Guided Minimum Spanning Forrest based vessel segmentation in optical image

Aashis Khanal

Georgia State University  
Atlanta, Georgia  
[akhanal1@student.gsu.edu](mailto:akhanal1@student.gsu.edu)

**ABSTRACT**

Eye imaging has been advanced so much in recent years. One of the extensively studied problems is studying the abnormality of blood vessels in the eye. A wide range of diseases like diabetes, Retinopathy of Prematurity(ROP) leads to some discrepancy in vessel structure. Also, specific defects in the vascular network can help identify possibility blindness in future. Current state-of-art techniques used by ophthalmologist are primarily based on manual inspection of vessels which can be unreliable because of human vision limitation and wide range of variation in optical images. A well-trained computer based system can robustly detect such variations and generate far more reliable finding than a general human being. The vascular system mimics the structure of tree having a root and expanding and branching away from it. I purpose a graph based segmentation of eye vessels. The proposed algorithm is a guided Minimum Spanning Tree algorithm that restarts itself from different sections of an optical image to correctly extract the vascular network.

**Keywords**

Image Processing, Vessel segmentation, Graph based image processing, Ophthalmology

# Introduction

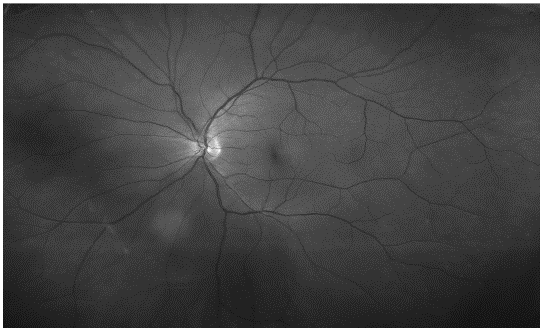
An ophthalmologist can deduce a lot of crucial information for diagnosis and prognosis of various diseases from the vascular structure of an eye. The methodology they use solely depends on the limitations of a human’s eye perception [1,2]. Some discrepancies are not perceivable to human eyes, and sometimes the speculation depends on different other external factors as well. But a computer-based prediction is always uniform. In this work, we have tried to develop a system that extracts the near-true vessel structure for a given optical image. Some notable attributes that are studied as certain diseases marker are AV ratio, branching angles, bifurcation, fractal dimension, tortuosity, vascular length-to-diameter ratio and wall-to-lumen length. For instance, AV ratio measures the diameter ratio of six largest vessels including three arteries and three veins. Likewise, branching angles are the measure of how the optical vessels are branched [1,2,3,4]. The diseases that are widely studied based on these attributes are Age Related Macular Degeneration (Wet/Dry), Optic Atrophy, Optic Neuritis, Papilledema, Ischemic Optic Neuropathy, Glaucoma, and Diabetic Retinography. The state of the art technique among ophthalmologist to calculate these metrics is manual inspection with manual tools like measuring scales [1]. Having all these, an accurate vessel segmentation and extraction method is a necessary step for automated disease diagnosis system.

The developed system here is applied for fundus image. A fundus image is captured using a fundus camera with a medical procedure. Figure 1.a is an example of fundus image. As we see it is difficult to exactly perceive the vascular structure with human eye. The methodology we purpose here goes through different filtering and enhancement steps to make sure that even the minute structural integrity of the vascular structure is preserved. We use the green channel of the image and apply bilateral filter to the image following by Gabor filter with a Gabor bank [1]. We use green channel because the filters used here respond better to the green channel. Once enhancement is done the fundus image is represented as a 4-connected lattice graph. The segmentation is the result of a customized Minimum Spanning Tree algorithm ran on the resulted lattice graph. I have grouped the paper in seven sections. First, general concept is introduced. The second section is dedicated to the procedure of image enhancement. The fourth section is dedicated to selection of seed to guide the vessel segmentation algorithm on graph. The graph representation concept and procedures are discussed in section 5. Similarly, section 6 is the actual Minimum Spanning Tree algorithm applied to the graph. However, the algorithm is modified to behave in specific desirable way. Section 7 is the result section where I discuss the results obtained and finally the last section is the conclusion.

A picture containing indoor

Description generated with high confidence

*Figure 1.a. A fundus optical image [5].*



*Figure 1.b. Green channel of fundus image from figure 1.a*

A close up of a map

Description generated with very high confidence

*Figure 1.c. Result of proposed vessel segmentation method*

As we can see from the images above, it is obvious that vascular network mimic tree structure. And tree structure can be represented as a graph [1].

A screenshot of a cell phone

Description generated with high confidence

*Figure 1.d. Flowchart of segmentation process*

Once the graph is constructed, we penalize a move from one node to another node (In lattice graph, each node is a pixel). The penalty is low if we move from a pixel to another with same pixel value and high while we go traverse from pixel with low intensity value to high intensity value or vice versa.

# Related work

One of the major challenge in vessel segmentation is preserving the width of the vessels. Although present methods for vessel segmentation tried to address this problem they achieved less than expected accuracy on that aspect. A significant of the existing methods used Gabor based filtering [6], thresholding, histogram based segmentation and clustering [7]. In this paper, I present a novel approach that combines thresholding [8], histogram, bilateral filtering, Gabor filtering and graph theory in a beautiful way. This combination therefore leads to self-adjusting Minimum Spanning Tree based segmentation. The problem with thresholding and clustering methods is unpredicted capture of noise. The problem with other method is not able to accurately maintain vessel’s width. In this paper I have tried to address these concerns.

# Image Enhancement

The original image as shown in Figure 1.a consist of a lot of variation in contrast, brightness among vessels and background. Our motive is to identify vessels and extract it regardless of how bright or dim it is compared to background. Well, this is one of the major challenging task because as stated earlier image comes in different variations and the motive is always to develop systems that is the most generalized version, i.e. works in all the cases with minimal or no difference. The image enhancement process consists of Bilateral filtering followed by a Gabor kernel bank convolution with elementwise maximum taken. The next subsection discusses each of these filtering and how these are applied to get the desired enhancement.

## Image Filtering

Image convolution is the basic operation done in image processing. Almost all filtering is some sort of image convolution. So, convolution and filtering are used interchangeably. Convolution means sliding a small matrix or kernel over the image so that the value of each pixel changes based on that kernel.

The kernel matrix is normally an odd dimension matrix like 9 \* 9, 11 \* 11 and so on. For example, a simple box blur kernel is as following table:

A picture containing shoji

Description generated with high confidence

*Figure 3.a. A 3 \* 3 kernel.*

The filtered value of pixel is the summation of each element of a kernel multiplied with the corresponding element from the image. In the box blur, the pixel at center position is the average of nine neighboring pixels. And the effect would look something like following.

A large white house

Description generated with very high confidence

*Figure 3.b. Image convolution example with a 3\*3 box blur [16].*

Similarly, different filters with different values can generate different effects. One of the widely used filter is Gaussian kernel filter. It is often used along with other filters to get advanced filtering results. One of them is Bilateral filtering which will be discussed in next section.

A close up of a door

Description generated with high confidenceA picture containing object

Description generated with very high confidence

*Figure 3.c. A 8 \* 8 gaussian kernel. 3.d. A visualization of gaussian kernel [16].*

*A house with trees in the background

Description generated with very high confidence*

*Figure 3.e. Gaussian blur example [16].*

Gaussian kernel for two-dimension space is given by following equation [17, 18].

A close up of a clock

Description generated with high confidence

Where, *x* is the distance from origin in the vertical axis, *y* is the distance from origin to horizontal axis, and *σ* is the standard deviation of the Gaussian distribution.

## Bilateral Filtering

Most of the blurring filters like Median Filter [10] and Gaussian Blur [9] generally average the pixel value based on the filter window size. In the process, the edge is lost, since pixels on either side of the edge are near averaged. But a Bilateral Filter uses non-linear transformation on the neighboring pixels values. Specifically, it computes a pixel output as a weighted average of neighboring pixels [11]. The popularity of Bilateral Filter is due to its edge preserving capability. The bilateral filter can be defined as [9,10]

A close up of a logo

Description generated with very high confidence

Where the normalization term ***Wp***preserves the image energy. ***Wp***is given by

A close up of a logo

Description generated with very high confidence

Where,

***I filtered*** is the filtered image.

***I*** is the original image.

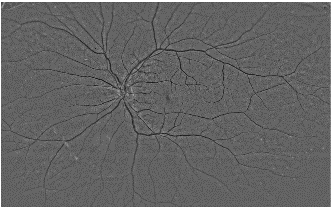
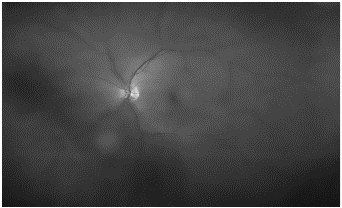
***x***are the coordinates of the current pixel to be filtered.

Ω is the window centered in *x.*

***Fr***is the range kernel for smoothing differences in intensities

**Gs**is the spatial kernel for smoothing differences in coordinates.

In this system the default value used for Ω is 41 and, for ***Fr*** and **Gs** it is 20. These values are best fit for this purpose and are carefully chosen from series of experiments on different values.

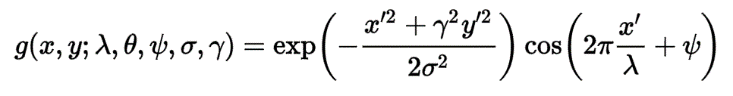


*Figure 3.f. The cartoonish looking image because of Bilateral filtering. We see the cartoonish change because the bilateral filter smooths the non-edge region but not the edge region. The parameters should be adjusted to specify how much difference in intensity value can make an influence and how much far pixels will do so. Figure 3.g. The difference between the original image 1.b. and 3.a. There were some negative values in the process which were rescaled to 0 to 255.*

The idea for the taking difference between image 1.b. and 3.a. is that the Bilateral filter was active only on non-edges region, so the difference gives the best distinction between vessels and background [1]. We used this differenced image as the input in Gabor filter [1] which we discuss in the next section in detail.

## Gabor Filter

Gabor filtering is a convolution technique in which the response is maximum to predefined orientation and texture structure. This property makes it very effective in vessel like texture separation and is being widely used for similar purpose. A 2d Gabor filter is used in this method which is a Gaussian kernel function in modulated with a sinusoidal plane wave [12, 13]. The Gabor filter can be generated based on following equation.



Where,

A close up of a clock

Description generated with high confidence

and

A picture containing object

Description generated with very high confidence

In the above equation, λ represents the wavelength of the sinusoidal factor. The wavelength factor responds well to different size vessels. θ specify the orientation of the normal to the parallel stripes of Gabor function. Different orientation of the kernel responds to differently oriented vessels. φ is the phase offset, σ(sigma) is the standard deviation of the Gaussian envelope and ϓ is the spatial aspect ratio, which specifies the ellipticity of the support of the Gabor function.

Blood vessels have different width and orientations. To get the maximum effect of convolution, I used different variations of Gabor filters called as Kernel Bank. The result of the convolution is the elementwise maximum across each of the filtered result for each kernel [1].

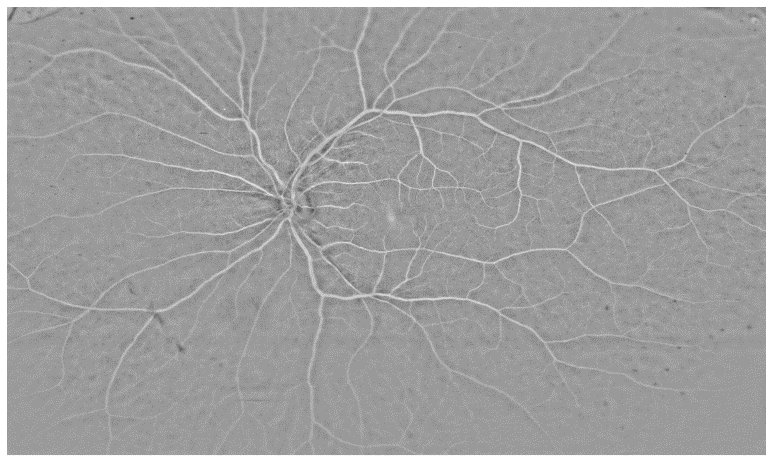
Where, is the convolution output of pixel (x, y) using pth kernel.

A close up of a white background

Description generated with high confidence

*Figure 3.h. Visualization of the Gabor kernel bank. The kernel size used is 31 and gamma is 0.7 and both were fixed. However, other parameters were adjusted to best respond to variations of vessels.*

I have used three kernel banks to create a bigger one. Each one of those three has 64 different values for theta. So that all the three kernels respond strongly to 64 orientations of vessels. The first one has parameters kernel size=31, gamma=0.7, lambda=5, sigma=2. This is supposed to respond strongly to small vessels. Second one has parameters kernel size=31, gamma=0.7, lambda=8, sigma=3, and the third one has kernel size=31, gamma=0.7, lambda=11, sigma=4. The last one responds to larger vessels. So, in this way the kernel bank in figure 3.c. responds strongly to small, medium and large vessels along with 64 different orientations. From the figure 3.h. we can observe that the kernel is supposed to respond to white vessels but the resulted difference in figure 3.g.­ has vessels portion dark. Therefore, the inverted response of image in figure 3.g. is given as input to the Gabor filter process [1].



*Figure 3.i. Inverted response of difference between original and Bilateral response (Figure 1.b. and Figure 3.f.)*

A close up of a coral

Description generated with high confidence

*Figure 3.j. The result of Gabor filter applied to inverted response of the difference between original (Figure 1.b.) and Bilateral response (Figure 3.f.). We can see that the white vessels responded very well to the designed kernel bank (Figure 3.h.)*

# Seed Selection

One of the major challenging task in segmentation process using graph algorithms is to find specific start points. The start point is called seed. We have used a histogram based approach for seed selection. The idea is to select all the pixels that have the highest pixel value from the result of Gabor filter (Figure 3.j). This sounds easy, but a major challenge remains. It’s not necessarily always the case that the pixel with highest intensity value fall within the vessel region. Having that, suppose a seed outside of pixel value is selected, the result is most likely to be unusual. To overcome this problem, and given that we cannot guarantee that pixel with highest intensity value falls within vessel region, we have applied a little bit different approach. The basic idea of that approach is to only select the pixels which are not isolated from vessel region as seed. On doing experiment on different test images the histogram approach always separates out the pixel with high intensity values. So, we are confident that this approach works well on wide range of images. However, it must be taken into consideration that selecting the right seeds only leads to correct vessel segmentation. This concept will be discussed in more detail in Minimum Spanning Tree algorithm section.

A screenshot of a cell phone

Description generated with high confidence

*Figure 4.a. is the histogram of the original image (Figure 1.a.).* A screenshot of a cell phone

Description generated with high confidence

*Figure 4.b. is the histogram of the response of Gabor filter. In Figure 4.b, we can clearly observe that a group of pixels stand out from other pixels.*

A picture containing outdoor object

Description generated with very high confidenceA picture containing outdoor object

Description generated with very high confidence

#### Figure 4.c. When we threshold the highest intensity value pixel from the histogram in Figure 4.b., we get this result. All these pixels are the candidate seeds but how many are used will be discussed later in more detail in algorithm section. Similarly, Figure 4.d. is the seed of another image obtained by similar approach.

Although we were able to extract a valid seed structure, there is a flaw with this approach. In the top-left of figure 4.c. there are isolated pixels. Likewise, in other places of both the images, there are isolated pixels. Those isolated pixels are a very bad choice to start the segmentation process. So, to overcome this problem, I convolve the skeleton image in Figure 4.c with the following kernel:

A close up of a keyboard

Description generated with high confidence

*Figure 4.d. Kernel used to remove isolated pixels from the result obtained by thresholding. The idea here is that a pixel will be considered as a final seed only if it has similar neighbors [1,2,3].*

*A picture containing outdoor object, sitting

Description generated with high confidenceA picture containing sitting, outdoor object

Description generated with very high confidence*

*Figure 4.e. The final seed of the image 4.d. Figure 4.f. The final seed of the image 4.c. We can see that in both images the structure of the vessel is preserved, and the isolated pixels are removed. Thus, it is very likely that our algorithm picks the pixels that lie within vessel region.*

# Graph Representation of Image

A graph is a most natural way of representing any things connectivity. A graph consists of nodes and edges. Formally, a graph G can be represented as G = {V, E}. Where V are the nodes and E are the edges among those nodes. An edge can be any sort or relationship. For example, if we represent cities as nodes, edge can represent the roads between those cities. Similarly, edge can have weights and in this example edge weights can be the distance between the cities.

## Image Lattice Generation

A similar concept has been used to represent the image as a graph upon which we use the Minimum Spanning Tree algorithm. Before jumping to the representation, I would like to introduce basics of image representations. Each image is composed of pixels. Each pixel’s abstraction is represented by x, y in cartesian plane. And in case of color image, we have three such abstraction dimensions. Each of those represent red, green and blue channel in RGB representation. Each value of pixel ranges from 0 (black) t0 255 (white).

Let us represent the image as following:

***Io*** = Original color image (Figure 1.a.)

***Ir*** = The red channel among three dimensions (R, G, and B)

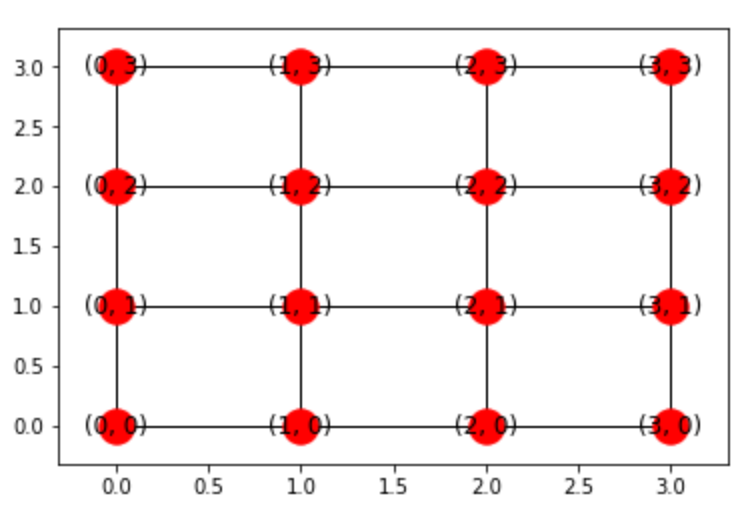
***Ig*** = The green channel among three dimensions (R, G, and B)

***Ib*** = The blue channel among three dimensions (R, G, and B)

***(M, N)*** = resolution of image.

An arbitrary pixel can be represented as ***(Xi, Yi).*** Given that, we have M Times N pixels. Thus, our graph representation of ***Ig*** is given as:

|V| = M \* N (Number of Nodes)



*Figure 5.a. An example 4 by 4 lattice graph. It has 4 \* 4 = 16 nodes and approximately 16\*4 edges. There are not exactly 16 \* 4 edges because boundary nodes have either 3 or 2 edges. As we can see in the figure, each of the node are labeled exactly as the pixel position. This significantly helps us to recollect all the calculations to an image with segmented vessels.*

Each pixel ***(Xi, Yi)*** is connected to four of its neighbor pixels (Nodes). So approximately we have M \* N \* 4 edges. A thing to note here each node that represents ***(Xi, Yi)*** pixel has node label ***(Xi, Yi)*** as well. This makes tracking and final image reconstruction easier which we will discuss on algorithm section in more depth.

## Path cost assignment

The advantage of using a graph is to use some sort of cost associated with traveling from one node to another (always neighbors) [1]. The output from Gabor filter (Section 3.2/Figure 3.e.) is used as input to the cost assignment method. Suppose ***I0*** is the original image (Figure 1.b.) and ***Ig*** is the result of Gabor filter (Figure 3.e.) from section 3.2. If the cost of traversal from pixel ***P = (x, y)*** to one of its four neighbor ***P` = (x`, y`)*** is represented by ***C (P, P`)***, and the final cost is calculated as:

+

The values of and are set based on experiments. One thing to note here is, we use inverted response of ***Ig*** instead of ***Ig*** to make ***I0*** and ***Ig*** uniform. This is also the one of the building block for cost assigning strategy. The cost for given original image ***Io*** (Original image means green channel) is given by the following equation.

Similarly, the cost for Gabor response (255 - ***Ig***)is given as:

I have used and for this experiment. The value of alpha used was 3. The term gives the maximum among . The original green channel ***Ig*** and inverted Gabor response (255 - ***Ig***) has the vessels region darker than the background. Which means the vessel regions will have intensity value that tend to go towards 0 as they become darker. Likewise, the intensity of background tends to go towards 255 as they become whiter. If we go away from darker region to whiter region the cost will be significantly increased.

# Algorithm

I have used Minimum Spanning Tree algorithm for the final segmentation process. Specifically, a customized variant of Prim’s algorithm is used [13]. A Minimum Spanning Tree is a tree that spans to the whole graph and has a minimum weight in weighted graph and minimum number of edges in unweighted graph. Suppose a courier company wants to connect to hundreds of cities. A MST can be the least costly path among those cities. Similarly, we have a lattice graph from section 5. And traversal from a pixel to one of its neighbor incur a cost. The cost will be higher if the neighbor pixel is white that this pixel. For example, A dark pixel at vessel region might have intensity value 50 and a neighbor, that falls in non-vessel region have intensity value 200. The cost incurred here will be huge. So, the algorithm prefers to traverse along the vessel to keep the cost minimum [1]. A min heap is maintained for computational efficiency. With min heap the next low-cost pixel (node) can be found with few heap operations ο (***log n***) rather that ο (***n log n)*** for sorting and finding the minimum. The Prim’s algorithm is follows:

*Algorithm 6.1: Prim’s algorithm for MST [15]*

***Input:*** *- Weighted graph: G = {V, E}*

***u****: start node.*

***Visited = [u]****: a list to hold visited nodes.*

***i****: incoming priority (i = 0)*

***Q****: a min heap based priority queue*

***push****: an operation that pushes to Q and maintain heap property as well internally.*

***Pop****: an operation that pops minimum value from Q and readjust to maintain min-heap property.*

***Output:*** *A Minimum Spanning Tree of G*

***MST\_LIST = []:*** *A list to hold minimum spanning tree*

1. *For node v in neighbors (u):*
2. ***push*** *(****Q****,* ***cost*** *(u, v), i++, u, v)*
3. *While Q is not empty:*

*w, p, u, v =* ***pop****(****Q****)*

*if v in Visited:*

*continue*

*Add v to visited.*

1. *For node w in neighbors(v):*

*If w not in visited:*

1. ***push*** *(Q, cost (v, w), i++, v, w)*
2. *Add edge (u, v) to MST\_LIST*

In the above algorithm, Q is a min heap implementation of priority queue.

A close up of a clock

Description generated with very high confidence

*Figure 6.a. An example of min heap*

Figure 6.a. is an example of binary heap. In our lattice based traversal, we maintain a heap with four children since each pixel has four possible pixels where algorithm can traverse. A simple way to implement binary heap is to use array as underlying data structure. The array implementation details are as follows:

*Array [0] will be the parent node*

*Given ith node,*

*Array [i/2] returns parent node*

*Array [2\*i+1] returns left child*

*Array [2\*i + 2] returns right child*

A priority queue has assigned a priority to each task. The task with highest priority will be the next to execute. In case two task has same priority, the one that arrived earlier will be used. In our case, the traversal cost from a node to another is the path priority. In case of same cost for two nodes, the tie breaker is the ***i (incoming priority).*** A min heap is an abstract data structure whose root element is the one with lowest cost in our implementation. Each child has a higher cost of traversal than that of parent. Adding and deleting a new element takes up to ο (***log n***) and a pop operation always yields a node with minimum cost.

Algorithm 6.1 is the simple Prim’s algorithm. The concept of Minimum Spanning Tree is that it traverses through all nodes following the minimum path cost greedily. In case of vessel segmentation, we want to run the algorithm to nodes that spans the vessels-Not the whole nodes (pixels). To achieve this goal, a little modification has been done in the Prim’s algorithm. Also, being greedy algorithm, there is one another drawback of Prim’s algorithm. Suppose, you started from a pixel outside of vessel region, the algorithm still runs following the greedy approach. Which means it will pick the surrounding least costly pixel to travel next. This problem is solved effectively by the histogram based seed selection approach as discussed in section 4.

A picture containing sky, tree, outdoor, text

Description generated with very high confidence

*Figure 6.a. Segmentation result when seed is selected in non-vessel region (circled with blue). As we can see, before finding the right vessel region and moving along it, the algorithm captured some noise.*

We call the modified MST algorithm self-guided because it restarts from new seed that hasn’t been visited before. This ensures that we do not miss isolated vessel regions. If we let the algorithm to run for considerable amount of time from one seed, then it’s sure that it captures some noise. This is because, once all the vessel region is visited, the algorithm will consider non-vessel region with least cost to travel. Starting from seed, the algorithm considers the next least cost vessel as a vessel only if the cost is less expensive then the threshold.

*Algorithm 6.2: Prim’s algorithm for vessel segmentation [15]*

***Input:*** *- Weighted graph: G = {V, E}*

***seeds****: List of seed nodes.*

***Visited = [u]****: a list to hold visited nodes.*

***i****: incoming priority (i = 0)*

***Q****: a min heap based priority queue*

***push****: an operation that pushes to Q and maintain heap property as well internally.*

***Pop****: an operation that pops minimum value from Q and readjust to maintain min-heap property.*

***Output:*** *A Minimum Spanning Tree of G*

***MST\_LIST = []:*** *A list to hold minimum spanning tree*

***Threshold****: A constant*

1. *Pop u from seed:*
   1. ***For*** *node v* ***in*** *neighbors (u):*
2. ***push*** *(****Q****,* ***cost*** *(u, v), i++, u, v)*
   1. ***While*** *Q* ***is******not*** *empty:*

*w, p, u, v =* ***pop****(****Q****)*

***if*** *v* ***in*** *Visited:*

***continue***

***if*** *weight (u, v)* ***<*** *Threshold:*

*Add v as vessel pixel.*

***else*** *break*

*Add v to visited.*

***If*** *v in seeds:*

*Remove v from seeds.*

* 1. ***For*** *node w* ***in*** *neighbors(v):*

*If w not in visited:*

1. ***push*** *(Q, cost (v, w), i++, v, w)*
   1. *Add edge (u, v) to MST\_LIST*

Algorithm 6.2 is the modified algorithm that distinguishes vessel and non-vessel pixels using Minimum Spanning Tree approach. The algorithm the Prim’s algorithm with two things added. In Prims, we traverse through the pixel with lowest cost to travel. In this, we only consider that lowest cost pixel if the cost is less than a threshold.

# Running Time

The running time of Prim’s algorithm depends on what data structure we use to maintain the immediate nodes and cost incurred. How fast we retrieve the node with minimum cost depends on that data structure. For example, if we use adjacency matrix, and search in that the time complexity would be ο (|V|2). If we use binary heap and adjacency list, the complexity would be ο (|E| log |V|). Where |E| is the number of edges and |V| is the number of nodes.

# Experiments

The experiment was done on 29 images. We choose 15 images as training set and 9 images as test set. To have a metrics for test score, we use F1 score test. F1 score formula is given as following:

Precision is defined as how many selected items are relevant. Whereas recall is defined as how many relevant items are selected. The ground truth values are to be

# REFERENCES

1. Rolando Estrada, Carlos Tomasi, Michelle T. Cabrera, David K. Wallace, Sharon F. Freedman, and Sina Farsiu, “Explanatory Dijkstra forest based automatic vessel segmentation: application in video indirect ophthalmoscopy(VIO)”, Biomedical Optics Express, OSA 2012
2. W. Tasman, A. Patz, J. A. McNamara, R. S. Kaiser, M. T. Trese, and B. T. Smith, “Retinopathy of prematurity: The life of a lifetime disease,” Am. J. Ophthalmol. 141, 167 – 174 (2006).
3. G. A. Gole, A. L. Ells, X. Katz, G. Holmstrom, A. R. Fielder, A. Capone, Jr, J. T. Flynn, W. G. Good, J. M. Holmes, J. A. McNamara, E. A. Palmer, G. Quinn, E., M. J. Shapiro, M. G. J. Trese, and D. K. Wallace, “The
4. Mayuri Bhargava, MD, “Current Concepts In Hypertensive Retinography”, https://www.retinalphysician.com/issues/2013/nov-dec/current-concepts-in-hypertensive-retinopathy
5. WIDE Dataset from Duke University, <http://www.duke.edu/~sf59/Datasets/Tree_topology_PAMI_2015.zip>, http://www.duke.edu/~sf59/Datasets/AV\_Paper\_TMI\_2015.zip
6. Qin Li, Jane You, Lei Zhang, and Prabir Bhattacharya, “A Multiscale Approach to Retinal Vessel Segmentation Using Gabor Filters and Scale Multiplication”, IEEE, October 2006
7. Sookpotharom Supot, Chaichana Thanapong, Pintavirooj Chuchart, and Sangworasil Manas, “Automatic Segmentation of Blood Vessels in Retinal Image Based on Fuzzy K-Median Clustering”, IEEE, 2007
8. A. Hoover, V. Kouznetsova, and M. Goldbaum, “Locating blood vessels in retinal images by piecewise threshold  
   probing of a matched filter response,” IEEE Trans. Med. Imag. 19, 203–210 (2002)
9. Mark S. Nixon and Alberto S. Aguado. Feature Extraction and Image Processing. Academic Press, 2008
10. E. Arias-Castro and D. L. Donoho, "Does median filtering truly preserve edges better than linear filtering?", Annals of Statistics, vol. 37
11. Kaiming He, Jian Sun, and Xiaoou Tang, “Guided Image Filtering”, Department of Information Engineering, The Chinese University of Hongkong Microsoft Research Asia
12. Daugman, J.G. (1980), "Two-dimensional spectral analysis of cortical receptive field profiles", Vision Res., 20 (10): 847–56, doi:10.1016/0042-6989(80)90065-6, PMID 7467139
13. Hans G. Feichtinger, Thomas Strohmer: "Gabor Analysis and Algorithms", Birkhäuser, 1998; ISBN 0-8176-3959-4
14. Prim, R. C. (November 1957), "Shortest connection networks And some generalizations", Bell System Technical Journal, 36 (6)
15. Aric Hagberg, Dan Schult, Pieter Swart, Loïc Séguin-C, 2015, <https://networkx.github.io/documentation/networkx-1.9/_modules/networkx/algorithms/mst.html>
16. <http://aishack.in/tutorials/image-convolution-examples/>, Image convolution examples “
17. R.A. Haddad and A.N. Akansu, "A Class of Fast Gaussian Binomial Filters for Speech and Image Processing," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 39, pp 723-727, March 1991.
18. Shapiro, L. G. & Stockman, G. C: "Computer Vision", page 137, 150. Prentence Hall, 2001