracted centerline labeled with color (cyan for backbone centerline and red for spine centerlines), and images on the third row are the result with the extracted boundary labels with color (magenta for backbone boundaries and green for spine boundaries).

image and set it to zero if it is smaller than a predefined *strength threshold*. That is,

$$\mathrm{SV}(p) = \begin{cases} 0 & \text{if } S_p < \mathrm{ST} \\ \mathrm{SV}(p) & \text{otherwise} \end{cases} \tag{4}$$

where SV(p) is the strength value for the pixel at location p and ST is the strength threshold value. The remaining pixels with non-zero strength values are then the line pixels noticeable in the foreground. We call it the *strength image* if each pixel is represented by its new strength value.

In the second step, starting from the first centerline point, we search the strength image along the two directions perpendicular to the local line direction until the strength value drops to zero. The stopping positions are then extracted as boundary points. The searching is applied to all the centerline points on each detected backbones. Each centerline pixel will have two boundary points associated with it. Let d_p represent the local line direction for the current pixel p. The two perpendicular directions, if rotated to the left and to the right from d_p, are denoted as d_p^L, d_p^R respectively. The points on the two searching directions are $P_p = \{p_1, p_2, K, p_K\}$ and $Q_p = \{q_1, q_2, K, q_K\}$, where K is the maximum possible line width. Then the boundary points b_p^L and b_p^R for p are:

$$b_p^L = p_i \ , \ \text{if } p_i \in P_p \text{ and } \mathrm{SV}(p_{i+1}) = 0$$

$$b_p^R = q_j \ , \ \text{if } q_j \in Q_p \text{ and } \mathrm{SV}(q_{j+1}) = 0 \tag{5}$$

If the centerline points of a line structure is represented as $C=\{c_1, c_2, K, c_N\}$, where N is the total number of the points, c_1 is the starting point, and c_N is the ending point of the line. For each point c_i, there will be two boundary points found by formula (5), then we find a set of boundary points for C as $\{(b_1^L, b_1^R),(b_2^L, b_2^R, K, (b_N^L, b_N^R)\}$. The search is very fast since it only involves finding the first zero-value point in the small sets of points P_p, Q_p.

Since only the boundary points on the two searching paths d_p^L, d_p^R are found, other boundary points that are not on the paths may be missing. In the third step, the missing boundary points are detected from the neighboring boundary points. Given two known boundary points, if they are adjacent, i.e., there are no any other boundary points between them, then the Bresenham (1965) algorithm is applied to find the points that can link them into a discrete line, with the two known boundary points as the starting and the ending points. These new points on the discrete line are regarded as the missing boundary points. That is, the complete boundary points for the centerline C are:

$$B = \left\{(b_1^L, b_1^R), l_{1,2}^L, l_{1,2}^R, (b_2^L, b_2^R), l_{2,3}^L, l_{2,3}^R, K, (b_N^L, b_N^R)\right\} \quad (6)$$

where $l_{i,j}^L$ is the set of points on the discrete line that links point b_i^L and b_j^L, $l_{i,j}^R$ for linking b_i^R and b_j^R, Fig. 4(a) shows an example results for the backbone boundary detection. We use different

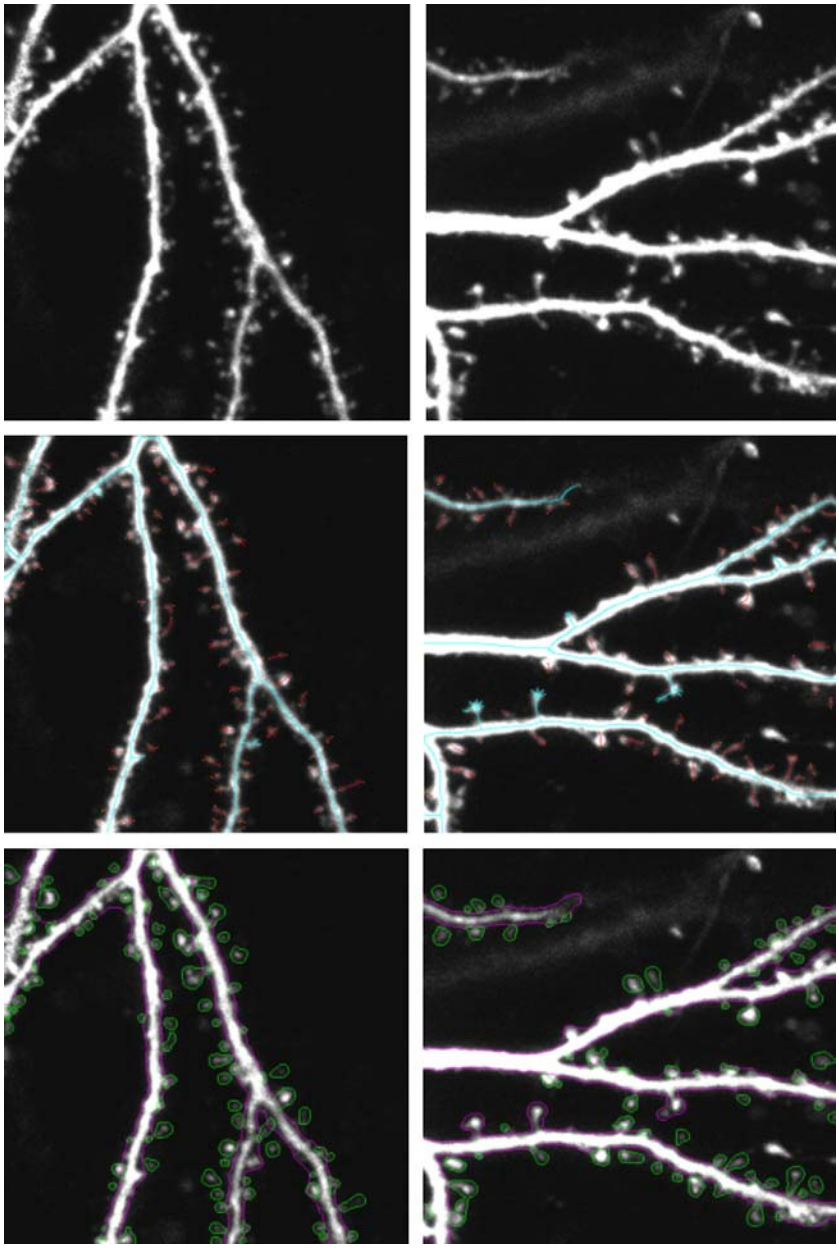

Fig. 7. Results for dendritic backbone and spine extraction. The images on the first row are the original neuron images, images on the second row are the results with extracted centerlines labeled with color (cyan for backbone centerline and red for spine centerlines), and images on the thrid row are the result with extracted boundary labels with color (magenta for backbone boundaries and green for spine boundaries).

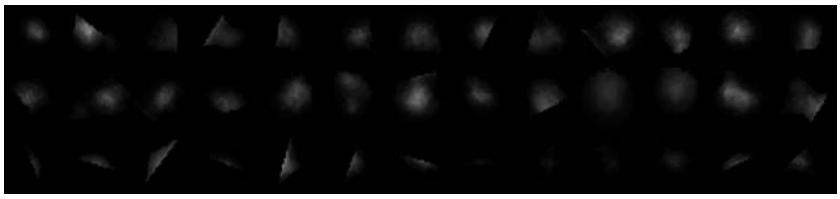

Fig. 8. Partial spine samples in the training set for spine qualifier. The first two rows are the samples of valid spine type; the last row is the samples of valid spine type; the last row is the samples of invalid spine type.

colors to represent the centerline points, boundary points, and the searching vectors.

The advantages of our proposed method for boundary detection are obvious: first, our proposed method is fast. The linear-time searching of the boundary points is based on the second-order derivatives and the local line directions obtained when extracting the centerlines, thus no further convolution and other computations are required. Second, the detected results are very accurate because of the use of the second-order derivatives, local line directions, and the removal of weak line pixels. Third, since each centerline pixel has two boundary points associated with it, this can be used to further detect the spine heads and roots, and also can be very useful for linking multiple backbones.

Multiple backbones linking

If more than one backbone are detected, they need to be linked together to construct the dendritic structures (see Fig. 6 as an example). The backbones are linked based on their boundary information. If the boundary points for two backbones are overlapped or located within eight-neighborhood, they are considered belonging to the same dendritic structure and then the Bresenham (1965) algorithm is used to directly link the two corresponding centerline points together. Otherwise the two backbones are said to belong to different dendritic structures and no linking is applied. If we denote one backbone C^U and its boundary B^U as two sets of points: $C^U=\{c_1^U, c_2^U, Kc_M^U\}$, $B^U=\{b_1^U, b_2^U, Kb_X^U\}$, another backbone and its boundary as $C^V=\{c_1^V, c_2^V, Kc_N^V\}$, $B^V=\{b_1^V, b_2^V, Kb_Y^V\}$, then the linking is applied if any one of the following four conditions is satisfied:

$$\begin{array}{l}(1)\ \text{if } b_1^U \text{ or } b_X^U = b_1^V \text{ or } b_Y^V \\ (2)\ \text{if } b_1^U \text{ or } b_X^U \in N_8(b_1^V, b_Y^V) \\ (3)\ \text{if } b_1^U \text{ or } b_X^U = b_i^V, b_i^V \in B^V \\ (4)\ b_1^U \text{ or } b_X^U \in N_8(b_i^V),\ b_i^V \in B^V\end{array} \quad (7)$$

where $N_8(p)$ represents all the eight-neighborhood points of the pixel p. We link the two centerline points whose boundary points satisfy formula (7).

Automated spine detection and classification

Detached spine detection

Dendritic spines are tiny membranous compartments consisting of a head (volume ~ 0.01–1 μm^3) connected to the parent dendrite by a thin (diameter ~ 0.1 μm) spine neck (Harris, 1999b). Once dendritic spines are detected, further analysis can be applied on morphological and statistical changes to study

Table 1
SFFS feature selection performance for different number of features

Feature number	8	10	12	14	16
Error rate	2.27%	1.36%	1.36%	1.82%	2.27%

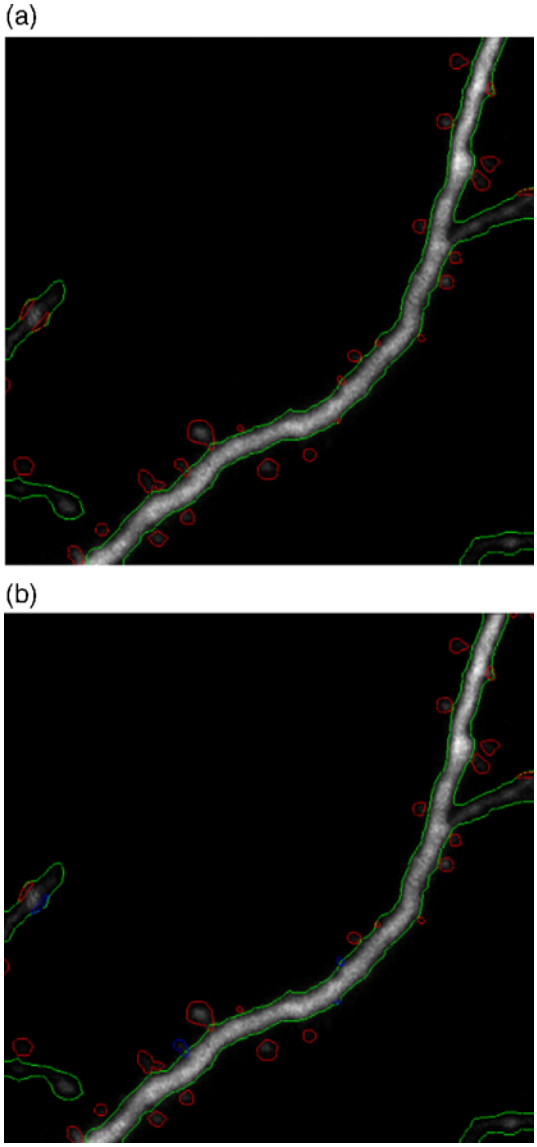

Fig. 9. Spine detection results (a) without spine classification; (b) with spine classification. The dendrite boundaries are labeled in green color; The spines are circled in red color. The pseudo attached spines detected by the classifier are labeled in blue color.

their activity patterns. Our algorithm extracts dendritic spines in four steps:

- Step 1: spine centerline extraction;
- Step 2: spine boundary extraction;
- Step 3: spine root point detection;
- Step 4: spine classification.

The dendritic spines are treated as small protrusions of variable shape attached or detached to multiple dendritic backbones. We use the same method as for dendritic backbone extraction to extract the centerlines of dendritic spines. That is, the line points are detected and linked to form the centerlines for all the spine structures. The major difference, if compared with backbone extraction, is that the length of the centerline for a spine is much less than that for a backbone. Furthermore, we keep only those extracted spines if they are close enough to the nearby backbones. Otherwise, the detected objects are considered as noise or imaging artifacts. For spine boundary extraction, the method is exactly the same as that for detecting backbone boundary. However, we expect cyclic boundaries for spines. Thus, the boundary points are: $\{(b_1^L, b_1^R), l_{1,2}^L, l_{1,2}^R, (b_2^L, b_2^R), l_{2,3}^L, l_{2,3}^R, K, (b_N^L, b_N^R), l_{1,1}^{L,R}, l_{N,N}^{L,R}\}$, where $l_{1,1}^{L,R}$, $l_{N,N}^{L,R}$ means the two linking lines between b_1^L and b_1^R, b_N^L and b_N^R respectively. Fig. 4 (b) shows an example result for spine boundary detection.

There are two types of spines: *attached spine*, which is attached to a backbone surface, and *detached spine*, which is detached but very close to a backbone surface. The method to distinguish them is as follows: the attached spines are those whose minimum distance to a backbone is no larger than the maximum width of that backbone. Otherwise they are detached spines. For attached spines, the edge of the spine intersects that of the backbone. We define the two intersecting points as the *root points* for that spine. The root points are very useful for calculating the spine features, such as length and width of the spine. Once the centerline and boundary of a spine are obtained, the root point detection is straightforward: we only need to consider the point on the centerline that is the closest to the backbone. The two corresponding edge points are then detected as the root points for that spine. Figs. 6 and 7 show some example results for dendritic spine extraction. The centerline, boundary, and root points for each spine are labeled in colors.

Attached spine detection

The above described method is good enough for detecting detached spines. However, it usually detects some pseudo attached

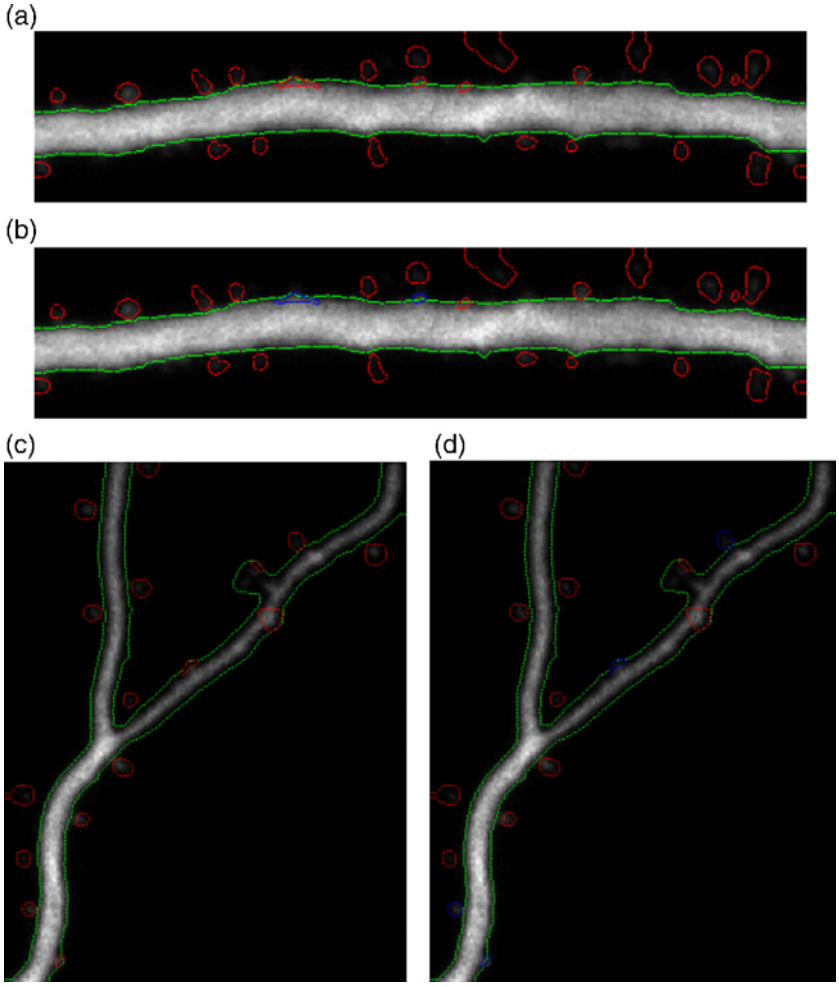

Fig. 10. Spine detection results (a) without spine classification; (b) with spine classification; (c) without spine classification; (d) with spine classification. The dendrite boundaries are labeled in green color. The spines are circled in red color. The pseudo attached spines detected by the classifier are labeled in blue color.

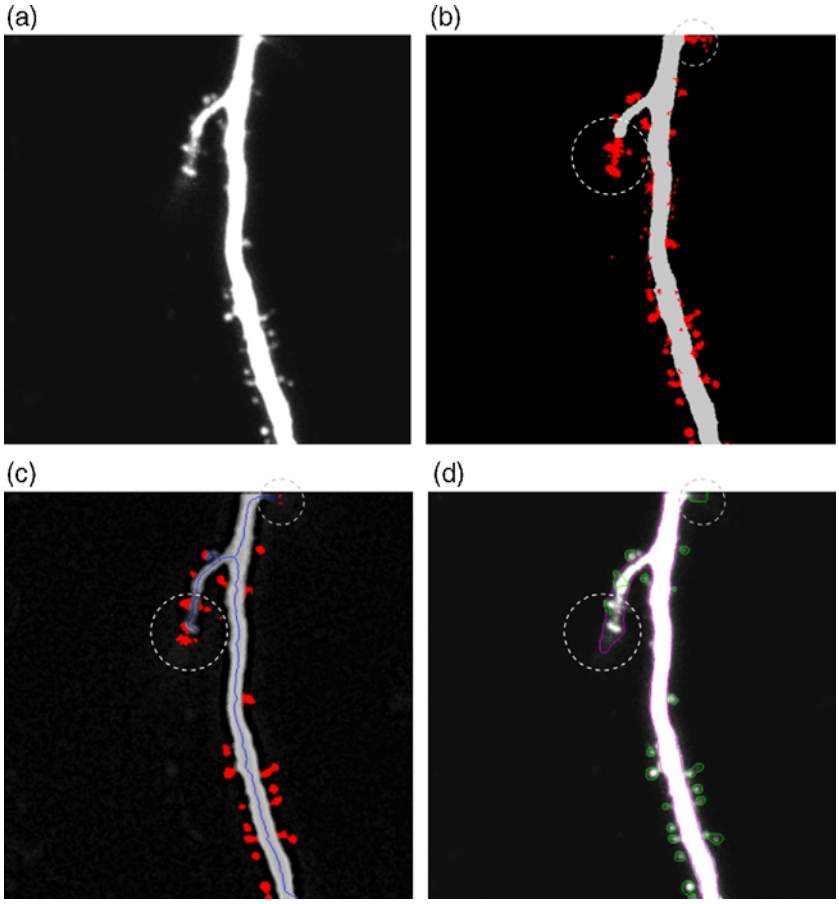

Fig. 11. Results comparison among our algorithm and the threshold based algorithms (see Koh et al., 2002; Weaver et al., 2004). (a) Original image; (b) extraction results using median axis and global threshold method, the detected spines are labeled by red color); (c) extraction results using adaptive method, the detected spines and backbones are labeled in color; (d) extraction results using our method. Note the major difference of the results at circled areas. The threshold based methods tend to detect lots of pseudo spines.

spines due to irregular shapes or noise around the boundary of the backbone areas. Here, we propose an automated attached spine selection procedure using feature selection and spine classification. The procedure is as follows: once the candidate objects are extracted and grouped as attached and detached, we manually pick up a set of attached objects and categorize them as spines or protrusions. These categorized objects will be used as a training set for feature selection and spine classification. We calculate the geometric and moment features for each sample in the training set. We then select and normalize a subset of the features for further spine classification. Finally a classifier is applied to candidate objects to determine if they are spines or protrusions. We explain each step in the following subsections.

Spine feature calculation. Automated identification of the spine type relies on feature calculation, the most critical step in pattern recognition problems. Efficient features could improve the performance and reduce the complexity of the entire system. Considering the geometric and appearance properties of attached spines and pseudo spines, we use two types of features: Zernike moments and geometric features. The feature lengths are 36 for Zernike moments and 12 for geometric features. If a total of m detected objects are manually selected as spine type (s_i, i=1, 2, K, m) and a total of n detected objects as protrusion type (t_j, j=1, 2, K, n), then the training set (T) can be represented as $T=\{s_1, s_2, K, s_m, t_1, t_2, K, t_n\}$. For each sample S_i or T_j in the training set, the calculated features are:

$$\mathbf{F} = [F_1, F_2, L, F_{m+n}]^T, \quad \text{and}$$
$$F_i = [z_{i1}, z_{i2}, L, z_{i36}, g_{i1}, g_{i2}, L, g_{i12}], I = 1 : m + nx \tag{8}$$

where z_{ij},i=1:$m+n$, j=1:36 represents the Zernike moment features for the ith sample and g_{ij},i=1:$m+n$,j=1:12 represents the geometric features for the ith sample. Thus, there are a total of $48\times(m+n)$ features calculated for the training set. We briefly introduce these features below.

Zernike moments are classical image features that have wide applications (Khotanzad and Hong, 1990; Teh and Chin et al., 1988). First, the center of mass is calculated for each spine polygon image and the spine pixels are redefined based on this center. Second, the radius is calculated for each spine, and the average of the radii is defined as R. Third, the pixel (x, y) of the spine image is mapped to a unit circle to obtain the projected pixel (x',y'). Let $I(x', y')$ denote the florescence of the pixel. The Zerike moment for each cell is defined as:

$$Z_{nl} = \frac{n+1}{\pi} \sum_x \sum_y V_{nl}^*(x', y') \cdot I(x', y') \tag{9}$$

where $x'^2+y'^2 \le 1$, $0 \le l \le n$, n-l is even, and $V^*_{nl}(x', y')$ can be calculated as:

$$V^*_{nl}(x',y') = \sum_{m=0}^{(n-l)/2} (-1)^m \frac{(n-m)!}{m!\left[\frac{n-2m+l}{2}\right]\left[\frac{n-2m-l}{2}\right]} \times (x'^2+y'^2)^{(n-2m)/2} e^{il\theta} \quad (10)$$

where $\theta = \arctan y/x$ and $i = \sqrt{-1}$. We select $n=10$ and use $|Z_{nl}|$ as the feature to obtain 36 moments features in total.

We also use a set of common region properties to describe the shape and texture characteristics (*geometric features*) of the spines in our work. For general texture description, we use the maximum, minimum, mean, and standard deviation of the intensity values in the segmented spine area. We also use some weak shape descriptions, such as the length of the longest axis $l_{\max}$ and the length of the shortest axis $l_{\min}$, the ratio $l_{\max}/l_{\min}$, the spine area s, the spine perimeter p, and the spine compactness which is calculated as: compactness $= p^2/(4\pi s)$. If the perimeter of minimum convex shape is p_c, then the spine roughness is: roughness $= p/p_c$. Finally, we have twelve general texture and shape features for each spine region.

Spine feature selection. The high-dimensional features bring two disadvantages in statistical analysis. First, the high-dimensional data with relatively sparse samples make it difficult and inaccurate to parameterize the statistical model when training the classifiers. Second, the high-dimensional data require more computation cost, including time and memory space. Therefore, dimensionality reduction is a common way to condense the features. Feature-subset selection (Cantu-Paz, 2002) and transformation based feature reduction are the two methods for reducing dimensionality of the feature space.

An optional choice is to apply the random search technique to derive an optimal feature subset. A genetic algorithm (GA) is a classical random optimization method, which mimics the evolutionary process of survival of the fittest (Holland, 1996). Another popular method for feature selection is the *Sequential Floating Forward Selection* (SFFS) procedure (Pudil et al., 1994). This method starts with a single feature subset and iteratively adds or removes features until the performance is deteriorated. During the sequential search, once a forward step is applied, a number of backward steps are dynamically executed, given the condition that the performance can be improved. Even thought very simple, the SFFS algorithm has been shown to perform very well comparable to the GA method and even better for high-dimensional problems (Ferri et al., 1994; Jain and Zongker, 1997). In our work, we apply the SFFS to the feature selection problem.

One important issue about feature selection is to determine the number of features that are appropriate for the classification

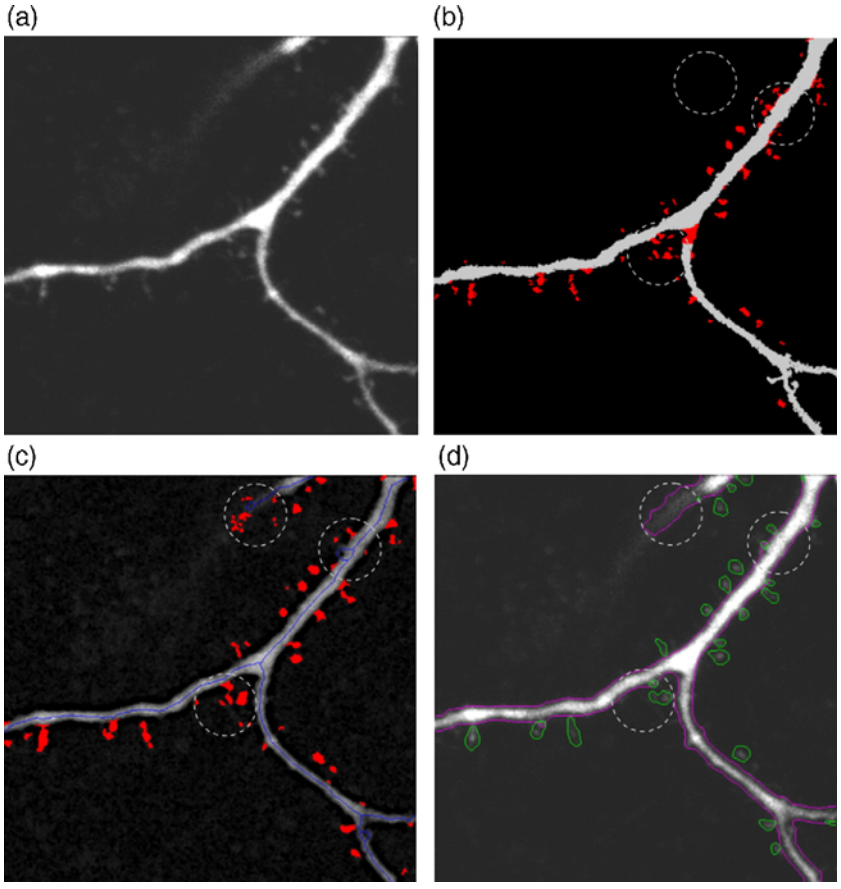

Fig. 12. Results comparison among our algorithm and the threshold based algorithm (see Koh et al., 2002; Weaver et al., 2004). (a) Original image; (b) extraction results using median axis and global threshold method, the detected spines and backbones are labeled in red color; (c) extraction results using adaptive threshold method, the detected spines and backbones are labeled in color; (d) extraction results using our method. Note the major difference of the results at circled areas. The threshold based methods tends to detect lots of pseudo spines.

problem. Because there is no criterion to determine this value, we have to try different number of features during feature selection. In practice, we select 8, 10, 12, 14, and 16 features and compare the performance (see Validation section). Once the features for the two classes are selected, they will be used to classify an unknown object into a spine or a protrusion.

Attached spine classification. There exist many classification methods in the literature, for example, the Nearest-Neighbor Classifiers (kNN), Support Vector Machines (SVM), Neural Networks, and Decision Trees. Most of these methods are either too simple or too complicated (computationally expensive) to be used in this project. We use the Linear Discriminate Analysis (LDA) as the classification algorithm in our work. LDA easily handles the case where the within-class frequencies are unequal and their performances have been examined on randomly generated test data. It is a transform-based method which attempts to maximize the ratio of *between-class* variance to the *within-class* variance in any particular data set, thereby guaranteeing maximal separability (Fukunaga, 1990). LDA is to find an optimal linear transformation, which can retain the class separablility while reducing the variation within each class. In the training set, suppose we have two feature sets for the two classes (spine **S** and protrusion **T**) in the forms given below:

$$\mathbf{S} = [S_1,\ S_2,\ L,\ S_m]^T,\ \mathbf{T} = [T_1,\ T_2,\ L,\ T_n]^T \tag{11}$$

where S_i=[s_{i1}, s_{i2}, L,s_{i10}], i=1:m, and T_j=[t_{j1}, t_{j2}, L, t_{j10}], j=1:n. Let μ_s, μ_T be the mean vector of the two sets respectively and μ be the mean vector of the entire training set. The between-class scatter matrix (S_b) of the training set defines the scatter of the expected vectors around the global mean:

$$S_b = (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^T + (\boldsymbol{\mu}_T - \boldsymbol{\mu})(\boldsymbol{\mu}_T - \boldsymbol{\mu})^T \tag{12}$$

The within-class scatter matrix (S_w) of the training set defines the scatter of samples around their respective means:

$$S_w = E[(S - \boldsymbol{\mu}_s)(S - \boldsymbol{\mu}_s)^T] + E[(T - \boldsymbol{\mu}_T)(T - \boldsymbol{\mu}_T)^T] \tag{13}$$

The transformation matrix is formed by combining S_b and S_w so as to maximize the separation of the classes in the transformed domain. In this work, the transformation matrix (**H**) consists of the eigenvectors corresponding to the dominant eigenvalues of the matrix $S_w^{-1} \times S_b$. Once **H** is obtained, the training data set (**S**, **T**) and the unknown data set (**U**) are transformed to the new domain. A class is assigned to each unknown data vector in **U** based on the Euclidean distance between the transformed unknown data sample and each of the class centers in the transformed domain. Fig. 5 shows the block diagram for the spine classification using the LDA method.

Experimental results and validation

Results for dendritic backbone and spine extraction

Figs. 6 and 7 show four original neuron images and their extracted backbones and spines labeled with color. Each image and

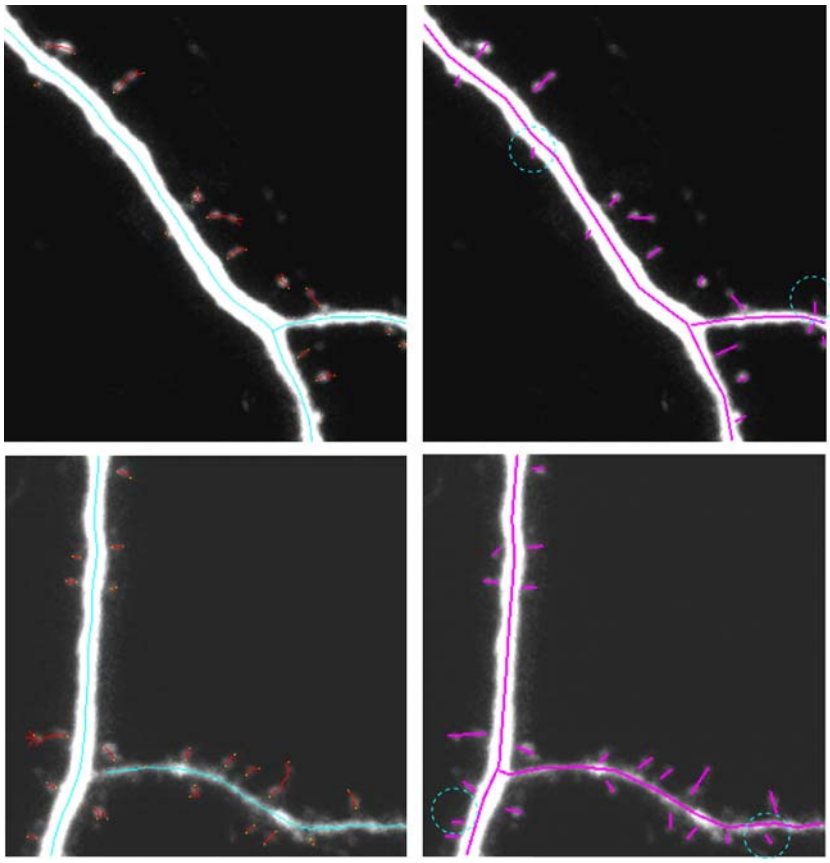

Fig. 13. Visual comparison: the two images on the left column are the results from the proposed automated method. The two other images are the results obtained manually. The spines circled in the manual results illustrate the ability of the automated method to remove the pseudo spines due to noise or protrusions.

its result represent one of the three situations: single backbone; multiple connected backbones; and multiple separated backbones. The results also show that, even with very noisy images, our proposed algorithm can accurately and robustly extract complicated backbone and spine structures. The boundaries for backbone and spines are also shown. The average running time to process one neuron image is in minutes, compared to time in hours spent if labeled manually, especially for applications of high-throughput screening, with thousands of such images generated from one experiment.

Results for spine classification

From 27 neuron images we pick 157 attached spine samples as valid spine type and 63 samples as protrusion type, for a total of 220 positive and negative spine samples in the training set. Fig. 8 shows some sample spine images from the training set. We calculate the 12 geometric features and 36 moment features for each of the sample spine image. A 220×48 feature matrix is then generated with each row representing a single sample spine and each column a single feature. After the feature normalization, we examine the correlation between each pair of columns in the feature matrix by calculating the pairwise linear correlation coefficients. The correlated features are removed in order to make the feature matrix positive definite. After the feature decorrelation, only 22 features are left for feature selection and spine classification. The SFFS method is applied to the 22 uncorrelated features to select 8, 10, 12, 14, or 16 features for further classification. The *error rate* is defined as the proportion of errors made over the whole set of testing samples. The error rate values are obtained using the 10-fold cross-validation method. First, the whole sample spines are split into 10 subsets of equal size. Second, each subset is used in turn for testing, the remainder for training. Third, the SFFS method is applied and the error estimates are averaged to yield an overall error rate. In Table 1 we show the performance of SFFS in terms of error rates for different number of selected features. In this work we use 10 features for classification.

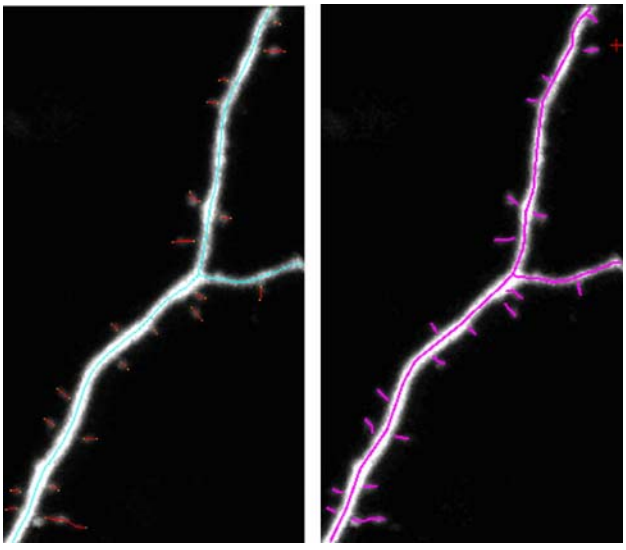

Fig. 14. Visual comparison: two example images on which both automated and manual method obtain the same result.

Table 2
Comparison of spine number and average spine length by two methods

Method	Spine number	Average backbone length (μm)	Average spine length (μm)	Spine density (in 1/μm)
Manual	254	49.133	1.147	0.35
Automated	260	52.158	1.247	0.33

We further test the feature selection method using a holdout procedure: from the 220 samples, we randomly select 80% of them (176 samples) as the training set and remaining 20% (44 samples) as the testing set. 10 Features are selected by SFFS. The classification result shows that there are 2 errors out of 44 samples and the error rate is 4.55% (2 out of 44). That means, the spine classifier can successfully predict the spine type for 95.45% of the samples.

Given the training set and its classification results, we apply LDA as the classifier to detect the spine type for unknown spines. That is, once an attached object is detected, it is fed to the classifier as an unknown test data. The classifier will determine whether or not it belongs to spines, based on the predefined training data set and its classification results. Figs. 9 and 10 show the spine detection results before and after the spine classification. It is observed that some protrusions are removed by the classifier.

We compare our spine detection results with those produced by the threshold based algorithms (Koh et al., 2002; Weaver et al., 2004). Figs. 11 and 12 show the comparison results for the four test neuron images. The threshold based method accepts many noisy spots as spines. Furthermore, some parts of the dendrites are incorrectly detected as spines (for example, see Fig. 12(b)). Our proposed method is robust enough to bypass noise around spines and the dendritic backbones are detected very accurately.

Validation

In order to evaluate our proposed algorithm for dendritic backbone and spine extraction, we compare our results with those of semi-automatic method and fully manual method by visual inspection and quantitative comparison. The validation process is designed as follows: we randomly select 14 neuron images from a

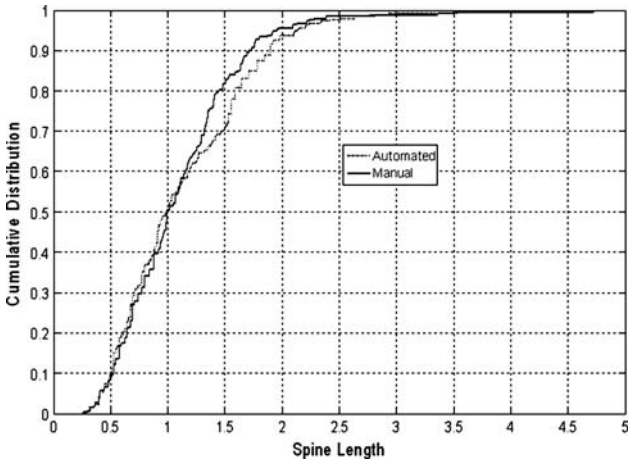

Fig. 15. Cumulative distribution of spine length comparison between automated labeling and manual labeling.

Table 3
Quantitative comparison of each individual test image

Image	Spine number			Total spine length (241) (pixel)		Backbone length (μm)	
	Manual	Automated	Both	Manual	Automated	Manual	Automated
fd124b003	20	20	19	17.524	19.387	56.24	62.229
fd124b005	16	14	14	14.784	20.706	48.414	51.899
fd124b007	16	18	15	11.537	13.725	46.464	50.040
fd124c003	11	13	11	14.213	13.913	28.149	30.565
fd124e006	12	11	10	10.695	9.951	70.128	71.976
fd125f002	21	22	20	29.151	32.098	44.666	46.067
fd136c007	17	16	16	22.696	28.267	72.993	77.874
fd136d004	9	13	9	9.798	10.831	43.682	47.017
fd136e004	21	19	19	19.850	23.124	48.456	49.006
fd136f003	33	34	32	42.065	49.040	60.818	66.409
fd136f005	22	22	22	26.970	27.116	32.964	34.061
fd140f003	20	20	20	19.498	19.851	50.874	55.322
fd140f005	15	14	13	12.535	12.354	43.079	45.274
fd149d003	21	24	21	25.095	20.154	40.933	42.472
Total	254	260	241	276.409	300.517	687.86	730.2127
Average				1.147	1.247	49.133	52.158

Note. "Both" means we only count spines that are detected by both manual labeling and automated labeling.

total of 109 images. For each image we randomly select one dendrite region and its spine structures to alleviate the burden of manual process. We manually label the backbones and the centerlines of the spines in each selected dendrite and use the results as the ground truth. Our proposed algorithm is then applied on each dendrite and the results are compared quantitatively with the manual results in terms of spine number, spine length, backbone length, and spine density. The spine density is defined as the number of spines per unit backbone length.

The manual method detects 254 spines and the automated method detects 260 spines. Among these spines a total of 241 spines are detected by both methods, 13 additional spines are detected only by manual method, and 19 additional spines are detected only by automated method. Visual comparisons are demonstrated in Figs. 13 and 14. In Fig. 13 the spines circled are detected only by manual method. Since these spines are due to the noise and the rough dendrite boundaries, the automated classifier treats them as invalid attached spines and removes them. Fig. 14 shows an example in which case the two methods obtain exactly the same number of spines and same locations for detected spines. This proves the accuracy of our proposed method.

Table 2 compares the two methods quantitatively. The average spine length and spine density are calculated based on the 241 common spines. Table 3 compares the spine number, spine length, and spine density for each individual test image. "Both" in Table 3 means that we only count the spines if they are detected by both manual labeling and automated labeling. The measured backbone length by the automated method is always larger than that by the manual method. This is partly due to the fact that the automated extracted centerline is a smoothed curve, while the manually extracted centerline consists of a set of straight lines, which is usually shorter than a curve. Fig. 15 compares the cumulative distributions of all the spine lengths between automated labeling and manual labeling. Fig. 16 compares individual spine length of our proposed method with those of manual labeling for test neuron image fd136f005. Both visual inspection and quantitative comparison show that the proposed method is accurate and complete for the automated detection of dendritic backbone and spines.

Table 4
Typical parameter values

Parameter	Typical value	Description
Upper threshold	3.0	Curvilinear structure detection
Lower threshold	1.0	Curvilinear structure detection
Linking distance	5	Curvilinear structure detection
Minimum line length	5	Curvilinear structure detection
Strength threshold	1.0	Boundary detection
Minimum backbone length	80	Minimum backbone length

Parameter selection

The standard deviation for the Gaussian kernel is determined automatically. Other parameters for the curvilinear structure detection, such as the lower and upper threshold, the minimum line length, and the linking distance, can be fixed loosely using a large range and does not affect the final results significantly. The parameters for the backbone and spine detection include the

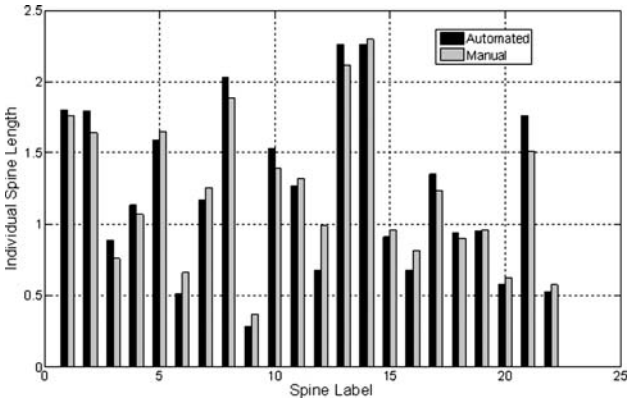

Fig. 16. Comparison of individual spine length between automated labeling and manual labeling for neuron image fd136f005.

weakness threshold and the minimum backbone length. These two parameters may vary for different images, but can be fixed for a set of images from the same biological experiment, or images with similar intensity distributions. Table 4 lists the typical parameter values that are used in this work.

Conclusion and discussion

We have presented a novel approach for automated detection of dendritic backbones and spines in neuron images. The algorithm is able to accurately and smoothly extract the backbones and centerlines of the spines based on the curvilinear structure detection method. A novel boundary detection method is derived and applied to rapidly segment the spine areas. The algorithm can detect and quantitate both attached spines and detached spines. For attached spines, a classifier is used to further remove protrusions based on their geometry and moment features. The algorithm is highly automated, requiring minimum or no human interaction. The automated process is fast. The running time is only in minutes for processing a 512*512*10 neuron image with a Pentium IV 2.8G Hz processor. The experimental results show that the algorithm is able to extract the dendritic backbones and spines accurately and completely, and provide detailed information consistently. Our algorithm outperforms other methods in that it provides fast and fully automated processing and more accurate detection of dendritic backbone and spines in images with poor quality and complicated structures.

At present, the algorithm is applied on the 2D projection of the 3D neuron image stacks. So, some of the 3D morphological characters, such as spine volume and spine shapes, are not available. In order to access the volumetric measurements, it is necessary to study the 3D dendritic structures, which can be achieved by extending the curvilinear structure detection method to 3D. The classification scheme used in this work is simple and effective, and it only can determine if an attached spine is valid or invalid. To study the functional significance of the dendritic spines, spines can be further classified as stubby, mushroom, and thin types, based on their shape characteristics. Such classification, however, requires more spine features and a more sophisticated classification scheme.

Acknowledgments

The authors would like to acknowledge the collaboration with their biology collaborators in the Department of Neurobiology at Harvard Medical School. The authors also would like to thank Ms. Maomao Cai for her kind help on validating the results. The authors would like to thank Mr. Haiying Liu at Surgical Planning Lab of Brigham & Women's Hospital for his help on image analysis and visualization with 3D Slicer. The authors would appreciate the members at HCNR CBI and the members at the Bioimage analysis group for their valuable advice, discussion, and facility support. This research is funded by the HCNR Center for Bioinformatics Research Grant, Harvard Medical School and a NIH R01 LM008696 Grant to Wong, S.T.C.

References

Al-Kofahi, K.A., Lasek, S., Szarowski, D.H., Pace, C.J., Nagy, G., Turner, J.N., Roysam, B., 2002. Rapid automated three-dimensional tracing of neurons from confocal image stacks. IEEE Trans. Inf. Technol. Biomed. 6, 171–187.

Bresenham, J.E., 1965. Algorithm for computer control of a digital plotter. IBM Syst. J. 4, 25–30.

Cantu-Paz, E., 2002. Feature subset selection by estimation of distribution algorithms. Genetic and Evolutionary Computation Conference (GECCO) San Francisco, CA.

Carter, A.G., Sabatini, B.L., 2004. State-dependent calcium signaling in dendritic spines of striatal medium spiny neurons. Neuron 44, 483–493.

Engert, F., Bonhoeffer, T., 1999. Dendritic spine changes associated with hippocampal long-term synaptic plasticity. Nature 399, 66–70.

Ferri, F.J., Pudil, P., Hatef, M., Kittler, J., 1994. Comparative study of techniques for large-scale feature selection. Pattern Recogn. Pract. 403–413.

Florack, L.M.J., Romeny, B.M.t.H., Koenderink, J.J., Viergever, M.A., 1992. Scale and the differential structure of images. Image Vis. Comput. 10, 376–388.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, San Diego, CA.

Harris, K.M., 1999a. Structure, development, and plasticity of dendritic spines. Curr. Opin. Neurobiol. 3, 343–348.

Harris, K.M., 1999b. Structure, development, and plasticity of dendritic spines. Curr. Opin. Neurobiol. 9, 343–348.

Herzog, A., Krell, G., Michaelis, B., Wang, J., Zuschratter, W., Braun, A.K., 1997. Restoration of three-dimensional quasi-binary images from confocal microscopy and its application to dendritic trees. Proc. SPIE Vol. 2984, p. 146–157, Three-Dimensional Microscopy: Image Acquisition and Processing IV Bellingham, WA, 1997.

Holland, J.H., 1996. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, 2nd ed. MIT Press, Cambridge, MA.

Jain, A., Zongker, D., 1997. Feature selection: evaluation, application, and small sample performance. IEEE Trans. Pattern Anal. Mach. Intell. 19, 153–158.

Kempen, G.M.P.v., Vliet, L.J.v., Verveer, P.J., Voort, H.T.M.V.d., 1996. A quantitative comparison of image restoration methods for confocal microscopy. J. Microsc. 185, 354–365.

Khotanzad, A., Hong, Y.H., 1990. Invariant image recognition by Zernike moments. IEEE Trans. Pattern Anal. Mach. Intell. 12, 489–497.

Kiraly, A.P., Higgins, W.E., McLennan, G., Hoffman, E.A., Reinhardt, J.M., 2002. 3D human airway segmentation methods for virtual bronchoscopy. Acad. Radiol. 9, 1153–1168.

Kiraly, A.P., Helferty, J.P., Hoffman, E.A., McLennan, G., Higgins, W.E., 2004. 3D path planning for virtual bronchoscopy. IEEE Trans. Med. Imag. 23.

Koh, I.Y.Y., Lindquist, W.B., Zito, K., Nimchinsky, E.A., Svoboda, K., 2002. An image analysis algorithm for dendritic spines. Neural Comput. 14, 1283–1310.

Lendvai, B., Stern, E.A., Chen, B., Svoboda, K., 2000. Experience-dependent plasticity of dendritic spines in the developing rat barrel cortex in vivo. Nature 404, 876–881.

Lim, J.S., 1989. Two-Dimensional Signal and Image Processing. Prentice Hall PTR.

Maletic-Savatic, M., Malinow, R., Svoboda, K., 1999. Rapid dendritic morphogenesis in CA1 hippocampal dendrites induced by synaptic activity. Science 283, 1923–1927.

Pudil, P., Ferri, F.J., Novovicova, J., Kittler, J., 1994. Floating search methods for feature selection with nonmonotonic criterion functions. Proceedings of the 12th IAPR International. Conference on Pattern Recognition—Conference B: Computer Vision and Image Processing.

Sabatini, B.L., Maravall, M., Svoboda, K., 2001. Ca(2+) signaling in dendritic spines. Curr. Opin. Cell Biol. 11, 349–356.

Sarbassov, D.D., Ali, S.M., Sabatini, D.M., 2005. Growing roles for the mTOR pathway. Curr. Opin. Cell Biol. 17, 596–603.

Sauret, V., Goatman, K.A., Fleming, J.S., Bailey, A.G., 1999. Semi-automated tabulation of the 3D topology and morphology of branching networks using CT: application to the airway tree. Phys. Med. Biol. 44, 1625–1638.

Steger, C., 1998. An unbiased detector of curvilinear structure. IEEE Trans. Pattern Anal. Mach. Intell. 20, 113–125.

Swift, R.D., Kiraly, A.P., Sherbondy, A.J., Austin, A.L., Hoffman, E.A., McLennan, G., Higgins, W.E., 2002. Automatic axis generation for virtual bronchoscopic assessment of major airway obstructions. Comput. Med. Imaging Graph. 26, 103–118.

Tavazoie, S.F., Alvarez, V.A., Ridenour, D.A., Kwiatkowski, D.J., Sabatini, B.L., 2005. Regulation of neuronal morphology and function by the tumor suppressors Tsc1 and Tsc2. Nat. Neurosci. 8, 1727–1734.

Teh, C.-H., Chin, R.T., 1988. On image analysis by the methods of moments. IEEE Trans. Pattern Anal. Mach. Intell. 10, 496–513.

Weaver, C.M., Hof, P.R., Wearne, S.L., Lindquist, W.B., 2004. Automated algorithms for multiscale morphometry of neuronal dendrites. Neural Comput. 16, 1353–1383.

Xiong, G., Zhou, X., Degterev, A., Ji, L., Wong, S.T.C., 2006. Automated neurite labeling and analysis in fluorescence microscopy images. Cytometry 69A, 494–505.

Zhou, X., Liu, K., Sabatini, B.L., Wong, S.T.C., 2006. Mutual information based feature selection in studying perturbation of dendritic structure caused by TSC2 inactivation. Neuroinformatics 4, 81–94.