Automated Biological Image Segmentation

Aaron Franczyk

Department of Electrical Engineering and Computer Science,
Case Western Reserve University, Cleveland, OH, Email: aaron.franczyk@case.edu

ABSTRACT

This paper presents a method for automated identification and counting of neural connections (axons) in a microscope image of an optical nerve bundle. The algorithm uses morphological image processing techniques, coupled with traditional image segmentation methods, to generate a binary image suitable for counting features. Results from gold-standard images show the algorithm accuracy to be highly dependent on both the average size of axons and image quality (i.e. sharpness, contrast and resolution). Input images containing several axons of similar size show the algorithm to be reasonably accurate and robust. The algorithm breaks-down when processing grainy, pixilated nerve bundles containing few, large axons.

KEYWORDS

Biological Image Segmentation, Retinal Sensing, Optical Nerve Bundle, Axon Counting, Morphological Image Processing

INTRODUCTION

Automatic segmentation of biological images is a highly studied topic, driven by the desire for quantitative or high throughput analysis of biological features in dense images.

In this study, researchers investigating retinal sensing are interested in counting the rods and cones on the retina of mice. The curved geometry of the retina makes it difficult to count these sensors in a reliable, efficient manner. An alternative approach is to cut the optical nerve bundle and count the neural connections (axons). Microscope images can be stained to enhance the visual appearance of the closed bundles. In these images, the background intensity varies at different locations. Axons can be tightly clustered, and of varying size, making it difficult for researchers to visually determine the boundaries between axons and background noise.

Traditional image segmentation methods such as thresholding, morphological operations, edge-based approaches and region-based processing have been successfully applied to many applications (refer to [1] as review).

Revolutionary thresholding techniques using the Otsu algorithm [2–3] are often used for automatic segmentation problems. Otsu [3] selects a threshold automatically from a gray level histogram using discriminant analysis, dealing directly with the “goodness” of thresholds. An optimal threshold is selected by maximizing the measure of separability of the resultant classes in gray levels.

Advanced segmentation methods using the watershed transform have been demonstrated to be more suitable for biological image processing [4–5]. Watershed transform considers the image as a topographic relief, with the height of each point directly related to its gray level. Consider rain falling on this “terrain”, so the watersheds are the lines that separate the catchment basins that form. The transform is generally computed on the gradient of the original image, so that the basin boundaries are located at high gradient points.

However, when applying either traditional image segmentation techniques or the watershed transform to non-trivial applications, issues of noise sensitivity, over-segmentation, overlapping features, and discretion between features of interest and background features all become problematic.

The algorithm design will incorporate traditional image processing “building-blocks” to process and count axons in the images. In parallel the software explores the advanced methods of Otsu thresholding and watershed transform, allowing the programmer to visually select a “best method”. Algorithm performance will be judged not only on its ability to count axons, but also robustness against these issues.

DEVELOPMENT OF THE ALGORITHM

To accurately identify and count the axons, several of the discussed methods were applied. To manage the problem, several axon image a priori assumptions were determined: partial axon bundles along the image border would be considered, and if the programmer could not distinguish an image feature to be that of an axon or background noise, then it is unreasonable to expect the algorithm to do so. The algorithm (Figure 1) contains many of the standard building blocks of traditional image segmentation.

The image was first filtered using a Gaussian low-pass filter to provide a smoothing effect and remove any high-frequency noise. Notice the low-contrast of a typical image and its resulting histogram (shown in Figure 2). The algorithm uses histogram equalization to prepare the image for further filtering and thresholding (resulting image and histogram shown in Figure 3). Next the grayscale image undergoes a top-hat transformation to enhance the detail in the shading [6]. This transformation subtracts an opening from the image (Equation 1).

$$h = f - (f \circ b) \tag{1}$$

Simply, the top-hat transformation is the erosion of f by b, followed by the dilation of the result by b, subtracted from the image. The “disk” structuring element was chosen due to the round geometry of the axons.

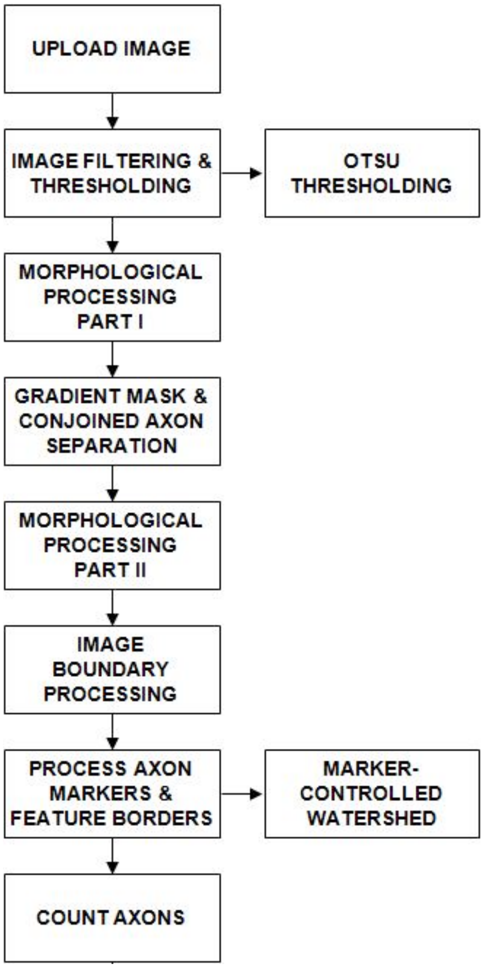
UPLOAD IMAGE
IMAGE FILTERING & THRESHOLDING
OTSU THRESHOLDING
MORPHOLOGICAL PROCESSING PART I
GRADIENT MASK & CONJOINED AXON SEPARATION
MORPHOLOGICAL PROCESSING PART II
IMAGE BOUNDARY PROCESSING
PROCESS AXON MARKERS & FEATURE BORDERS
MARKER-CONTROLLED WATERSHED
COUNT AXONS

Figure 1. Algorithm Functional Block Diagram

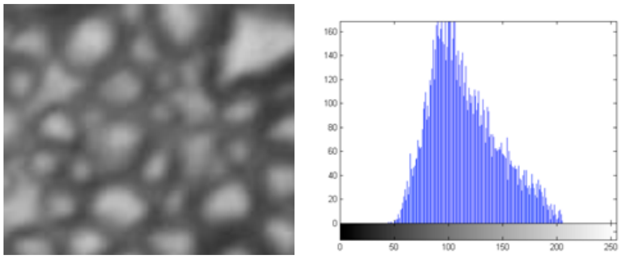

Figure 2 (a-b). Original Image – Test Image #4

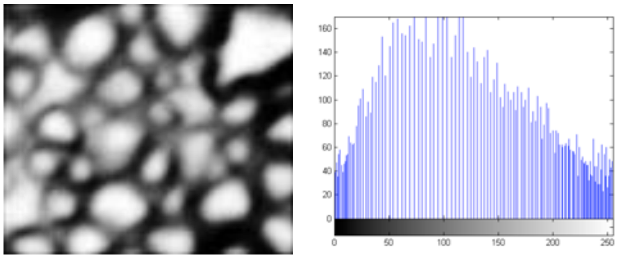

Figure 3 (a-b). Histogram Equalization – Test Image #4

It should be noted that the Otsu thresholding method [3] was investigated as the algorithm evolved. However, when compared side-by-side with the top-hat result, the Otsu method was abandoned due to inferior performance in this particular application.

Thresholding completed, the image is converted to a binary image to begin identification of "blobs of interest" which will eventually be down-selected as axons. The result undergoes the first battery of morphological image processing to identify blobs. Image "spurs" are removed using a series of erosions and dilations that are handily pre-packaged as a morphological operation (refer to [6] to review MATLAB® image processing toolbox). Likewise, the result undergoes a "clean" operation in which isolated pixels are removed.

The gradient magnitude is used to process the image for use as a mask. The gradient magnitude image has high pixel values along object edges, and low pixel values elsewhere. The Sobel edge operator is used during this operation, as recommended for being robust [6]. The resulting gradient image is "thinned" and then "opened" to achieve a masking of intended "blob boundaries". The "blobs of interest" are then combined with the boundary masking to separate the conjoined axons. (See Figure 4.a with conjoined blobs of interest and Figure 4.b with resulting separation.)

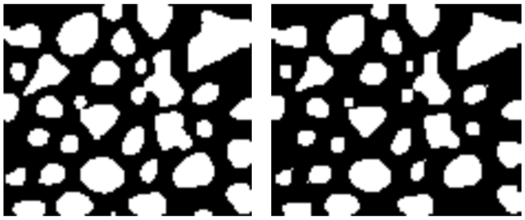

Figure 4 (a-b). Blob Separation – Test Image #4

The image then undergoes a second battery of morphological image processing. Spurs, created during the masking, are removed. The image is processed to fill around diagonally connected pixels. Finally, the image is "cleaned" of rogue pixels.

In anticipation for creating borders around each blob [7], the image is processed to create a white border around the image perimeter with a depth of one pixel.

The resulting image is shrunk to form the blob (axon) borders as showing in Figure 5.a. Likewise, the compliment is shrunk to create an axon "seed pixel" to be used as markers for the watershed transform (see Figure 5.b).

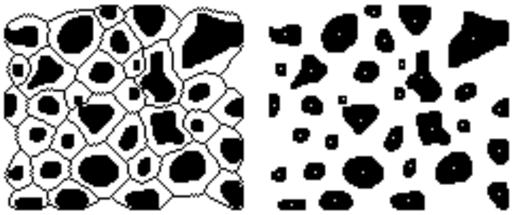

Figure 5 (a-b). Axon Borders, Markers – Test Image #4

Using the defined axon seeds and boundaries (internal and external markers, respectively), the algorithm investigates the performance of the marker-controlled watershed transform due to its popularity in the fields of biology and medicine [4-5]. However, when compared side-by-side with the results achieved using the methods depicted in Figure 5, the watershed transform was set aside as it visually offered no additional benefit to the simple border-creation.

The algorithm takes each identified blob and counts it as an axon. Graphical results are processed and output for manual identification of false-positives and false-negatives.

RESULTS AND DISCUSSION

Expert researchers supplied six "test images" (inputs for the algorithm) with accompanying "gold-standard" images that have identified axons.

It is important for the algorithm not only to output a numerical result, but also that the numerical result represents actual axons and not other image features. Therefore, when discussing results it is critical to examine the amount of false-positives and false-negatives. False-positives are defined to be image features identified as axons by the algorithm, but not listed on the gold-standard image. False-negatives are defined to be instances where the algorithm fails to identify an axon that has been listed on the gold-standard image.

Table 1. Quantized Results – All Test Images

Test Imag	Gold Std Axo [Goal]	Algorithm Co [Result]	False Positive	False Negativ
1	40	31	4 13%	13 33%
2	5	19	14 74%	0 0%
3	32	35	5 14%	2 6%
4	36	33	1 3%	4 11%
5	13	21	8 38%	0 0%
6	33	23	0 0%	10 30%

As evident from the results listed in Table 1, the algorithm's performance varied greatly from image to image.

In general, the algorithm worked well on images with tight clusters of clearly defined axons having similar size radii (i.e. test images 3, 4).

The algorithm falls apart with images containing large voids between the axons (i.e. test images 2, 5, 6) and dark contrast throughout (i.e. test image 1). These differences drive varying results in the thresholding of the original images, which leads to inconsistent algorithm results.

For example, Figure 6 shows the results for the algorithm's best result, test image #4. (Figure 6.a is the original image. Figures 6.b, 6.c, 6.d, show the axon boundaries, counted axons, and the combination of both, all overlaid with the original image, respectively.) Notice the boundaries are accurately identified around most axon bundles. The single false-positive result is labeled by the yellow circle (note: this was manually placed for discussion purposes only). False-negatives are labeled by green circles. Two of the four false-negatives are sites where two axons were erroneously identified as a single blob. The remaining two were small axons residing on the image boundary.

Figure 7 depicts the algorithm's worst result, test image #2. (Again, false-positives identified by a yellow circle.) While the algorithm did identify all five axons, it also identified fourteen false-positives. This poor result originates back to the image output from the histogram equalization step. Due to the large surface area of voids (empty background) the histogram equalization resulted in a highly-contoured grayscale image that did not respond favorably to the thresholding and subsequent binary conversion. The texture in the background ultimately produced erroneous "phantom axons" that the algorithm processed and counted.

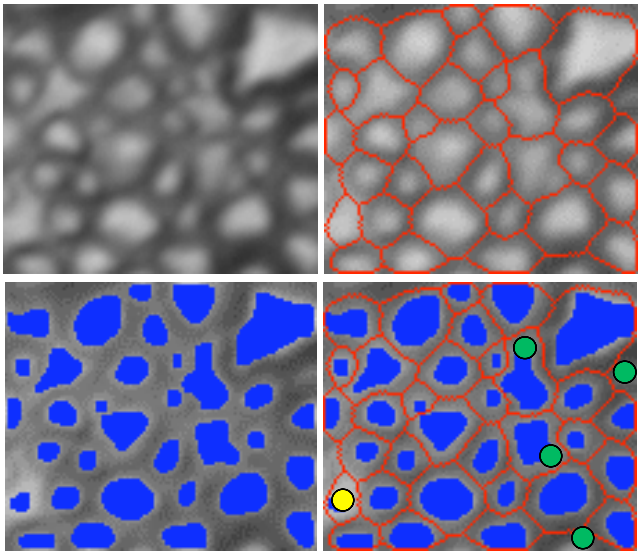

Figure 6 (a-d). Final Results (Best) – Test Image #4

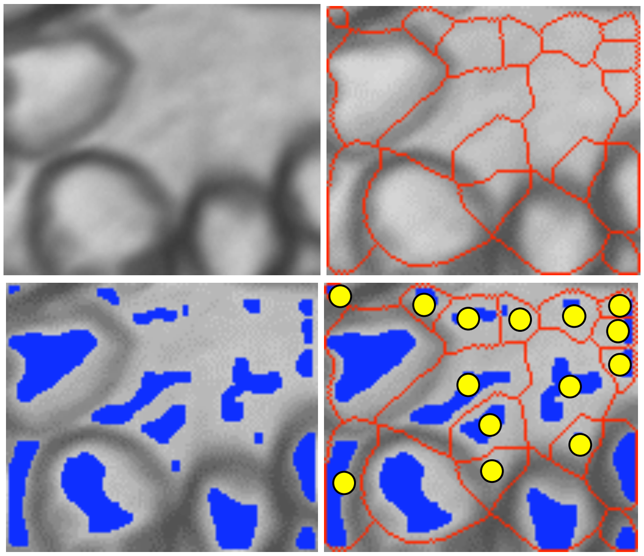

Figure 7 (a-d). Final Results (Worst) – Test Image #2

Figure 8 depicts the algorithm's results for test image #6. (Again, false-negatives identified by a green circle.) While the algorithm did not identify any false-positives, it also failed to identify ten axons that were identified in the gold-standard images. This image was seemingly taken with a given perspective of a curved-section of the nerve bundle. Notice the algorithm falls apart on the left boundary of the bundles, where the void (empty background) is present.

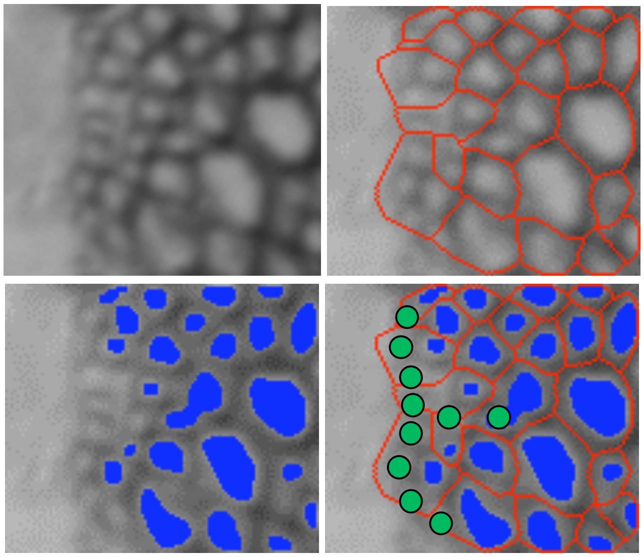

Figure 8 (a-d). Final Results – Test Image #6

These results highlight several strengths of the algorithm. Given an image with reasonable contrast and boundary resolution, the algorithm finds and identifies a watershed-like basin [5] that is an axon boundary. The algorithm also processes and displays the results in a manner that makes error-checking straightforward.

The results also demonstrate the algorithm's shortcomings. Each basin (or boundary) that is identified is assumed to contain an axon. This is not necessarily the case. The algorithm lacks a method to check the geometry of the identified feature to see if it is indeed a rounded, oblong-shaped axon. Another inadequacy is that the cookie-cutter approach to thresholding is not appropriate for the inconsistent sampling of images. When images vary greatly in contrast, perspective, quality and feature size, the fixed-size of the structural element (disk) used in the top-hat transformation leads to variable results.

An improved algorithm would build upon these results to become more robust. One such method of achieve a more versatile algorithm would be to implement region-based image processing, to identify regions such as large voids or dense bundles of axons during the pre-processing phase. Then, the algorithm could decide to vary thresholding parameters for each region in an effort to minimize the false-positives seen in Figure 7. Likewise, a region-based approach could help increase the sensitivity of the blob detection to avoid large numbers of false-negatives experienced in Figure 8. Another upside to region-based processing is that it will cut-down on computing resources for large input images.

A second area for improvement is refining the "cookie-cutter" approach to thresholding which uses a fixed-size (disk) structural element. Since it is unreasonable to plan for constant-size axons, the size of the structural element should adapt to the statistics of the input image. A crude grayscale thresholding could be conducted as a pre-processing step to identify the largest, most pronounced axons in the image. Then, by declaring a region of interest around the largest axons, the algorithm could proceed to experimentally determine the appropriate size structural element by repeated dilations and erosions.

Another key aspect of the algorithm's performance that must be mentioned is the quality of images provided as inputs. It is highly unreasonable to expect that a microscopic image of any given quality may be input to the system for analysis. An agreed-upon specification should be set between the algorithm designer and researchers regarding critical image criteria. Included would be focusing/clarity requirements, zoom requirements and lighting requirements.

SUMMARY

The design and feasibility of an automated algorithm for biological image segmentation is presented. Microscopic images of a sliced neural bundle are input and identified (counted) bundles of axons result.

The algorithm design incorporates traditional image processing building-blocks, including morphological image processing and geodesic image segmentation, to process and count axons in the images.

Experimental tests with gold-standard images show the algorithm to have issues with robustness across a range of image features and qualities. Several test images yield minimal false-positives and false-negatives, while other images suggest an over- or under-sensitive axon counting algorithm.

The limitations of the algorithm include an unintelligent, non-adaptive thresholding procedure that is the root of the robustness problem. Design and logic refinements, including a region-based approach to segmentation, could dramatically improve overall performance across the different types of test images.

The results are promising and suggest that further studies of this application would result in a system that is both robust and versatile.

ACKNOWLEDGMENTS

This work was conducted under the guidance of Dr. Frank Merat, Case Western Reserve University, as a final project for EECS 490: Digital Image Processing. This project supports the research of Professor Scott Howell, University Hospitals of Cleveland.

REFERENCES

[1] R. Gonzalez and R. Woods, *Digital Image Processing*, *3^{RD} Edition*, Prentice-Hall, 2007.

[2] D. Garcia, "Automatic Image Segmentation (OTSU Implementation), " *MATLAB Central*, Accessed December 2007, <http://mathworks.com>, Path: MATLAB Central; File Exchange; Image Processing; Morphology and Segmentation; Automatic Image Segmentation.

[3] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-9, no.1, pp. 62-66, 1979.

[4] V. Grau, A.U.J. Mewes, M. Alcaniz et al., "Improved watershed transform for medical image segmentation using prior imformation," *IEEE Transactions on Medical Imaging*, pp. 447-458, vol. 23, no. 4, Apr 2004.

[5] F. Long, H. Peng and E. Myers, "Automatic Segmentation of Nuclei in 3D Microscopy Images of C.elegans," *4^{TH} IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 536-539, 2007.

[6] R. Gonzalez, R. Woods and S. Eddins, *Digital Image Processing Using MATLAB®*, Prentice-Hall, Ch. 9-10, 2004.

[7] Reindeer Graphics, Inc., "Binary Image Processing," *Image Analysis Cookbook 6.0*, Accessed Dec 2007, <http://reindeergraphics.com>, Path: Documentation; Image Analysis Cookbook; Part 15.