A RAPID 2-D CENTERLINE EXTRACTION METHOD BASED ON TENSOR VOTING

Zhe Leng[1], Julie R. Korenberg[2], Badri Roysam[3], Tolga Tasdizen[4,5]

[1]The Department of Mechanical Engineering, University of Utah, Utah, USA
[2]The Brain Institute, University of Utah, Utah, USA
[3] Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, New York, USA
[4]Electrical and Computer Engineering Department, University of Utah, Utah, USA
[5]The Scientific Computing and Imaging Institute, University of Utah, Utah, USA

ABSTRACT

Centerline extraction is widely used in medical image processing. It can benefit applications such as building the connectivity map of neurons from microscopic images as well as examining retina vessels for preventing blindness. Many methods have been developed to extract centerlines from 2-D images. An algorithm based on 2-D rapid tensor voting is proposed in this paper. This method uses the Canny edge detector and a simple ridge finding algorithm to roughly extract centerlines, which is fast, does not require any seeds and allows the object to be disconnected. Then efficient 2-D tensor voting is applied to enhance the centerline, which can rapidly bridge the gaps caused by the earlier step and reject artifacts due to noise. We demonstrate the robustness of the algorithm and compare with existing methods. The result shows good computational efficiency as well as accuracy.

Index Terms— 2-D centerline extraction, tensor voting, axon extraction, vessel extraction

1. INTRODUCTION

2-D centerline extraction for tubular objects is widely applied in medical image processing. In learning the morphological structure and studying the function of the brain, 2-D axon centerline extraction, with the help of maximum intensity projection and 3-D recovery, is a efficient and effective method for reconstructing the neuron network structure. In retinal image analysis for pathology detection and computer aided screening, vessel centerline extraction is a prerequisite that provides the most stable and predominant image structure for later registration and other processing.

For 2-D centerline extraction, computation time is one of the key concerns. For reconstructing large scale neuronal networks, data volume greater than 10 terabytes [1] require the method to be efficient. In vessel extraction from retinal images, medical systems prefer a rapid algorithm to work with live data.

This work was supported under NIH grant 59308256.

There are generally two approaches for 2-D centerline extraction. The first approach is to evaluate each pixel and identify the pixels belonging to the object of interest [2]. The second is to generate a set of ”seed” pixels and use these as starting points to identify the set of connected pixels that are very likely part of object of interest [3, 4]. The first approach normally costs more computational time since it needs to process all the pixels in the image, but it can be implemented with pipeline accelerators. Besides, this approach can have more constant time consumption with respect to the amount of centerlines. The second approach costs less time since it only process a small portion of the pixels in the image. It can also provide an useful partial result if a computational deadline occurs. However, it needs an effective seed generating strategy to better balance recall and time consumption. If the first approach can have similar or even higher computational efficiency to the second approach with similar accuracy, it may be preferable due to this reason. Our motivation in this paper is to introduce a fast method based on efficient tensor voting that does not need seed points, is as accurate and as fast as tracing based methods.

The method proposed in this paper applies rapid 2-D tensor voting [5], which largely enhances the output of centerline extraction by removing artifacts due to noise and bridging gaps due to weak contrast. Therefore, the early stage of centerline extraction can focus on identifying the centerline sections with high fidelity only without attempting to extract the full set of connected centerline pixels. We compare our method to a rapid tracing-based method [4]. The result shows that the computational efficiency of this method is much higher, while the accuracy is similar.

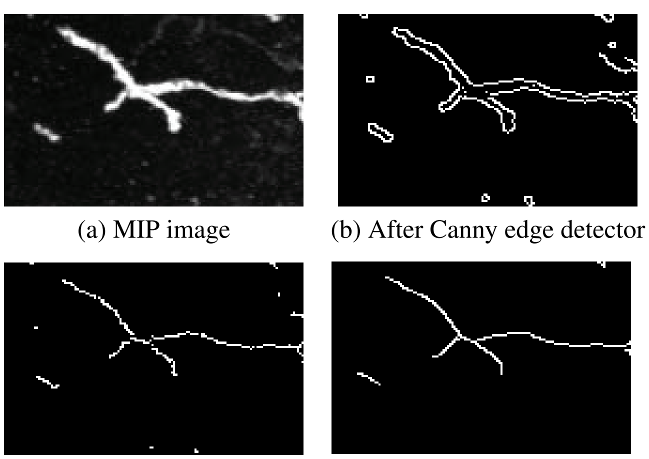
The initial centerline extraction is presented in section 2. The rapid tensor voting method is discussed in section 3. Experiment results are demonstrated in section 4 and the paper concludes in section 5.

2. INITIAL CENTERLINE EXTRACTION

We first apply the Canny Edge Detector using non-maxima suppression and hysteresis thresholding. Then, starting from every edge pixel, we move along the gradient direction (or the negative of the gradient, if objects of interest have lower intensity than the background), until we arrive at another edge pixel or reach the maximum search length. If another edge pixel with near-opposite gradient direction is found before the maximum step length is reached, the middle point between the two edge pixel can then be marked as centerline pixel. Each centerline pixel has two parameters: saliency $I \in [0,1]$ and orientation $\alpha \in [-\pi/2, \pi/2]$. Saliency is the intensity of the pixel. Orientation is the direction perpendicular to local gradient. The image containing all of these centerline pixels is referred to as the "center map" in the rest of the paper. One example of centerline extraction is shown in Fig. 1(c).

The advantage of this initial centerline extraction is that the object that is significantly larger (or smaller, if minimum search length is used) than the object of interest will not be extracted. Therefore, the cell bodies in axon extraction and the outer dark area in retina vessel extraction will be ignored. One example is shown in Fig 4(a). However, as it can be observed, though the process is fast, the centerline is not continuous and is noisy. We use tensor voting to enhance this result.

For the axon extraction problem, the image stack is first projected onto a single image before initial centerline extraction. The intensity of each x-y position in the new image is the maximum intensity along the z axis of the original image stack at this x-y coordinate. This projection is called Maximum Intensity Projection (MIP). MIP image is used as input to the initial centerline extraction. 3-D recovery is needed as a post-processing step to recover the 3-D structure, which is discussed later.



(a) MIP image (b) After Canny edge detector

(c) After centerline extractor (d) After tensor voting

Fig. 1. Example of centerline extraction process.

3. EFFICIENT 2-D TENSOR VOTING

Tensor Voting (TV) is a method first proposed by Guy and Medioni [6] for extraction or enhancement of local features (lines, curves or surface) extraction results. Local feature extraction by itself is often unreliable in noisy and complicated images. That is, the lines or curves are often noisy and interrupted. TV enhances or predicts local features by integrating clues from nearby features. In our case, the influence of nearby features is based on a given voting field designed to extract smooth curves.

In tensor voting, a 2-D tensor can be represented by symmetric, positive semidefinite 2 by 2 matrix as follows:

$$\mathbf{T} = \begin{pmatrix} a_{xx} & a_{xy} \\ a_{xy} & a_{yy} \end{pmatrix} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T \qquad (1)$$

where λ_1 and λ_2 are the eigenvalues ($\lambda_1 \geq \lambda_2 \geq 0$); $\mathbf{e}_1$ and $\mathbf{e}_2$ are the orthonormal eigenvectors. Graphical representation of this kind of tensor is ellipse, as shown in Fig 2(a). One common parameterization is to define a tensor with three parameters: orientation β, stickness s and ballness b.

$$\begin{aligned} \beta &= arccos\left(\mathbf{e}_x^T \mathbf{e}_1\right) && (2) \\ s &= \lambda_1 - \lambda_2 && (3) \\ b &= \lambda_2 && (4) \end{aligned}$$

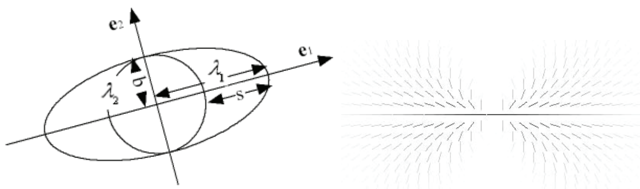
where $\mathbf{e}_x = (1,0)^T$. A tensor with ballness equal to 0 is called stick tensor. A tensor with stickness equal to 0 is called ball tensor.

In TV, the "vote" is a tensor calculated from the geometric relation between the voter and the votee. All the votes to a pixel will be summed and form the output tensor of that pixel. A typical voting method is to filter the tensor image with a non-holonomic filter, called voting field, aligned with the local tensor orientation. The original TV method [6] aligns the voting field with the tensor of every feature pixel in the spatial domain, which is very time-consuming if the feature pixels are not sparse. A better option is to use the efficient tensor voting [5] based on steerable filters [7].

The steerable filter idea is to decompose the filter into several basis filters that only consist of a single frequency of filter orientation angle. Then this filter can be steered in frequency domain using a linear combination of the basis filters. Although a general decomposition method of any filter is given by [7], a type of stick tensor voting field (the votes are all stick tensors) that has limited bandwidth (in terms of the frequency of the filter orientation angle) is proposed in [5] as follows:

$$\begin{aligned} \tilde{V}(r,\phi) &= \frac{1}{G} e^{\frac{-r^2}{2\sigma^2}} \cos^{2n}(\phi) T' && (5) \\ \mathbf{T}' &= \begin{pmatrix} 1+\cos(4\phi) & \sin(4\phi) \\ \sin(4\phi) & 1-\cos(4\phi) \end{pmatrix} && (6) \end{aligned}$$

where, (r,ϕ) is the polar coordinate of a pixel; G is the normalization term; σ is the parameter deciding the decay rate of weight vs. r, which decides the scale of the voting field; n is the order of the voting filed, which defines the decay rate of the weight vs. ϕ. It can be proven that the necessary number of basis filters is $2n + 5$. The second order ($n = 2$) stick tensor voting field is shown in Fig. 2(b).



(a) Ellipse represnetation of a tensor

(b) a typical tensor voting field

Fig. 2. Tensor and tensor voting field

In this paper, the tensor at a centerline pixel (x, y) is built from its parameters $(I_{x,y}, \alpha_{x,y})$ as follows.

$$\mathbf{T}_{x,y} = I_{x,y} \begin{pmatrix} \cos(\alpha_{x,y}) \\ \sin(\alpha_{x,y}) \end{pmatrix} \begin{pmatrix} \cos(\alpha_{x,y}), \sin(\alpha_{x,y}) \end{pmatrix} \quad (7)$$

As can be seen, these tensors are all stick tensors. For the non-centerline pixels, the local tensors are set to all zeros. Then the tensor image is decomposed into $2n + 3$ basis images, while the voting field is decomposed into $2n + 5$ basis fields. After convolution between basis images and fields, the output tensor image can be reconstructed. Detailed implementation of efficient 2-D tensor voting can be found in [5].

After tensor voting, non-maxima suppression and hysteresis thresholding is used to refine the centerline. One problem is that tensor voting tends to extend the open ends. To avoid extending the centerline too far, newly generated end points need to be extracted. A simple method that can be implemented in frequency domain is described in [8]. End point set extracted from the new center map is compared to the original center map given by early centerline extraction. The end points that is not a part of original center map will be eliminated. The result after tensor voting and post-processing including end point elimination is shown in Fig. 1(d).

It should be noted that the basis voting fields are decided by the size of the image and the two parameters. These parameters are normally constant for a series of images from the same application. Therefore, by avoiding calculating the same voting field, time cost can be reduced.

4. 3-D RECOVERY

Similar to most methods involving MIP, the recovery is to check the intensity along the z-axis for each centerline pixel and mark the voxel with maximum intensity as part of the centerline.

However, when cross-over pattern occurs, two independent axons may seem to intersect with each other on the MIP. For the case of cross-over, assigning the centerline to the voxel with maximum intensity will miss other centerline voxels at different z coordinates. These cross-overs are shown as intersections in MIP. To identify intersections, ballness and the ratio between stickness and ballness can be used as criterions. After an intersection pixel is identified, we check the peaks of intensity curve along z-axis and mark the local peaks (higher than closest local minimum plus 100 in this paper) as centerline voxels.

This method can only recovery simple cross-overs where the two axon intersect each other in MIP with a large angle. For more complicated cases, a rapid 3-D method [9] is recommended. Although 3-D based methods tracks axon better, they are generally slower than the 2-D methods.

5. EXPERIMENTS

We first compare our method to [10], which is a tracing method based on matched filter, confidence and edge measures, for extraction of vessels in retina images. We used a computer equipped with Intel Core 2 CPU and used a single core (2.00 GHz). The method proposed here cost about 6.1 seconds when working on a 565 by 584 retina image whereas [10] costs approximately 16 seconds. If it is used in processing a serial of images with same size and similar contrast, the voting field does not need to be calculated every time. In this case, the time cost of the proposed method can be farther reduced to approximately 4.2 seconds per frame. Most of the time is spent on Fast Fourier Transformation (FFT) and double floating point multiplication. These can be accelerated on proper hardware or with further code optimization.

To compare the results, DRIVE data set [11] is used. It is a retina image set for vessel extraction. After rough parameter tuning on the training data set, the proposed method obtained 78% recall and 97% precision on the test image set. In comparison, the method in [10] achieves 78% recall and 88% precision with the tracer sensitivity set to 1.0. The extracted centerline can be totally different from the manually marked centerline but still be correct. Therefore the precision and recall are calculated based on following rules: if a computer-marked centerline pixel overlaps or is adjacent to a manual marked centerline pixel, it is considered as true positive. A comparison is demonstrated in Fig. 3, which shows that our method has less false positive at the similar recall level while the time efficient is much higher.

Several cases of axon extraction are shown in Fig. 4. The color of centerline shows its position along the z-axis. More greenish means closer to the lens. The centerline extraction near cell body is shown in Fig. 4(a). The comparison between a real intersection and a cross-over are shown in Fig. 4(b) and (c). It can be seen that the depth smoothly changes in the real intersection case, but is discontinuous in cross-over case. This characteristic and the local ballness can be used in 3-D recovery for cross-over identification.

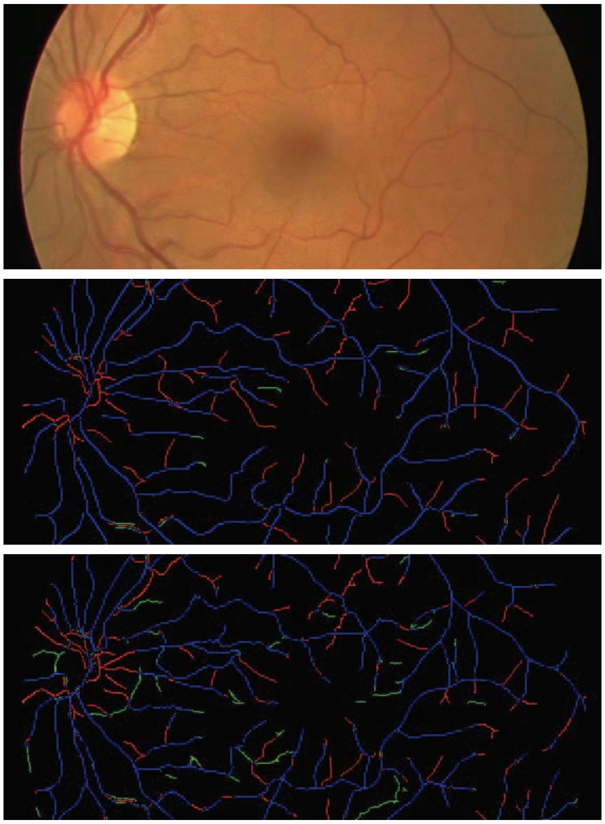


Fig. 3. Result comparison: the upper image is the original retina image, the middle one is from the proposed method, the lower image is from the method of [4]. True positives are in blue, false negatives in red, and false positives in green.

6. CONCLUSION

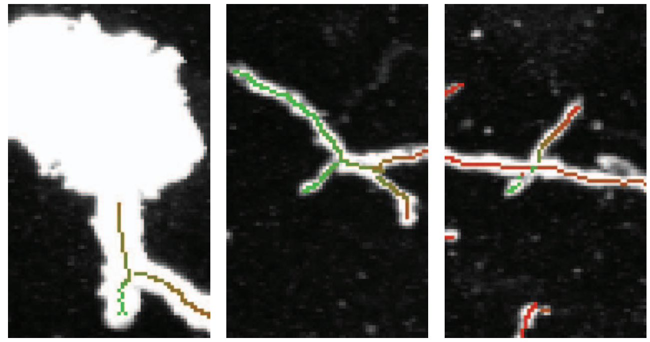
We proposed and implemented a rapid 2-D centerline extraction method involving tensor voting. This method is proven to be much more efficient than and has similar accuracy with existing method. Further improvement might be possible using machine learning techniques to tune the parameters.

7. REFERENCES

[1] J.R. Anderson, B.W. Jones, J-H Yang, M.V. Shaw, C.B. Watt, P. Koshevoy, J. Spaltenstein, E. Jurrus, Kannan U V, R. Whitaker, D. Mastronarde , T. Tasdizen, R.E. Marc. "A Computational Framework for Ultrastructural Mapping of Neural Circuitry" PLoS Biology, March, vol. 7, no. 3, pp. e74, 2009.

[2] R. Poli and G. Valli, "An algorithm for real-time vessel enhancement and detection," Computer Methods and Programs in Biomedicine, vol. 52, pp. 1-22, 1997.

[3] A. Can, H. Shen, J. N. Turner, H. L. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extrac-



(a) Avoid Cell (b) Real intersection (c) Cross over

Fig. 4. Example of centerline extraction results.

tion from retinal fundus images using direct exploratory algorithms," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 3, pp. 125-138, 1999.

[4] M. Sofka and C. V. Stewart, "Retinal Vessel Centerline Extraction Using Multiscale Matched Filters, Confidence and Edge Measures," Medical Imaging, IEEE Transactions on, vol. 25, pp. 1531-1546, 2006.

[5] E. Franken, M. van Almsick, P. Rongen, L. Florack, and B. ter Haar Romeny, "An Efficient Method for Tensor Voting Using Steerable Filters," *European Conference in Computer Vision ECCV 2006*. vol. 3954: Springer Berlin / Heidelberg, 2006, pp. 228-240.

[6] G. Guy, G. Medioni, "Inferring global perceptual contours from local features," *International Journal of Computer Vision*, 20(12), 1996, pp. 113-133.

[7] W.T. Freeman, E.H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9), 1991, pp. 891-906.

[8] G. Medioni, M.S. Lee, C.K. Tang, "A Computational Framework for Segmentation and Grouping", Elsevier, 2000

[9] R. Srinivasan, Z. Xiaobo, E. Miller, L. Ju, J. Lichtman, and S. Wong, "3-D Centerline Extraction of Axons in Microscopic Stacks for the Study of Moter Neuron Behavior in Developing Muscles," 4th IEEE International Symposium in Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007., 2007, pp. 93-96.

[10] M. Sofka and C. V. Stewart, (2007) "Likelihood Ratio Vesselness tracing excutable trained on DRIVE dataset for Windows," Available: http://www.cs.rpi.edu/~sofka/download/windows/vessel_tracer_DRIVE.zip

[11] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M. Abramoff, J. M. Fitzpatrick and M. Sonka, Eds., "Comparative study of retinal vessel segmentation methods on a new publicly available database," in Proc. SPIE Conf. Med. Imag., 2004, pp. 648656.