# Automated Axon Counting in Normal Optic Nerves Using Image Morphology and Area Based Segmentation

William F. Lipstreu

Department of Electrical Engineering and Computer Science

Case Western Reserve University, Cleveland, OH, USA e-mail: wfl2@cwru.edu

## Abstract

This paper presents an application of image morphology and segmentation in the field of medical imaging for the use of ascertaining the number of axons from cross section images of an optic nerve. Because of the complexity of the nerve and inconsistency of individual axon sizes and shapes, a general approach is conceived to provide an accurate estimation of the number of axons present. By taking into account image statistics and evaluating regions according to their characteristics, results can be obtained that count axons to within a few percent of the actual number present.

## KEYWORDS

Axon Counting, Normal Optic Nerve, Image Processing, Computer Vision, Segmentation, Morphology

## INTRODUCTION

Axons are fiber-like extensions of the nerve cell through which electrical impulses propagate in the nervous system. Nerve cells (neurons) are composed of bundles of these axons. They are similar to fiber optic cables and carry outgoing messages.

Currently there is research being conducted in the area of retinal sensing in mice, and it is necessary to count the number of rods and cones. Unfortunately, the curved retina makes it problematic to count these sensors. As an alternate solution, it is possible to take a nerve bundle cross section and analyze the number of axons present. Professor Howell at Case Western Reserve University is cutting optical nerve bundles from mice which connect the retinal sensors to the brain. He creates microscope images of cross sections of these nerve bundles after staining the nerve to enhance the neural connections (axons). Presently, axons are counted manually by specialists who are experienced at properly identifying the fibers.

There are many groups developing automated cell counting systems to speed up the process and produce more precise results. One such group is counting Erythrocytes (red blood cells) to aid in determining the presence of certain diseases. The synthesis of their algorithm involves the first step of image segmentation in which a histogram is created and a suitable threshold is selected. Next, image conditioning provides for hole-filling and removing borders or unwanted particles. Finally, the image analysis step counts the Erythrocytes [6].

Another group is currently using a technique that involves a hybrid (raster and vector) algorithm that separates the objects of interest (axons) from the background. This algorithm uses edge detection to create an edge map which is then optimally thresholded with a Fuzzy C-Means method. After thoroughly skeletonizing the image, the outline of all closed shapes allows the program to decide whether or not the shape is an axon. The classification is based on parameters such as size, intensity, grouping characteristics and shape complexity. The results of this research are frequently within the 95% confidence interval but tend to count more axons than are actually present [1].

A recent endeavor in tumor cell counting utilizes a robust local adaptive thresholding method to segment regions of interest from the background. Such thresholding improves results in feature extraction for tumor cell identification [4].

Still further efforts in the biomedical applications of digital image processing include research on counting cancerous cells in a tissue with breast cancer. The concept presented in this paper will closely resemble the scheme devised by this Thai research team. In their approach, local adaptive thresholding is applied after noise removal. Next morphological operations assist in categorizing the cells by size and obtain a practical cell count of cancer cells. To further ensure this result, a watershed segmentation operation is performed. This solution produced outcomes comparable to the manual counts of specialists [5].

Axons are typically one micrometer in diameter and require the use of a microscope to see. They are found in clusters and range in sizes that include extremely small and large axons. Using digital image processing, several morphological techniques are explored to create regions that allegedly represent these individual axons. Given sample images of nerve bundles, it is possible to count the individual axons manually and write a program to achieve a similar count. The goal is to write an algorithm which locates and counts these sample axons with reasonable accuracy so that the program can be used on an entire cell image containing tens of thousands of axons.

This problem demands a solution that is able to count only objects of interest. A well known and widely used way to extract image components is morphology. This type of image processing relies on mathematical morphology which is founded on set theory. Morphological operations typically deal with the processing of binary images to rep-

resent the ordering of pixels in an n-dimensional Euclidean space (as opposed to numerical value) [2].

Structuring elements are relied on to search the image for certain properties. These subimages help assess images based on which pixels are members of its matrix and which are not. Many useful tools such as erosion, dilation, opening, closing, skeletonization, and pruning rely on such structuring elements. The careful design of an appropriate structuring element can be extremely effective when attempting to extract axons from an image, especially when used on an optimally thresholded image.

## IMAGE ANALYSIS AND PREPROCESSING

To implement some of the aforementioned morphological techniques, it is necessary to first analyze and preprocess the image. First, histograms of the various sample images are generated to view the distribution of grayscale intensities in the images. The color image is separated into its three separate layers: red, green and blue. Histograms are created for each layer to determine if an individual layer has more useful characteristics than the others. Characteristics such as high contrast make the task of thresholding significantly easier.

Upon observation, the red and green layer distributions are practically identical, and the blue layer has a wider distribution but not enough to be more useful. Therefore, the conclusion is to simply convert the RGB image to grayscale and perform all operations on a single layer. A cross section of a few hundred axons in grayscale is used to generate the histogram in Figure 1.

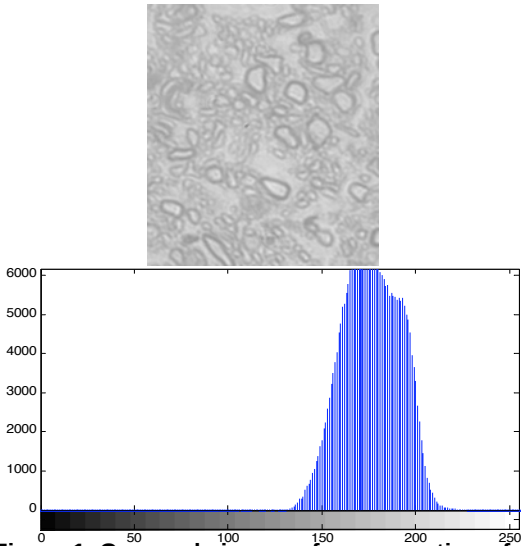


**Figure 1. Grayscale image of cross section of a few hundred axons and its corresponding histogram**

Analyzing Figure 1 reveals that there is not a very wide dynamic range of intensity values. In an attempt to adjust the distribution and make it more favorable to thresholding by increasing the contrast, histogram equalization was used (Figure 2). However, the resulting histogram had adverse effects on the computational portion of the algorithm. Therefore, the above histogram is suitable for calculating the best threshold.

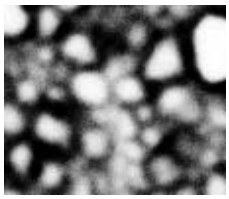

**Figure 2. Histogram equalization of image**

## THRESHOLD DETERMINATION

The thresholding step is the most crucial part of the algorithm. Even the best decision tree will be useless if the input image is missing axons that are erroneously thresholded out. Three methods of thresholding were attempted to optimize the threshold value for creating a binary image. The Otsu method was first used for its renowned ability to find the optimum threshold value based on minimizing the intraclass variance of the black and white pixels. Next, the Otsu values were compared to an iterative algorithm conceived by Gonzalez and Woods which produced almost identical results [2]. Finally, the threshold was computed by taking a simple average of the intensity data. Using all three methods on six separate and unique sample images, the following results were obtained in Table 1.

**Table 1. Table of threshold values obtained from three different methods**

| **Threshold Value** | Otsu's Method | Iterative Threshold | Average Threshold |
|---|---|---|---|
| Sample 1 | .2627 | .2709 | .2183 |
| Sample 2 | .5451 | .5447 | .5813 |
| Sample 3 | .5451 | .5498 | .5426 |
| Sample 4 | .4824 | .4863 | .4515 |
| Sample 5 | .4549 | .4589 | .4391 |
| Sample 6 | .4510 | .4503 | .4435 |

Notice in Table 1 that Otsu's Method and Gonzalez and Wood's iterative method are very close, whereas the average threshold is typically 5-20% less. These results were essential to properly assign the appropriate threshold to individual images.

In the algorithm, many methods of computing the threshold are involved to provide an optimum range of data points that include excessively faint objects. First the Otsu method is used to glean information about the image's brightness. Using *graythresh.m* defined in MATLAB's library, a value is returned. If it is below .4 (namely images like Sample 1 with values of .26), additional preprocessing must take place to brighten the image. This is accomplished with *adapthisteq.m* defined in MATLAB's image processing toolbox. This function is useful because it operates on regions of the image rather than treating the image as a homogenous grouping of objects. Benefits of this technique include its ability to enhance the contrast of regions of the image without amplifying any noise present.

Figure 3 displays the result of applying an adaptive histogram to a dark sample image (Sampel_1.tif).

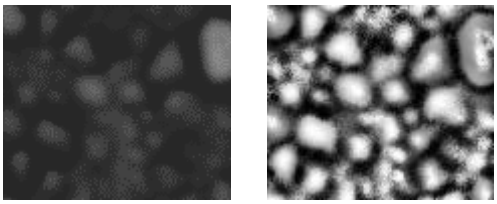

**Figure 3. Dark image of axons and its corresponding image after application of adaptive histogram**

Images that exceed the Otsu threshold value will skip the adaptive histogram step and go onto the iterative thresholding method to produce a value to send to the function *im2bw.m* function which will convert the grayscale image to black and white (binary) based on the supplied threshold (Figure 4).

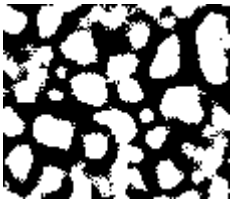

**Figure 4. Binary thresholded image before morphological corrections**

## MORPHOLOGICAL PROCESSING

The resulting binary image will contain clusters of white pixels and a few stray pixels. Before applying morphological techniques to improve the connectivity of regions, an area-based analysis is performed to gain information about the nature of the axon sizes. It is not useful to use large scale erosion and dilation techniques on images in which the average size of an axon is only 20-50 pixels. Such an oversight would lead to the elimination of many smaller sized axons.

To accommodate images with clusters of unusually small axons, MATLAB's command *bwlabel.m* is used to create matrices for connected regions. While this will later be used as the primary counting tool, it is initially used to assign labels to connected regions. Each label has region properties that can be accessed from their stored structural arrays. One of the region properties available to users is the *area* property. Each binary image is sent through *bwlabel.m* to compute the minimum, maximum, and average areas of the axons present. The resulting areas of the six sample images are presented in Table 2.

**Table 2. Table of minimum, maximum, and average binary region areas (axon areas in pixels)**

| **Area of Cells** | Smallest Area | Largest Area | Average Area |
|---|---|---|---|
| Sample 1 | 4 | 1261 | 119.2222 |
| Sample 2 | 103 | 3725 | 976.8333 |
| Sample 3 | 5 | 787 | 169.5 |
| Sample 4 | 14 | 457 | 125.5714 |
| Sample 5 | 6 | 1097 | 258.75 |
| Sample 6 | 7 | 3029 | 181.3333 |

Table 2 provides useful data for interpreting the sizes of axons. Sample images with very large average areas have an excessive amount of empty space. No axons were found to be greater than 1000 pixels in area. Therefore, this was set as the global threshold for counting criteria. It can be inferred from the table that samples with rather low average areas will contain a lot of delicate data that should be dealt with using more moderate morphology techniques.

Once it is determined whether an image has large or small groupings, it will be passed onto the morphological processing. Small groupings will be passed to a series of operations that fill the holes of the binary image. Erroneous pixels and outliers will be eliminated in this stage. Next, an area opening is performed to remove connected components that contain fewer than 5 pixels. Finally, the structuring element sizes are set for the next step of eroding and dilating. This step is crucial to properly separate the individual regions.

For the images that have larger average areas of their regions, they are sent immediately to a more aggressive erosion and dilation procedure. More specifically, a 1-pixel 'disk' structuring element is used to erode away irrelevant pixels followed by a 4-pixel 'disk' structuring element used to dilate the remaining regions. Figure 5 shows the image after this morphology stage.

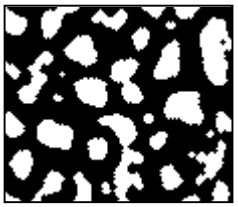

**Figure 5. Binary thresholded image after morphological corrections**

Notice that small regions were preserved in Figure 5, but some larger regions remained connected which will feed the counting stage of the algorithm with some false data.

## AXON COUNTING

After the morphology has cleaned the image up and reduced it to separate regions, the binary image is passed to *bwlabel.m* again to count the connected regions with 4-point connectivity. The function will return the number of regions present in the image, but it has not taken into account the "empty space." Therefore, a final correction is implemented by subtracting the total number of regions with an area greater than 1050 pixels from the final count. Many images will have anywhere from 1-5 areas that are too big to be axons. This check is a safe way to avoid counting the background.

## RESULTS AND DISCUSSION

This method of axon counting achieved reasonably acceptable results. Table 3 summarizes the experimental results

of the described algorithm. The hand count column represents the number of axons in the image according to the manual count of a medical expert. The automated count column tabulates the number of axons determined to exist by the computer. The False Areas column indicates how many regions were found to be too large to be an axon and therefore classified as the image background. Finally, the % error column displays the percentage deviation of the experimental value from the actual value.

**Table 3. Table of results comparing actual and experimental numbers of axons present in six sample images**

| # of Axons | Hand Count | Automated Count | False Areas | % error |
|---|---|---|---|---|
| Sample 1 | 40 | 35 | 0 | 12.5 |
| Sample 2 | 4 | 5 | 1 | 25 |
| Sample 3 | 32 | 28 | 1 | 12.5 |
| Sample 4 | 37 | 36 | 0 | 2.7 |
| Sample 5 | 13 | 15 | 2 | 15.38 |
| Sample 6 | 33 | 28 | 1 | 15.15 |

Figure 6 is an example of a low contrast image with smaller axons that received a more gentle morphological procedure than other images. While only 35 of 40 axons were counted, it is evident that all regions were identified correctly but were not able to be separated. To achieve this separation may jeopardize the integrity of smaller axons. With a 12.5% error, this image represents a fairly successful trial.

**Figure 6. Sample_1.tif – original image, processed regions, and official counted image**

Figure 7 is a less complex image with four large axons present. The algorithm counted 6 regions and correctly removed the large area that constitutes empty space, reducing the count to 5 axons. The correct number is 4, but it appears that the corner of the image has interfered with the count. The corner is surrounded by thick edges which make it look like an axon and therefore produced a false positive. This is a shortcoming in the algorithm.

**Figure 7. Sample_2.tif – original image, processed regions, and official counted image**

Figure 8 is an example of axons with axon edges that are somewhat blurry and less defined. The right side of the image segments well, but the left side is plagued by false connections, bringing the final count to 28, which is 4 less than the actual count. Also, the algorithm subtracted one from the count because it thought the connected cells were part of the background. This is an unintended consequence of failing to distinguish the axons.

**Figure 8. Sample_3.tif – original image, processed regions, and official counted image**

Figure 9 is the most successful of the sample images. The thick and clearly defined edges of the axons allow the algorithm to operate almost perfectly. The automated count is 36, which is 2.7% off of the target, 37 axons.

**Figure 9. Sample_4.tif – original image, processed regions, and official counted image**

Figure 10 counted too many regions (15 instead of 13). This is due to some background area that was not as large as usual and therefore appeared as an axon to the algorithm. Otherwise, the segmentation performed reasonably well.

**Figure 10. Sample_5.tif – original image, processed regions, and official counted image**

Figure 11 counted successfully where there is a large and distinct distribution of axons. Toward the left of the image the axons begin to group into an almost indistinguishable blur of small clusters. The algorithm correctly identified and subtracted the background but was unable to separate the large space from some of the small neighboring axons.

The resulting count was 5 less than the known number of 33.

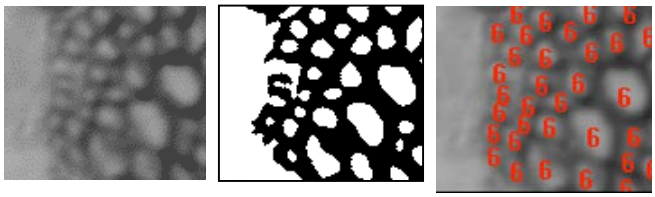

**Figure 11. Sample_6.tif – original image, processed regions, and official counted image**

The next two sets of images are not "gold standard" images. Rather, these images were chosen at random from the larger nerve bundle and used as test images. Figures 12 and 13 demonstrate the algorithm's effectiveness when used in a practical situation. Figure 12 identifies the most obvious axons but leaves out some smaller axons in between the larger ones. Figure 13 also suffered from this problem.

**Figure 12. Sample_7.tif – original image and processed regions**

**Figure 13. Sample_8.tif – original image, processed regions**

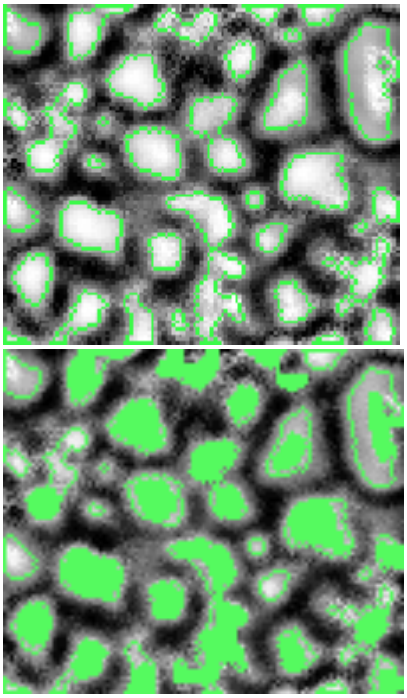

**Figure 14. Image Overlay**

Figure 14 utilizes an algorithm provided by the Mathworks website as a step in the watershed process [3]. This image serves to demonstrate what the computer sees in the algorithm. With the edges overlaid on the original image, it is clear where the computer has trouble distinguishing axons.

Some research endeavors have gone to great lengths to extract axons based on contours (or snakes) to more accurately cater to the physical appearance of axons. Even with such complex optimization schemes, the false detection of axons is still unavoidable. Sometimes insufficient image quality and unusual cell features contribute to these errors [7]. For the time being, it seems that a small margin of false detections will have to be acceptable to the biomedical community until a better feature recognition system is developed.

## SUMMARY

Knowing the number of cells *a priori* provided a useful means to test the algorithm against actual data. The main problems encountered included the lack of contrast and the highly-varied shape of the axons. Therefore, a uniform morphological technique cannot be applied to the image with the expectation that all axons will be included. Most previous research efforts encountered similar problems.

In a project to count hepatocytes in medical images, results were plagued by low contrast, uneven illumination, irregular cell shapes, and gray intensity variety. The team used a similar three stage image processing scheme to assess the problem: image conditioning, segmentation, and morphology. A local adaptive thresholding technique was em-

ployed in the segmentation stage cell count and produced an 85% success rate [8].

Fortunately, processing power is not an issue in this problem. Any amount of code complexity is acceptable. Some attempts at noise removal and top-hat operations were made but achieved little utility. The end result was to use basic morphological operations such as dilation to expand boundaries and erosion to contract boundaries. Contours were smoothed with opening operations and islands and sharp peaks were removed.

## CONCLUSION

This paper discussed a unique approach that involves image morphology with a strong emphasis on region areas. Although the results are not revolutionary, they are certainly tolerable compared to what previous endeavors have produced. By using adaptive thresholding techniques and treating portions of the image according to their respective characteristics, axon counting can become more accurate.

Pruning the six sample images to perfection produces disastrous results when new test images are generated. A custom algorithm cannot be devised to search for objects that have no defined or consistent shape, size, and brightness. Threshold values are sensitive, and one or two pixels can make the difference of counting 5-10 axons in some clustered images.

This algorithm is scalable and could be modified to analyze the entire nerve in segments. There would need to be a section of code written to tally all segments for the final count.

It is important to acknowledge the limitations of this algorithm and seek future investigations into characterizing the shapes of the axons. Although the edges of the axons are not well-defined or predictable, an edge-based detection algorithm should be implemented to help separate clusters in areas where there is less certainty about the object's classification. This would improve results significantly.

## ACKNOWLEDGMENTS

This work was done in support of Professor Howell's research at Case Western Reserve University.

## REFERENCES

[1] J.Reynaud, G.Cull, L.Wang, C.F. Burgoyne, and G.A. Cioffi., "A New Hybrid Algorithm for Automated Axon Counting in Normal Optic Nerves," presented at Devers Eye Institute on Non-clinical Image Analysis, 2007.

[2] Gonzalez, R.C. and Woods, R.E., *Digital Image Processing 3ed*, Prentice-Hall, Inc, Upper Saddle River, New Jersey, USA, 2008.

[3] S. Eddins, "Image Processing: Cell Segmentation," The Mathworks MATLAB Central File Exchange, 2006.

[4] B. Fang, W. Hsu, and M. Lee, "Tumor Cell Identification Using Feature Rules," Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[5] P. Phukpattaranont and P. Boonyaphiphat, "An Automatic Cell Counting Method for a Microscopic Tissue Image from Breast Cancer," presented at 3rd Kuala Lumpur International Conference on Biomedical Engineering, 2006.

[6] T Ng and A. See, "Automated Computer Controlled Erythrocytes Counting System," Monash University Malaysia, National Instruments Life Science and Research, 2005.

[7] F. Ying-Lun, J. Chan, and R. Chin, "Automated Analysis of Nerve Cell Images Using Active Contour Models," presented in IEEE Transactions on Medical Imaging, 1996.

[8] H. Refai, L. Li, T. Teague, and R. Naukam, "Automatic Count of Hepatocytes in Microscopic Images," presented in Proceedings, International Conference on Image Processing: Digital Object Identifier, 2003.