

Automated Counting Axons in Optic Nerves using Hough Transformation

Van Anh Tran

Department of Electrical Engineering and Computer Science

Email: vananh.tran@case.edu

Abstract: I developed an algorithm to count the number of axons in optic nerves automatically using Hough Transformation to detect axon edges. Due to the low quality of nerve images, a pre-process is required using Gaussian filter.

Keywords: automated counting, axons, optic nerve, Hough transformation

Introduction

Knowing the number as well as size of axons in normal and diseased brain has a high potential in understanding the developing of mental diseases, aging process in the brain [1]. This work has been done by manually counting by neuroscientist. It is very time consuming and not effective. This emerges a need for an automated counting number of axons in human brains.

Current methods are based on simple automated thresholding method [2]. Another approach is using Hough Transformation to detect Ellipse boundaries and using active contours [1]. This approach is limited in only ellipse boundaries but in fact, optic nerve axons have non-analytical edges.

So I implemented a new method based on generalized Hough transformation for arbitrary shaped axons.

Pre-processing

Because optic nerve images are in low quality so to improve the accuracy, I first

preprocess the images with Gaussian filter several times to smooth the edges and get rid of noise edges. Then I use Canny edge detection algorithm to get the binary edge images to be used in Hough Transformation. The reason for choosing Canny edge detection is it provides a good and less noise edge image.

Hough Transformation and Generalized Hough Transformation

Hough transformation is a technique used to detect particular shapes in an image. Classical Hough transformation can be applied for only analytical shapes such as lines, circles, ellipses...

For example, to detect a line $y=ax+b$ within a collection of points, we can also write the line equation in the normal form:

$$r = x.\cos\theta + y.\sin\theta$$

So for a particular point (x_0, y_0) :

$$r = x_0\cos\theta + y_0\sin\theta \quad (1)$$

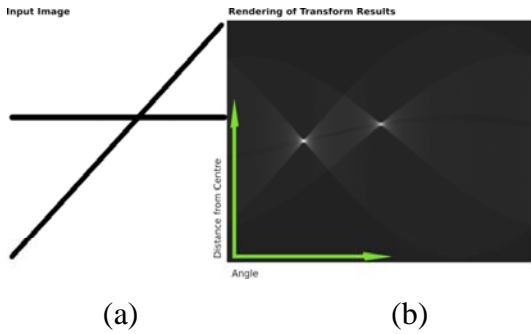


Fig 1.1. Example of Hough transformation for a line

(a) input image and (b) Hough space
 r is a function of θ . Hough space for this point is the line in plane (r, θ) that satisfies (1). After getting Hough spaces for all points in the image, Hough transformation with detect the local maxima in the Hough space and get the value of r and θ of this local maxima and considers each maxima point as parameters for a potential line.

For non-analytical shapes, generalized Hough transformation is used. In this case, instead of using parametric equations for boundaries of objects, we use a look-up table between the positions, orientations and Hough parameters. We can use gradient of the image computed in pre-processing steps by many edge detection methods as the look-up table.



Fig 1.2. Example of arbitrary shape

For a generalized shape, we need to define three parameters $\{y,r,\beta\}$ in which, $y=(x_{ref},y_{ref})$ is reference point for the shape, r is the radius, and β is orientation of the shape [3].

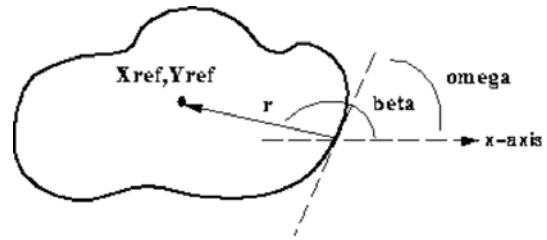


Fig 1.3. Example of Hough parameters

For any point (x_0,y_0) :

$$x_{ref} = x_0 + r \cos(\beta)$$

$$y_{ref} = y_0 + r \sin(\beta)$$

Value of r and β can be computed from the look-up table.

Experiment results

This method is implemented in Matlab using Hough transformation code from Dimitrios Ioannou [7]. I used three provided images to test the accuracy of this method.

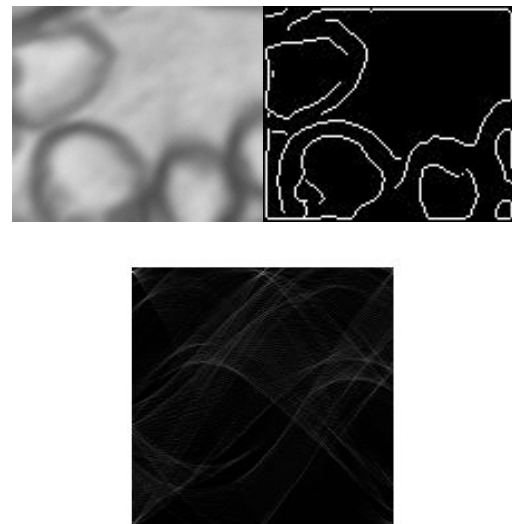


Fig 1.4: Example of results

- (a) Original image
- (b) Edge image
- (c) Hough Transformation

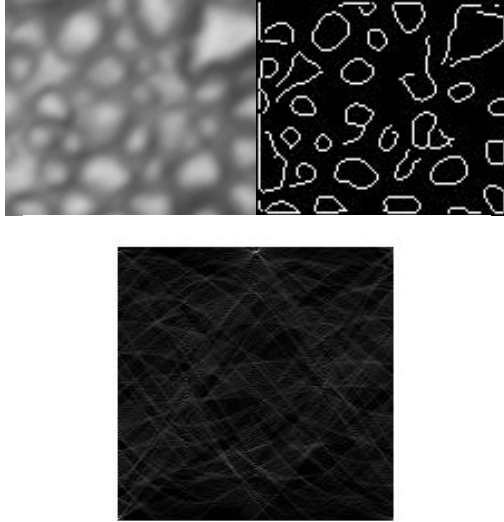


Fig 1.5. Example of results

(a) Original image

(b) Edge Image

(c) Hough transformation

Image	Standard	Results	Percentage
Sample 2.tif	4	25	625%
Sample 3.tif	25	14	56%
Sample 4.tif	35	46	130%

Table 1.1. Accuracy of this method

The reason that I got high number of axons in 'sample.tif' because this is a close-up image of the axons, the boundaries are big so for each axon, there are two edges, and Hough transformation detects them as two separate objects. The conclusion can be drawn from the result is this implementation has low accuracy.

Although in this implementation of Hough transformation, you can define a

threshold for local maxima, but it's optional.

Discussions

This algorithm still has a lot of limitations and low accuracy. The first step needed to improve it is reconstructing the objects after peaking 'high score' points in Hough transformation bins to eliminate any redundant peaks. Also this implementation detects any shapes so it's prone to noise. This is shown in large number of axons detected in 'sample2.tif' example.

Another limitation is it can be applied for only odd size square images.

References

- [1] Ying-Lun Fok, Joseph C. K. Chan, and Roland T. Chin, *Automated Analysis of Nerve-Cell Images Using Active Contour Models*, IEEE transactions on medical imaging, VOL. 15, No. 3, June 1996
- [2] Istvan Cseke, *A Fast Segmentation Scheme for White Blood Cell Images*, Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on Pattern Recognition, 1992. Vol.III.
- [3] D.H. Ballard, *Generalizing the Hough Transformation to Detect Arbitrary Shapes*, Pattern Recognition 13, 2, April 1981
- [4] Richard O. Duda, Peter E. Hard, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Technical note 36, April 1971.
- [5] R. C. Gonzalez, R. E. Woods, *Hough transformation*, Digital Image Processing, 3rd Edition
- [6] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins, *Hough*

Transformation, Digital image
processing using Matlab, Prentice-Hall
2004

[7] Hough Transformation
implementation belongs to Dimitrios
Ioannou : dimitris@cyra.com