A SURFACE BASED APPROACH FOR DENDRITIC SPINE DETECTION

K. Sam Eliezer*

Mr.S.Athinarayanan**

***Abstract*—** Neuron reconstruction and dendritic spine identification on a large data set of microscopy images is essential for understanding the relationship between morphology and functions of dendritic spines.Dendrites are the tree-like structures of neuronal cells, and spines are small protrusions on the surface of dendrites. Spines have various visual shapes (e.g., mushroom, thin, and stubby) and can appear or disappear over time. Existing neurobiology literature shows that the morphological changes of spines and the dendritic spine structures are highly correlated with their underlying cognitive functions.How to accurately and automatically analyse meaningful structural information from a large microscopy image data set is a difficult task. One challenge in spine detection and segmentation is how to automatically separate touching spines. In this paper, based on various global and local geometric features of the dendrite structure touching spines are detected and to segment them a breaking-down and stitching-up algorithm is used.

***Index terms:* Microscopy images, normalized cut, spine detection, surface-based segmentation, single spines, touching spines.**

* PG Student, M.Tech (IT), PSN College of Engineering and Technology,Tirunelveli.

** M.Tech, AssociateProfessor, M.Tech(IT), PSN College of Engineering and Technology, Tirunelveli.

I . INTRODUCTION

Nowadays, microscopy image processing techniques have been widely used in diverse fields such as medicine, biological and cancer research, drug testing, and metallurgy. These techniques aim to enhance,extract, analyse information from digital images acquired by microscope systems [7],[8]. Microscopy images could be 2-D, 3-D, and in time series. Currently, microscopy image processing faces significant challenges due to the following reasons:

1)objects of interest are often touching/overlapping each other or irregularly arranged, with no definite shapes; 2) Illumination variations are not distinctive in thick specimens. Absorption, scattering, and diffraction of the light by structures located above and below the focal plane cause image intensity fall off in deep specimens; 3) The background of microscopy images is usually very noisy.

In neurobiology research,neuron reconstruction and dendritic spine identification on a large data set of microscopy images is essential for understanding the relationship between morphology and functions of dendritic spines.Existing neurobiology literature shows that the morphological changes of spines and the dendritic spine structures are highly correlated with their underlying cognitive functions.Therefore, how to efficiently and accurately detect and extract spines is crucial yet challenging problem. In this paper, we propose a novel 3-D surface based dendritic spine detection and segmentation approach (its pipeline is illustrated in Fig. 1). Basically, we first extract the backbones and reconstruct the 3-D neuron surface from the noise-reduced neuronal images. Then, by analyzing three geometric features on the 3-D surface, spines are separated from the dendrite. After that, based on the spine classification outcomes (i.e., single or touching spines) from our shape analysis module, a normalized cut algorithm [5] is adapted to separate the touching spines in the following two-phase protocol: i) The touching spines are decomposed into small patches, and then ii) the patches are stitched together through maximization of an energy function.

The rest of the paper deals with the following sections: section II explains the previous work, section III discuss about preprocessing, section IV discuss about geometric feature extraction,section V describes separating spines with dendrite, section VI about spine shape analysis,section VII shows the expereimental results,and section VIII about the conclusion of the paper .

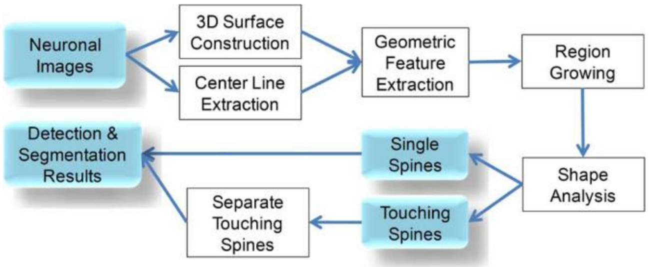

Fig. 1. Pipeline of the proposed spine detection and segmentation approach

II. PREVIOUS WORK

Existing dendritic spine detection methods can be roughly divided into the following two categories: 2-D MIP maximum intensity projection) image-based algorithms [2] and 3-D databased algorithms [3],[5],[14]. The major Drawbacks of the 2-D MIP methods are: 1) 3-D microscopy images are projected to a 2-D plane; a significant amount of information such as spines that are orthogonal to the imaging plane will be inevitably lost; 2) Dendritic structures that overlap along the projection direction are difficult to extract. 3-D data-based algorithms either use voxel clustering [3] or extract the dendritic skeleton structure of neurons using a medial geodesic function [14].Some existing commercial software tools (e.g., Imaris) perform semi-automated dendrite and spine detection. However, such a semi-automated process is typically costly, time-consuming,and subject to human bias.

Previous efforts on dendritic spine detection can be roughly divided into two categories: classification-based approaches [3], [16] and centerline extraction-based approaches [3], [14], [9]. Classification-based approaches separate points into different groups using a trained classifier.For example, Rodriguez et al. [3] proposed an automated 3-D spine detection approach using voxel clustering. Li et al. [16] proposed a 3-D surface-based classification algorithm. In their work, 3-D neuron surface is reconstructed through the marching cubes algorithm. Each vertex on the surface is classified into vertex on the spines or vertex on the dendrite based on three geometric features. Then, touching spines are separated by adopting surface-based watershed algorithm.Centerline extraction-based approaches detect all the possible centerlines of certain objects in the image (e.g., using a curvilinear structure detector [14] or local binary fitting model of level sets) and treat dendritic spines as small protrusions attached to the dendrites. However, these methods often require empirically designed post-processing procedures and limited to processing relatively simple neuron structures. Later, Koh et al.[9] adopt a thinning method to extract centerlines and apply the grassfire propagation technique to assign each dendritic point a distance to the medial axis of the dendritic structure.Since segmentation is achieved by global thresholding and limited geometric information is considered for spine detection,this method may detect pseudo spines. Janoos et al. [4] present a method for dendritic skeleton structure extraction using a curve-skeletons approach based on the medial geodesic function which is defined on the reconstructed isosurfaces.

III. IMAGE PREPROCESSING

In the data preprocessing stage, a 3-D median filter with a 3X3X3 kernel size was first applied to the images to remove noise. The median filter is a commonly used nonlinear operator that replaces the original gray level of a pixel by the median of the gray levels of the pixels in a specified neighborhood. As a type of ranking filters, the median filter is based on the statistics derived from rank-ordering a set of elements. It is often useful because it can reduce noise without blurring edges in the image. The noise-reducing effect of the median filter depends on two factors: 1) the spatial extent of its neighborhood; and

2)the number of pixels involved in the median calculation.

We chose the median filter since it is able to remove certain noise that cannot be removed by conventional convolution filtering. Then, to correct uneven illumination degradation, a top-hat filter was adopted. After that, fuzzy C-mean clustering was used to cluster the image into 3 clusters: background,weak spines, and dendrite with strong spines. Weak spines and the dendrite represent the neuron. Subsequently, we employed the marching cubes algorithm to reconstruct the 3-D surface of the neuron. Then, a low-pass filter and mesh decimation were used to remove noise and reduce the tessellation density.The number of iterations in the low-pass filter and the decimation factor control the smoothness of the resultant 3-D neuron surface.

After the above data preprocessing, the dendrite backbone and the approximated radiuses along the backbone are generated by extending the Rayburst sampling algorithm.Inspired by this idea, our Rayburst sampling algorithm works as follows: at the beginning, a seed point (an initial center point) inside of the dendrite is selected by users. Then, 2-D Rayburst sampling in both the XY and XZ (or YZ) planes are adopted. A threshold is used to control the maximally allowed intensity difference between the center point of a neuron and its dendritic boundary. The length of the shortest ray in XY plane is the estimated diameter, and the location of the center point is updated as the midpoint of the shortest ray. Then, rays sampled in the XZ (or YZ) plane are used to adjust the Z coordinate of the center point. The next two center points toward both the ends of the dendrite are assumed to follow the local orientation of the current center point: the orientation of the longest ray in XY plane is the approximated local orientation of the dendrite. This procedure repeats until the predicted center point reaches the border of the stack or it goes into the background. If the dendrite contains a branch structure, a user-specified seed point is needed for each branch. In this paper, the number of rays and are experimentally set to 36 and 80, respectively. Fig. 2 illustrates this process and one example result.

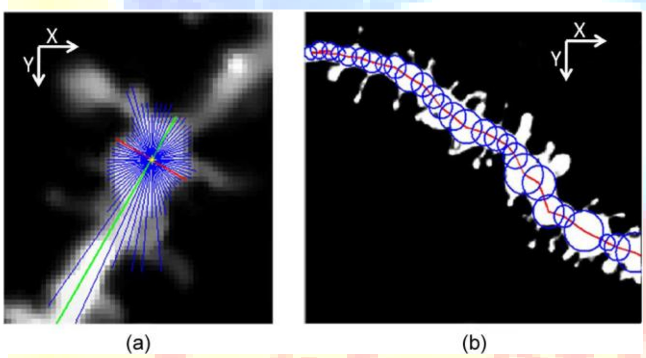

Fig. 2. Illustration of Rayburst sampling. (a) Rayburst sampling in XY plane.(Blue lines) Rays casting out in all the directions from the predicted center point. (Red line) Ray with the shortest length (diameter). (Green line) Longest length (local orientation). (b) Extracted backbone of the neuron and the estimated radiuses are illustrated in the maximum intensity projection image.

IV.GEOMETRIC FEATURE EXTRACTION

In our approach, the following three geometric features are calculated for each 3-D face to separate neuronal spines from the dendrite.

1) Distance to the dendrite backbone*:* The distance between a vertex v on the neuron surface and the backbone is defined as the shortest distance between v and any point of the backbone. In this way, the distance between each face f and the backbone is defined as the average distance between f's vertices and the backbone. This distance feature is chosen based on the observation that vertices on the spine surface usually have larger distances (to the backbone) than those on the dendrite.

2) Mean curvature on the surface: The mean curvature of each face on the 3-D neuron surface is computed as the average curvature of its surrounding vertices.

3) Normal variance*:* Normal variance of each face f is defined as the average angle between the face normal and the vector that is perpendicular to the backbone and passes each of 's surrounding vertices.

V. SEPERATING SPINES WITH DENDRITE

We normalize the above features and add them altogether to generate a score map. Then, we use a surface-based region growing algorithm to separate spines from the dendrite. In this process, faces whose values in the score map are greater than (a user-specified threshold) are randomly selected as seed points. If the score value of a neighbor of each seed point is larger than and it has not been visited previously, then it will be chosen as a new seed point. This process repeats until all the faces have been visited. In this paper, is experimentally set to 0.2. One example result after the region growing is shown in Fig. 3. From this figure, we can see that although most spines are detected and separated from the dendrite, some spines are still touching together. As such, the next step of our approach is to automatically detect and separate touching spines.

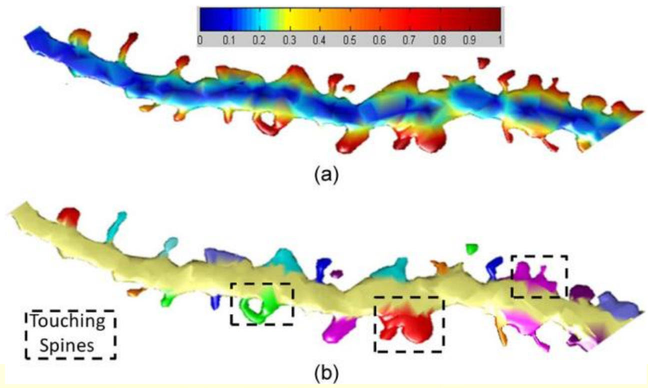

Fig. 3. Visualization of the computed score map after region growing. (a) Visualization of the score map. Each face on the surface has a score from 0 to 1. (b) Threshold is used in the region growing algorithm to separate spines with dendrite. Spines are labeled in different colors. (Black rectangles) Touching spines.

VI. SPINE SHAPE ANALYSIS AND CATEGORIZATION

After spines are separated from the dendrite, we perform 3-D shape analysis on all the spines in order to automatically categorize them into single spines or touching spines. The core idea of our shape analysis algorithm is illustrated in the left of Fig. 4. Basically,if any sampling line is sent from one side of the spine to other side, a single spine will have at most two faces intersecting with the sampling line. By contrast, in this case a touching spine will have at least four faces intersecting with the sampling line.Based on this key observation, for each spine segment outputted from the above region growing algorithm, we first randomly select sampling vertices on the spine surface. Then, sampling lines are sent out from the sample vertices and their directions are in parallel to the normals of the sampling vertices. In our experiments,we found worked sufficiently well. To ensure the sampling vertices are from one side of the spine, they are only chosen from the vertices whose normal variance features are high . If more than two sampling lines intersect with four or more faces on the spine segment, we categorize the spine segment as a touching spine segment; otherwise, we categorize it as a single spine segment.

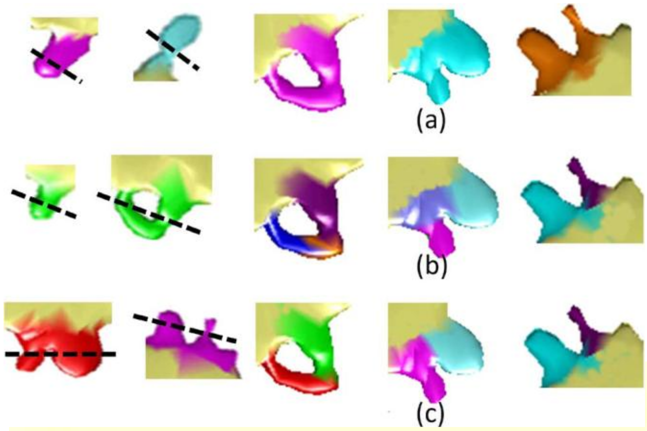

Fig. 4. (Left) Illustration of our spine shape analysis and categorization principle.When any sampling line is sent from one side of a spine to the other side,if it intersects with at most two faces, then it is a single spine; otherwise, it is a touching spine. (Right) Visualization of how we separate touching spines:(a) initial touching spines, (b) the results after the touching spines are broken into small patches using a normalized cut algorithm, and (c) the final results after the patches are stitched together to form spines.

Breaking a touching spine into patches:A normalized cuts based algorithm is usedto break a touching spine into patches based on its local geometric features. The (b) panel in the right of Fig.5 shows some example results after we break touching spines into patches based on local geometric features.

Stitching spine patches: In this step, a spine stitching algorithm to group some patches together to form a new spine based on high-level geometric features. First examine the boundary of each patch as follows: if one patch is connected with another patch, then add them as a pair in a candidate list. Whether two patches and should be stitched together is primarily based on two high level geometric metrics (illustrated in Fig. 5). The first geometric metric is the projected distance between the centroids of the two connecting patches.The second geometric metric is the intersecting volume ratio of the bounding boxes of the two connecting patches.

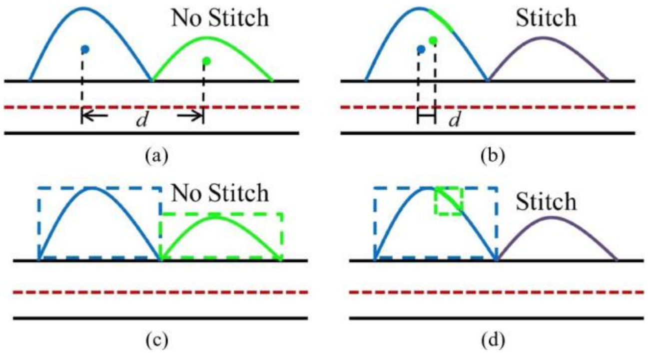

Fig 5: Illustration of two high-level geometric features/metrics. Blue and green curves: two connecting patches. Red dashed line: backbone.

VII. EXPERIMENTAL RESULTS

To validate the segmentation performance of our approach, we compared the detection performance of our algorithm with two state-of-the-art, neuronal spine detection algorithms [3], [16] and the ground truth.Our approach achieved a higher spine detection accuracy,a smaller spine missing rate, and a lower false positive rate than the two chosen spine detection and segmentation algorithms [3],[16]. However, we also can see that our approach still missed the detection of a few tiny spines, though the same tiny spines were also misdetected by the two other algorithms [3],[16].

VII. DISCUSSION AND CONCLUSION

This paper introduces a novel 3-D surface based dendritic spine detection and segmentation approach. It is noteworthy that although our approach can achieve a high spine detection and segmentation accuracy, it still misdetected a few tiny spines. This in part comes from our region growing algorithm that only uses a single global threshold to separate spines from the dendrite. In the future, to further improve the accuracy of our algorithm, we plan to investigate new algorithms to adaptively learn this parameter based on the local geometric features.

Since the test data set we were able to acquire only encloses straight neurons, not curvy neurons, we were not able to test our algorithm on curvy neuron data sets. Considering curvy neurons typically have a higher complexity than straight ones, directly applying this current work to curvy neurons may not work well; however, we believe a proper extension and adaptation of this work will help to solve the problem. Another limitation of the current work is that it does not handle single spines with complex shapes. For single spines with complex shapes (e.g.,multi-headed spines), more sophisticated shape analysis algorithms need be developed.We believe that with a sufficient spine shape training data set, accurately distinguishing complex shape spines (e.g., multi-headed spines) from touching spines can be achieved.

REFERENCES

[1] J. Cheng, X. Zhou, E. Miller, R. M. Witt, J. Zhu, B. L. Sabatini, and S. T. Wong, “A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy,” J. Neurosci. Methods, vol. 165, no. 1, pp. 122–134, Sep. 2007

[2] Q. Li, Z. Deng, Y. Zhang, X. Zhou, U. V. Nägerl, and S. T. Wong, “A global spatial similarity optimization scheme to track large numbers of dendritic spines in time-lapse confocal microscopy,” IEEE Trans. Med. Imag., vol. 30, no. 3, pp. 632–641, Mar. 2011.

[3] A. Rodriguez, D. B. Ehlenberger, D. L. Dickstein, P. R. Hof, and S. L. Wearne, “Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images,” PLoS One, vol. 3, no. 4, p. e1997, 2008.

[4] Y. Zhang, K. Chen, M. Baron, M. A. Teylan, Y. Kim, Z. Song, P.Greengard, and S. T. Wong, “A neurocomputational method for fully automated 3-D dendritic spine detection and segmentation of mediumsized spiny neurons,” Neuroimage, vol. 50, no. 4, pp. 1472–1484, May 2010.

[5] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 888–905, Aug.2000.

[6] E. Kalogerakis, A. Hertzmann, and K. Singh, “Learning 3-D mesh segmentation and labeling,” in Proc. ACM SIGGRAPH Papers, 2010, pp.102-1–102-12.

[7] F. Li, X. Zhou, J. Ma, and S. T.Wong, "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," IEEE Trans. Med. Imag., vol. 29, no. 1, pp. 96–105, Jan. 2010.

[8] A. Kuijper and B. Heise, "An automatic cell segmentation method for differential interference contrast microscopy," in Proc. 19th ICPR,2008, pp. 8–11.

[9] I. Y. Y. Koh, W. B. Lindquist, K. Zito, E. A. Nimchinsky, and K. Svoboda, "An image analysis algorithm for the fine structure of neuronal dendrites," Neural Comput., vol. 14, pp. 1283–1310, 2002.

[10] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin, "Fast mesh segmentation using random walks," in Proc. ACM Symp. Solid Phys. Model., 2008, pp. 183–191.

[11] A. Golovinskiy and T. Funkhouser, "Randomized cuts for 3-D mesh analysis," ACM Trans. Graph., vol. 27, no. 5, pp. 145-1–145-12, Dec.2008.

[12] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonisation using the shape diameter function," Visual Comput., vol. 24, no. 4, pp. 249–259, Mar. 2008.

[13] A. Golovinskiy and T. Funkhouser, "Technical section: Consistent segmentation of 3d models," Comput. Graph., vol. 33, no. 3, pp. 262–269,Jun. 2009.

[14] F. Janoos, K. Mosaliganti, X. Xu, R. Machiraju, K. Huang, and S. T. Wong, "Robust 3-D reconstruction and identification of dendritic spines from optical microscopy imaging," Med. Image Anal., vol. 13, no. 1, pp. 167–179, Feb. 2009.

[15] Q. Li, Z. Deng, "A surface based 3-D dendritic spine detection approach from confocal microscopy images" IEEE Trans. on Image processing., vol. 21, no. 3, pp. 1223–1230, Mar. 2012.

[16] Q. Li, X. Zhou, Z. Deng, M. Baron, M. A. Teylan, Y. Kim, and S. T. C.Wong, "A novel surface-based geometric approach for 3-D dendritic spine detection from multi-photon excitation microscopy images," in Proc. ISBI, 2009, pp. 1255–1258.

Author's Profile:

K. Sam Eliezer received the B.Tech degree in Information Technology from, SCAD college of Engineering and Technology, Tirunelveli in 2010. Presently pursuing M.Tech degree in Information Technology in PSN College of Engineering and Technology, Tirunelveli.