Background

Automated vessel segmentation is an important image processing problem. Much of the demand is from analysis of medical images taken via different modalities such as MRI and CT scans. In our lab (Steven George) there arises a need to quantify vessel growth formation from fibrin based tissue construct prevascularized with human umbilical vein endothelial cells (HUVEC).[1]

These constructs are used to accelerate anastomosis upon implantation into the hosts. In order to optimized the seeding parameters and validate the method, GFP was transduced into the HUVEC cells in order to visualize the vessels. Confocal microscopy was used to take stacks of fluorescent images at various timepoints (2-14 days).

In the study, the seeding density and confirmation of vascularization was carried out via human quantification with ImageJ. The metrics were total vessel length/mm^2, average length of vessel network and average number of branches per vessel network.

Clearly in order to provide scalability (i.e. to grow prevascularized tissues in huge quantities) the quantification has to be automated. This project proposes to use a vessel enhancement filter that searches for geometrical structures that are 'tube-like'.

Method

The image shown below is vessel network taken at day 7 (Figure 1). The raw image is a 640x640 8bit tif image converted from the original LSM format file. I did not do any processing of the image to illustrate the performance of the vessel enhancement filter. The actual measurement of the image area is 1.3x1.3mm. The goal was to segment the image into a vessel region and background and then to skeletonize the vessel region so that geometrical lengths and branchpoints can be calculated. Clearly the crucial step is segmentation of the vessel itself; the better the segmentation the better the skeletonization that follows.

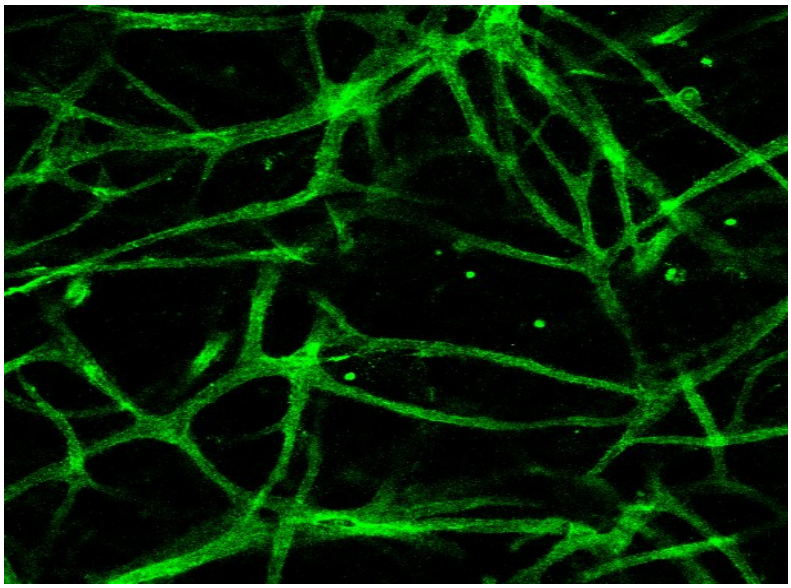

Figure 1: GFP image of vessels derived from HUVECs in fibrin gels

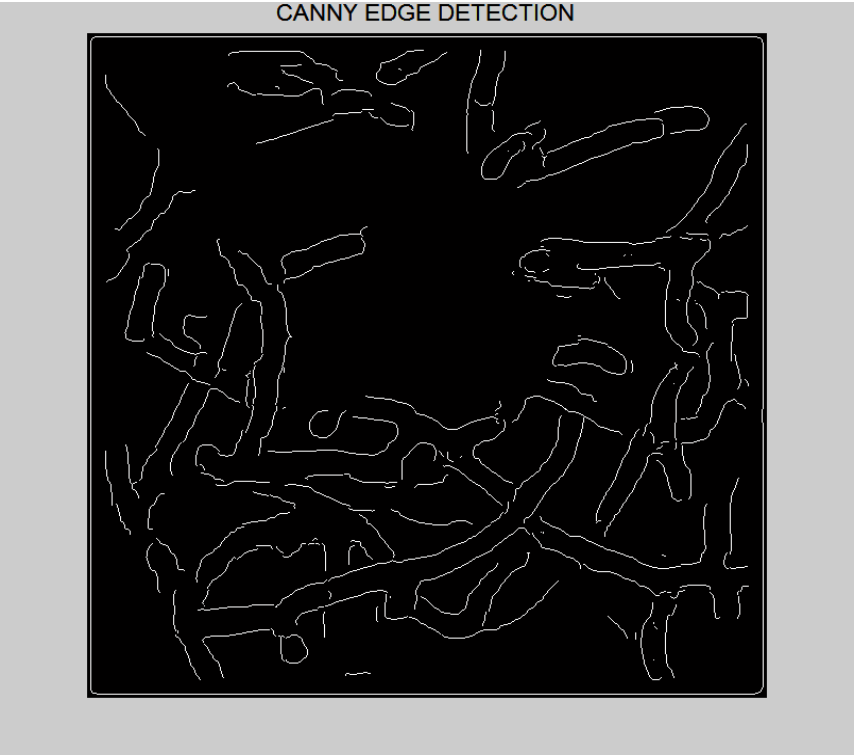

Figure 2: Canny edge detection of vessel image

Gradient Edge Detection

One of the most common method for image segmentation is using a gradient edge detection method (i.e. the first derivative). This is not a good method for tubes because we get two edges for each tube (Figure 2). The gap between the two edges has to be closed, such as using morphological closing or convolving with a gaussian to blur the edges again. The use of morphological closing makes it difficult to preserve geometrical features, is hard to control scale and is dependent on the structuring element used.

Principal Curvature based Vessel Detection

A better filter would search for geometrical features of vessels which are essentially tubes. This can be done using principal curvatures which can be found using second derivative methods[2].

The second derivative of a one dimensional Gaussian defined as :

$$G(x,\sigma)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{(x^2/2\sigma^2)} \quad (1)$$

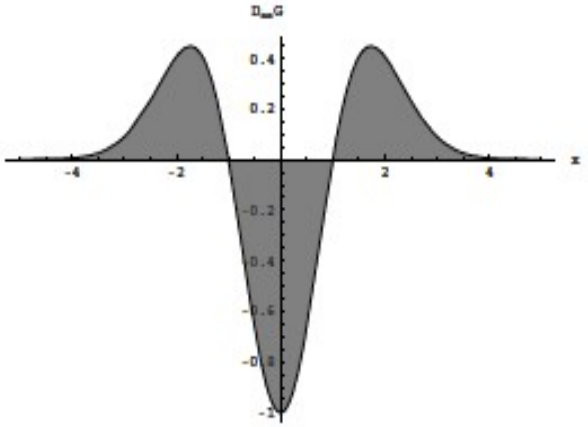

Figure 3: Second order derivative of one dimensional Gaussian kernel probe

is a kernel that has the ability to probe regions 'inside' and 'outside' the range of $(-\sigma, \sigma)$ (Figure 3), taken from [3].
Therefore a Gaussian convolution of the original image combined with the second derivative can be used to tune the response of a filter to specific line widths and additionally suppress noise.

In 2D or more dimensions the second derivatives of local intensity variations around each pixel of an image forms a Hessian matrix of partial second derivatives, which for 2D is:

$$\nabla^2 I(x, y) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \quad (2)$$

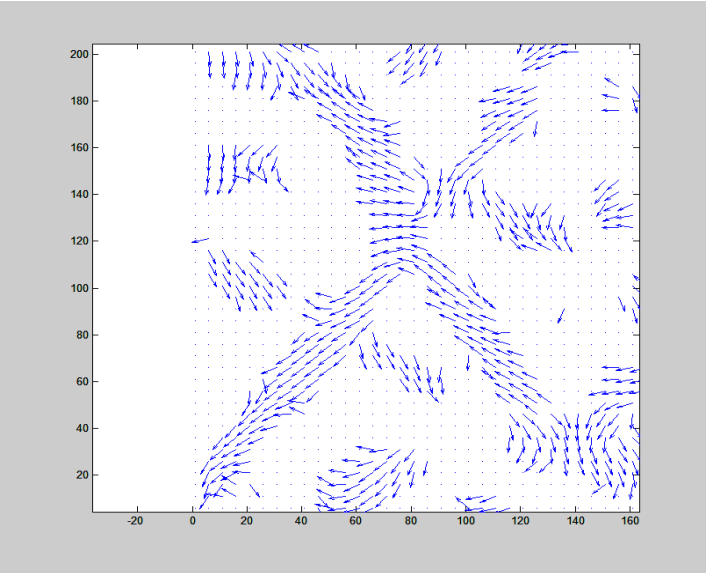

Figure 4: Vector field of first eigenvector of a cropped portion of the image, showing the direction along the medial axis line of the tubes

The principal curvatures of the image are the eigenvalues of the Hessian matrix above and their eigenvectors describe the orthonormal directions where the second derivative magnitude is given by the corresponding eigenvalues. Figure 4 shows a plot of the eigenvectors pointing along the axial line of the vessels in the image using MATLAB *quiver ( )* function. (the code is in the *principaldirection.m* file. For a 2D tube, there will be two eigenvalues, $|\lambda_1|<|\lambda_2|$.

The key assumption we make in determining that a pixel is in a tube region is that the first eigenvalue is small (or zero) and the second eigenvalue is large and negative (see Figure 3 , gaussian probe). This is the simplest possible measure, but other measures exist (such as λ_1/λ_2 to measure 'blobness' of the vessel). I found that the simple measure works best for my image.

MATLAB CODING EXPLANATION

Below is a brief walk through of the main script ('*proj.m*'). After loading the image (in this case 'pic3', the following is performed:

1) Convolution of image with a gaussian with sigma set at 11 (user set parameter) using imfilter( ) and fspecial('gaussian')
2) For each pixel ,construct a Hessian matrix of partial second derivatives (being careful with MATLAB row indexing when specifying the Laplacian operator)
3) For each Hessian matrix, calculate the eigenvalue/eigenvectors using eig( )
4) For each pixel,sort the eigenvalues, check the sign of the larger eigenvalue, if its <0, return that eigenvalue to the pixel position, otherwise set to zero (i.e pixel not in tube).

   The image below is after the above steps.(Figure 5)

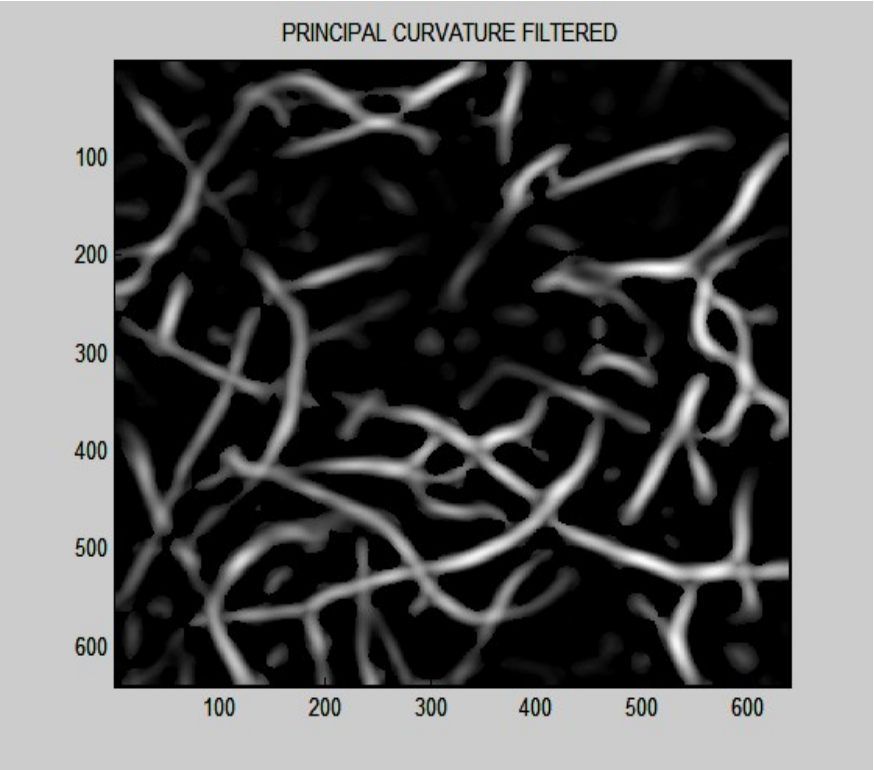

Figure 5: Resulting image after running vessel enhancement filter

Once the image is filtered for tubes , the data is thresholded using kmean( ), with 2 classes (vessel and background). The tresholded image is shown in Figure 6. Note that since the kmeans algorithm does not always converge on first iteration, I've rewrote the thresholding step as a separate function *kmean.m*.

Lastly skeletonization of the thresholded vessel network is performed, using bwmorph('skel'), which uses eight connectivity. The result is shown in Figure 7.

To quantify the vessel networks, I used bwconncomp( ) to automatically label each vessel network. Filtering was done to eliminate vessel networks that were too small (in my case <15 pixels). Then the previously mentioned metrics (# of branches in vessel network, average vessel length per vessel network and total vessel length) was calculated. The code for quantification is in *skelanal.m* The results are discussed in the next section

Figure 6: Kmeans thresholded binary image of vessel enhanced image

Figure 7: Skeletonized binary image

Results and Discussion

The results of automated quantification is shown in the table below and compared to actual human quantification[1].

7 days	Human	Filter
Average Vessel length	0.6 mm	0.09 mm
Mean Number of branches	1.5	14
Total length	12mm	10.8mm

We immediately notice that although the total length measurements are quite close,the average vessel length and number of branches measurements are vastly different. An analysis of the skeletonization reveals why: because the image was in 2D, overlaying vessel networks in the z-axis could not be differentiated (ie they lack depth perception). Hence we get huge vessel networks with lots of branches such as the one shown below (Figure 8). This overestimation of branchpoints in turns leads to an underestimation of the average vessel length / network.

Figure 8: Overlaying vessel networks obscure depth perception of the image

This means that we need three dimensional images with 3D vessel enhancement filtering. Fortunately, the principal curvature vessel enhancement filter is easily extended to 3D, just by increasing the hessian matrix terms to :

$$\nabla^2 I(x,y,z) = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (3)$$

Additionally, a simple measure[2] for eigenvalue evaluation would be :

$$\lambda_2 \approx \lambda_3 \ll 0$$
$$\lambda_{mean} = \sqrt{\lambda_2 \lambda_3} \quad (4)$$

This extension to 3D was not performed due to difficulty in converting the stack images into voxels. However it should not be difficult given more time to extend my script to a 3D image.

Finally, the program could be further optimized to improve the speed. On a AMD quadcore 2.6GHz, the program took ~15s to complete the vessel segmentation. Surprisingly eventhough a large filter window was used for the gaussian convolution (15*sigma, ~150 pixels), this was not the bottlenect of the program, showing the speed of IMFILTER.

The slowest part of the program was the necessity to evaluate 640x640 eigenvalue/eigenvectors and also to calculate the second derivatives of the image using two for-loops. Modifying the code to utilized MATLAB's parallel programming (such as using the gradient( ) function to calculate the hessian) should be doable, although a parallel implementation of eigs( ) is beyond the scope of this project.

CONCLUSION

A vessel enhancement filter using principal curvatures has been implemented on MATLAB to analyze vessel formation in fibrin based tissue constructs seeded with HUVEC s. It was found that this method could quantify total length of vessel formations but not finer morphological features due the lack of depth perception information from the 2D slices. However this method shows promise and a method has been proposed to extend it to 3D.

REFERENCES

1. Chen, X. et al. Prevascularization of a fibrin-based tissue construct accelerates the formation of functional anastomosis with host vasculature. *Tissue Engineering Part A* **15**, 1363 (2009).
2. Sato, Y. et al. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis* **2**, 143-168 (1998).
3. Frangi, A.F., Frangi, R.F., Niessen, W.J., Vincken, K.L. & Viergever, M.A. Multiscale Vessel Enhancement Filtering. *Medical Image Computing and Computer Assisted Intervention* **1496**, 130--137 (1998).