Algorithm for Identifying Axons

Jason Creighton

Abstract—**This paper presents a method for detecting and counting axons, or nerve fibers, in an image. This method is demonstrated on images containing a cut-away view of a nerve bundle, which contains many axons that must be identified and counted. This technique can be applied to a variety of applications of similar nature. The method utilizes spatial filtering, morphological operations, threshold, and binary image manipulation techniques.**

Index Terms—**Axon identification, Histogram equalization, Morphological operations, Threshold**

I. INTRODUCTION

AXONS are defined as the nerve fiber that connects the receiving and transmitting ends of a neuron [1]. These fibers are essentially the biological wiring of the nervous system and while they are microscopic in diameter (approx 1 mm), axons can be up to a meter in length [1].

The purpose of this project is to develop an algorithm for identifying and counting axons in the optical nerve of a mouse. The Case Western Reserve University research labs, under direction of Professor Howell, provided a series of reference images of the cross-sectional area of this nerve bundle for the purpose of developing and testing the algorithm. These color reference images have been prepared through staining and magnification in order to help the axons stand out against the background.

Historically, in order to gain an understanding of the number of rods and cones on the retina of the eye, cells had to be counted manually [2]. This process is met with a variety of complications such as the large number of cells contained on the surface, which can be on the order of hundreds of thousands and even millions [2]. Also, the curved surface of the back of the eye, and the ambiguity of the cell having dendrites appearing in multiple locations, further compounds the issue of simply counting by human inspection. For these reasons, analyzing the nerve bundle leading from the eye to the brain with a form of image processing is an obvious choice in lieu of the manual retinal identification process.

There exist numerous techniques for identifying objects of interest in a digital image, such as gray-scale thresholding [3,4], object recognition [5,6], contour modeling [2], and morphological operations such as thinning and edge detection [7]. Thresholding techniques for gray-scale images can be useful if the objects of interest have an intensity level that differs from the surrounding background information [2]. The axons in the images used for this project have intensity levels near that of the surrounding background, which makes the thresholding process more complicated than simply using an automatic threshold algorithm such as the Otsu technique [4].

These issues will be discussed with regard to the development of the algorithm, along with the strengths and weaknesses of the process.

II. ALGORITHM

A. Pre-processing Analysis

The images used in this project as test images need to undergo some pre-processing analysis, as would images in any application, in order to understand how best to extract the axon data. These images, for the part, are low contrast, low resolution, and contain fairly unsaturated colors. This poses a variety of complications that make some previously mentioned techniques less affective at extracting the desired image data.

Also, the axons themselves have a relatively darker border than the centers. The distribution ranges from densely packed regions to regions containing much background information. The sizes and shapes of the axons are not uniform, but are fairly elliptical. Identifying these properties of the base images is key to the proper implementation of the algorithm.

Other aspects of the images of particular note are the non-uniform backgrounds and the ambiguity of axon versus background information that arises from these aforementioned observations. These all play an important role in the functionality, and ultimate effectiveness, of the following steps of the algorithm.

B. Gray-scale Conversion

The original reference images are 24-bit truecolor; however, they have low color saturation. For this reason and the ease of morphological operations with gray-scale data, the reference images are converted using the MATLAB *rgb2gray* function.

C. Histogram Equalization

As discussed above, the base images are of low contrast; therefore, histogram equalization must take place to extend the range of the image intensity.

The Matlab *imhisteq* function is used to extend the range of intensities from a narrow band to the full range of 0 – 255 in order to enhance the lighter areas within the center of the axons. Equation 1 is a representation of the probability density function that *imhisteq* uses in which s and r are the number of output and input intensity levels, respectively [8].

$$s = T(r) = \int_0^r p_r(w)\, dw \tag{1}$$

D. Filtering

Next, a Gaussian spatial filter of size 3x3 with σ=0.5 is passed over the equalized image to remove noise from the image. The histogram equalization process produces many non-uniform regions of noise pixels, which could potentially cause issues in the following stages of the algorithm.

The Gaussian filter is based on Equation 2 below, where u and v dictate the size of the mask, and σ is the standard deviation.

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-\left(u^2+v^2\right)/\left(2\sigma^2\right)} \tag{2}$$

The size of the filter is chosen based on the pixel information in the input images. The size of the smallest axons is between 4 and 5 pixels in diameter, and the finer details of the edges of the axons should not be lost during the smoothing operation. Also, the input image must be padded prior to this smoothing operation; therefore the *'replicate'* parameter of the MATLAB function *imfilter* is used to prevent false object generation around the border of the images.

E. Binary Operations

Once the Gaussian filter has smoothed the image to remove noise, it can then be converted to a binary image through the use of the *im2bw* MATLAB function. Converting to binary image simplifies the detection and labeling of the axons (explained in the following sections). The critical aspect of the conversion process is properly choosing the threshold value. Based on analysis of the reference images, a threshold value of T=80% is chosen. Observing the given input images reveal that the center of the axon is of higher intensity than that of the edge of the

Essentially, all pixels with a value less than the threshold will be converted to 0 and those above to 1. As previously stated, automatic threshold algorithms that use density functions such as Otsu [4] do not produce consistently favorable results in this particular application. A more manual approach for deciphering the proper threshold value is needed for this algorithm. Choosing a value too low will introduce background pixels in the output, and choosing a value too high will not capture the smaller axons with lower intensities.

F. Morphological Operations

Next, a morphological dilation is performed. A circular structuring element of radius 1, Figure A, is created for this operation.

0	1	0
1	1	1
0	1	0

Figure 1.

Dilation "grows" the object by expanding the pixel regions through Equation 3 [8], where A is the base image and B is the structuring element.

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \tag{3}$$

This process serves to expand the pixel regions of the axons in order to merge any outlying pixels in the vicinity that may not have been removed from the smoothing phase. Any outlying pixels could create errors in the detection and labeling phase of the algorithm. In theory, using morphological erosion could remove these unwanted pixels; however, some axons are only a few pixels in diameter and could potentially be removed.

Pixels within a distance of 3 will merge after the dilation is complete. A potential side effect of this could be the merger of multiple axons, which is equally detrimental to the accuracy of the algorithm. Once again, careful preprocessing examination of the image data reveals that most axons are more than 3 pixels away from one another. Also, connectivity is taken into account in the final phase, which aids in minimizing errors.

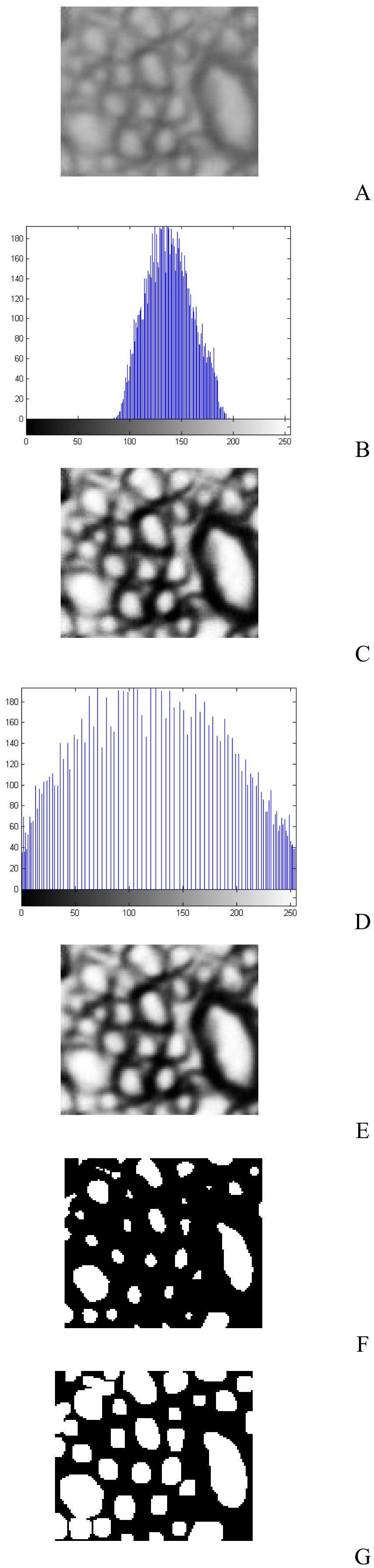
G. Identification and Labeling

The final phase of algorithm is to properly identify, count, and label the axons. Identifying and counting the axons is performed through the use of the MATLAB function, bwlabel. This function uses connectivity (in this application 4 connectivity is chosen) to differentiate pixel regions containing 1's. For this reason, properly removing outliers and pixel grouping is essential to the accuracy of this algorithm. The 4-connectivity parameter helps to alleviate instances where axons may be touching at 45° angles.

Bwlabel stores the [r,c] coordinate matrix for the location of each object (in this case, axons) and returns the total number of objects. This coordinate data can be passed to a labeling function [9] that places an asterisk at center of the axons in the original input image.

III. Results and Discussion

The algorithm operates with roughly 80% accuracy. This is determined by comparing a set of "gold standard" images, which are versions of the reference images that have had the axons pre-labeled, with the output images. An example that reflects the typical output is shown with all of the intermediate images. See Figure 2 below.



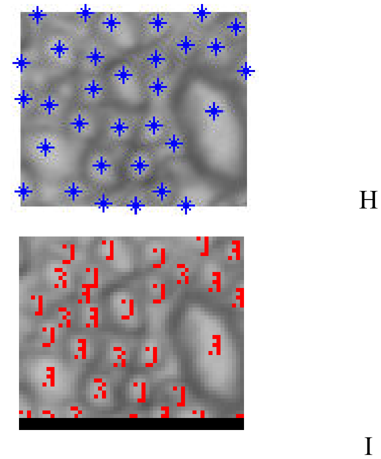


Figure 2: Example of algorithm. A) Gray-scale base image. B) Histogram of A. C) Histogram equalization. D) New histogram. E) Gaussian filter of C. F) Binary threshold of 80%. G) Dilation of F. H) Labeled output. I) "Gold Standard" image

In order for the algorithm to perform accurately, several factors have to be taken into consideration. First and foremost, the input (reference) images have to be analyzed with regard to resolution, contrast, background uniformity, and axon size. These factors define what size filter is needed, what threshold level is to be used, and also the nature of the structuring element used for the morphological operations.

In the above example of Figure 2, the image contains a large number of closely grouped axons. The axons have fairly equivalent intensity levels at their centers, and the smallest of these is greater than 4 pixels in diameter. Additionally, since the axons a relatively tightly packed, the non-uniformity of the background is negligible. The algorithm is best suited for this collection of parameters.

Several other techniques were tested before defining the current algorithm. The first process tested was the enhancement of the foreground information and/or removal of the background information. It is difficult to differentiate axons from the background with regards to the size, shape and contrast. A morphological "top-hat" function [8] was tested in an attempt to improve the uniformity of the background so that the foreground and background shading would become less ambiguous. The "top-hat" operation, through the MATLAB function *bwmorph,* produced unfavorable results in this particular application. A possible reason for this could be that the edges of the axons are not sharply defined, thus making the top-hat function less effective. Overall, the most effective method found was histogram equalization. This was able to push the intensity levels of the centers of the axons to a higher value, while pulling the darker banding around the edges lower.

Edge finding was another type of technique attempted to better define the axons. Sobel and Canny methods [8] were tested on the image after the histogram equalization had taken place to optimize the effectiveness of the algorithm. These did not prove to be valuable, however, in detecting the edges of the axons without also capturing the elliptically shaped background areas as well. Several attempts were made in adjusting the threshold of the edging finding algorithms, but the axon and background edges could not be separated consistently. This is mainly due to the fact that the lighter areas of the background form shapes similar to that of the axons.

This shape similarity also poses a problem for another technique: object recognition. The axons themselves vary greatly in size and shape, thus making pattern recognition [6] or contour modeling/matching [2] conceivably difficult. Also, background area between the axons is of similar size and shape, which introduces even more difficulties for an algorithm to discern axons from non-axons.

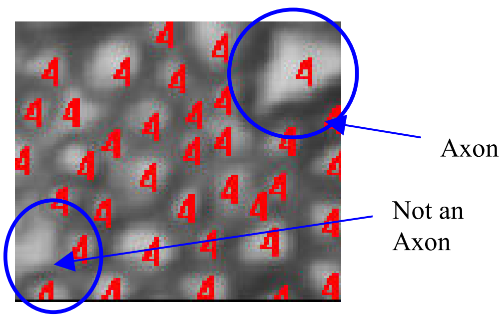


Figure 3: “Gold Standard” Example of an object region classified as an Axon versus a region not. Notice the similarities the two.

Thresholding of the enhanced gray-scale image was also examined greatly. The Otsu technique was attempted first to reduce most of the background information to a zero value while pronouncing the axon information. The Otsu technique proved ineffective in this application due to the above stated reasons regarding similarities of intensity levels. This issue of foreground/background ambiguity is apparent in Figure 4 below.

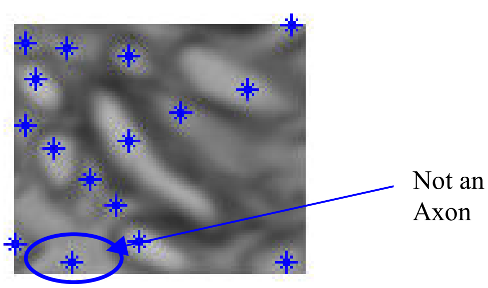


Figure 4: Background areas incorrectly labeled as Axons. Threshold of T=80% used.

The algorithm is robust enough to achieve good results with various “gold standard” reference images, which contain differences previously discussed. More examples of the processed images compared to their respective “Gold Standard” images can be found in Figure 6, and ways to improve these results are discussed in Section IV.

IV. Areas for Improvement

A. Gray-scale Threshold algorithm

The greatest challenge in developing this algorithm was the optimization of the threshold value used to convert the enhanced gray-scale image to a binary image. Care must be taken to have the optimum threshold value chosen; else the algorithm can give rise to false labeling.

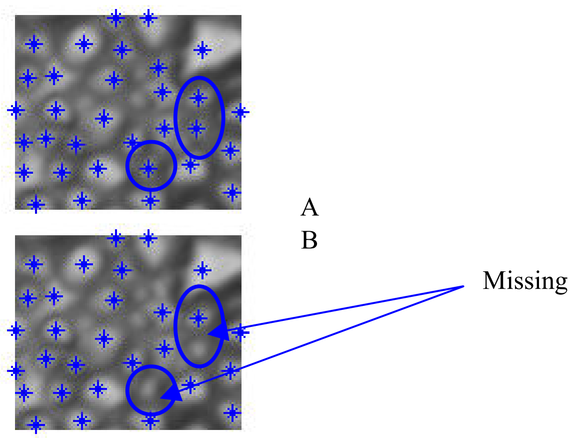


Figure 5: A) Threshold T=75%. B) Threshold T=80%.

B. Additional Morphological Operations

It is hypothesized that threshold algorithms like Otsu or morphological techniques like Top-hat equalization may be more effective if applied to a smaller region of the image. Experimentation into region-based segmentation is an area that should be explored to optimize the effectiveness of the current algorithm. This may allow the threshold algorithms to produce better results than when applied to a larger image.

V. Conclusion

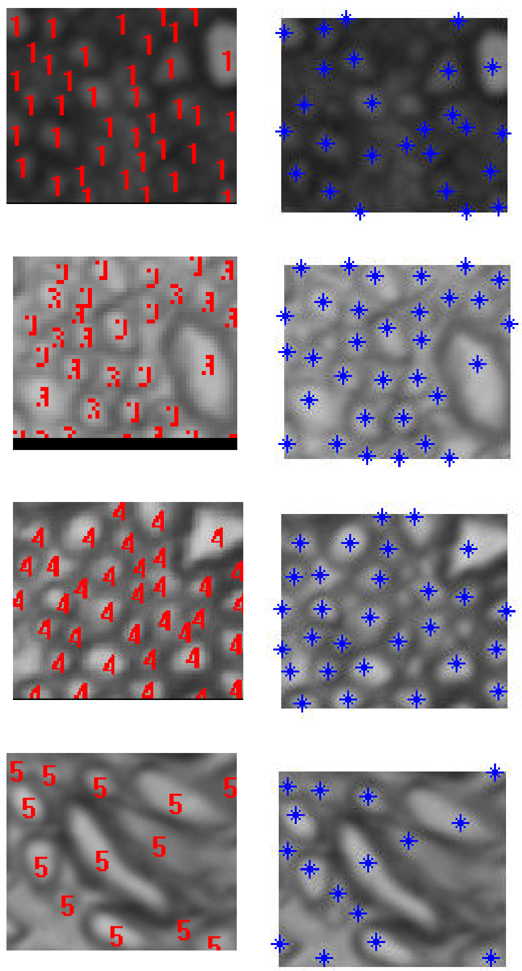
The morphological algorithm for locating and counting axons developed for this application performs with approximately 80% accuracy. Additions to this algorithm [2]-[5] could conceivably in increase this percentage and create an even more robust design; however, this could potentially slow the analysis process greatly when operating upon very large images. Much learning was gained through the development of this project, and the key takeaways can be applied to future endeavors as well as improvements to this algorithm.

Appendix

Appendix A: M-file code.

Acknowledgment

J. Creighton would like to thank Dr. F. Merat for his guidance, patience, and supportive role in this project, and Professor Howell the reference images.



A B

C D

E F

G H

Figure 6: A,C,E,G) Labeled Gold Standard. B,D,F,H) Process Output.

References

[1] H. Gray, Anatomy, 15th ed, Barnes & Nobel Books, NY, 1995 pp. 43.

[2] Y.L. Fok, J.C.K Chan, and R.T. Chin, "Automated Analysis of Nerve-Cell Images Using Active Contour Models," *IEEE Trans. Medical Imaging*, vol. 15, no. 3, pp. 353-368, June 1996.

[3] H.J. Jeong, T.Y. Kim, H.G. Hwang, and H.J. Choi, "Comparison of Thresholding Methods for Breast Tumor Cell Segmentation," 2005. *IEEE 01500489*.

[4] I. Sekita, T. Kurita, N. Otsu, "A Threshold Method Using the Mixture of Normal Density Functions," *International Symposium on Speech, Image Processing and Neural Networks*, Hong Kong, April 1994.

[5] I. Sekita, T. Kurita, and N. Otsu, "Complex Autoregressive Model for Shape Recoginition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 4, pp. 489-496, April 1992.

[6] W.N. Lie, H.C. Hsieh, "Lips Detection by Morphological Image Processing," in ICSP 1998. *IEEE 00770804.*

[7] V. Espinosa, "Mathematical Morphology approaches for Fingerprints Thinning," IEEE ICCST, 2001. *IEEE 0104224*.

[8] Gonzalez, R. and Woods, R., Digital Image Processing. Prentice-Hall, Upper Saddle River, NJ, 3rd Ed. 2007.

[9] S. Eddins, "Gray scale pixel values in labeled regions," MATLAB Central, Aug. 21, 2007. <http://blogs.mathworks.com/steve/2007/08/21/gray-scale-pixel-values-in-labeled-regions/>