3D Reconstruction of Collagen Fibers in Cornea

Zachary DeStefano

Due Date: June 11, 2014

Introduction

I work in the graphics group with Gopi and Aditi. The group was approached by the Department of Ophthalmology to see if we can reconstruct various parts of the eye using the images provided. I decided to work on the part that involved reconstructing collagen fibers in the cornea. I attempted to take the cross-sectional images of them and use that data to display their shape in a 3D environment. The images are somewhat noisy and grainy and the fibers vary in shape and thickness so I had to use a variation of clustering and filtering in order to accurately locate the fibers in each image. I then took the segmented image and made it a slice in a volumetric data set. At the end, I took the data set and used a volume renderer to show it. Here are the steps at a glance that I will go into detail on:

1. Do Initial Smoothing of the Image
2. Do Clustering to get the fibers
3. Smooth the segmented image
4. Compare image to previous one and put results into 3D data set
5. Add image from step 3 into data set
6. Repeat Steps 1-5 for all images in data set
7. Use Volume Renderer to show the images in a 3D environment

There was an attempted step here of trying to align the images together. I tried various methods to see what the best overlap would be between segmented images but in the end, it made sense to just overlay the images on top of each other without doing any transformations of them.

Segmentation Steps Performed

This section describes in detail the steps performed in order to achieve a nicely segmented image showing the fibers.

Initial Smoothing

For the initial smoothing, I tested out filters that would make the image look less grainy than the original image. That way whatever clustering algorithm I was using would have an easier time locating the fibers. I tried a Gaussian filter and a median filter but those did not work as well as the simple averaging filter.

*INSERT IMAGE OF GAUSSIAN FILTER ON ORIGINAL IMAGE**
*INSERT IMAGE OF MEDIAN FILTER ON ORIGINAL IMAGE**
*INSERT IMAGE OF AVERAGING FILTER ON ORIGINAL IMAGE**

Clustering

This was the most difficult and most important part of the project. Locating the fibers is all about clustering together the different regions of the image where the brightness is high. With this in mind, I tried the EM algorithm, the min-cut algorithm for MRF, k-Means, simple thresholding, as well as object correlation, and in the end k-Means performed the best.

I first tried k-Means with just the brightness values. The result was good but there was a lot of noise in the clustering and it was not taking spatial proximity into account when doing the clustering, which is important for the problem.
*INSERT PICTURE OF NAIVE K-MEANS**

I tried simple thresholding, where any brightness value over a certain amount was classified as a fiber. While this produces some good results, it still had the problem of naive k-Means in that it was not taking spatial proximity into account thus there was a significant amount of noise.
*INSERT PICTURE OF THRESHOLDING RESULTS**

I tried the EM algorithm. I found Matlab code written by Michael Chen on the Matlab File Exchange in order to run it. This is the link:
http://www.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model
Unfortunately, the EM algorithm did not perform better than k-means. It did not bring out the thin fibers in the top very well. It could be that the EM algorithm is meant to be used when the distribution follows a Gaussian model, whereas this data set may not really be following that model.
*INSERT EM PICTURE HERE**

I tried the MRF algorithm. I used the min-cut code that I did for an earlier homework. It performed well at selecting certain fibers but it did not select all the fibers that I wanted it to. This could have happened because the min cut algorithm is optimized for selecting foreground and background. The fibers do not quite have the image properties of a foreground in that they are not all in one region of the image, so using min cut is not as effective. Perhaps if I had done the algorithm but with multiple seed points then it might have worked well. Unfortunately, the number of seed points needed is not always known and their location varies between images. Since there is a large number of images in each data set, having to select seed points would be impractical. Thus, it would have not been worth trying to continue making the MRF algorithm would well with my data set.
*INSERT MIN CUT PICTURE**

I finally tried object correlation. I tried a few different templates. I used one template that was just a little square where a fiber was located. I also tried a larger middle square that had a few fibers. Unfortunately, when I did the correlation and then displayed the results as an image, it just looked like a blurrier version of the old image in both cases. It did not bring out the fibers in the way I was hoping. I did also think about using correlation for alignment but the results were not reliable enough to be trusted to use on

all the images of the data set.

Correlation likely failed due to differences in the fibers in size, shape, and thickness, making them hard to detect. **EXPAND UPON THIS MORE**

*INSERT PICTURE OF LITTLE SQUARE TEMPLATE**
*INSERT PICTURE OF LARGER TEMPLATE**
*SHOW RESULTS WITH LITTLE SQUARE TEMPLATE**
*SHOW RESULTS WITH LARGER TEMPLATE**

When I tried k-Means but with $k = 30$ and using a normalized x and y coordinate as well as brightness values, I ended up with very nice looking fibers. Around the top of the image, the fibers start to thin out so they are harder to detect, but k-Means was still able to detect them. **MENTION EARLIER THE THIN FIBERS IF POSSIBLE**

Smoothing of Segmented Images

I tried some different filters for this and in the end, a median filter and bilaterial filter gave me the best result. I used a bilateral filter because it is both noise reducing and edge preserving and takes into account both spatial proximity and brightness when doing the averaging. In this way, it would be really good at bringing out the fibers. When I implemented bilateral filtering, I used Matlab code from the Matlab file exchange. The particular code that I used was written by Douglas Lanman and the code can be found here: http://www.mathworks.com/matlabcentral/fileexchange/12191-bilateral-filtering

*INSERT MEDIAN FILTER IMAGE**
*INSERT BILATERAL FILTER IMAGE**

Making the Data Set

This describes what was done once the images were segmented.

Every time I generated slices, I did add a slice in between that consisted of pixels where both images had fibers. This was meant to bring out the fibers better in the final data set.

Alignment

I had images taken at different depths but I was not told that that they should be overlaid on top of each other so I tried to see if it would make sense if the images should be shifted slightly as I make the 3D data set.

I took an image and shifted it on top of another image. I then took the overlap region for both images and considered a pixel a fiber if both of them had fibers there. Each of the overlap regions were different sizes so I made sure to normalize the number of matches by the total number of pixels. Even after doing this, the most overlap occured when you did not do any shifting.

I did notice that the fibers tend to move up. I decided to see if correcting that would have any effect. I looked at the first image and saw the min and max row that contains a fiber. I used that height. For the rest of the images, I computed their min row and then selected the window of min to $min + height$. I compiled the slices together but in the end it did not look much different then when I did a normal overlay.

Thus in the end I decided not to use any special alignment and just put the images together.

Displaying the Data Set

I used the volume renderer VolView (http://www.kitware.com/opensource/volview.html) to view the 3D data set that I generated from the segmented images. Here are some screenshots of it displaying the volume rendering: **INSERT SCREENSHOTS OF VOLUMETRIC DATA SET**

Results and Future Work

After all was said and done, I ended up with a nice looking rendering of the collagen fibers in a rabbit cornea. I showed it to the Ophthalmologists and they thought it was a good first step in using 3D visualization to analyze the fibers. There were thin fibers at the top and I was especially pleased that those were brought out very well. Using techniques from this course allowed me to bring out the fibers very well. The volumetric rendering would have been hazy and uninformative had I not done any segmentation on the images. Since the fibers were brought out, I will be able to do isosurface extraction to make them into a mesh.

The pictures above show a data set of collagen fibers for the cornea of a rabbit. There are other data sets that we were given. I ran the above algorithms on them but the segmentation was not as successful. The above tools turned out to be the ideal choice and their parameters were tuned for the rabbit data set. Since the other data sets have images that look different though, a different set of tools and parameters with those tools are required. In addition to finding a different set of tools for segmenting a different data set, there is work to be done on improving the current segmentation. There are also further things we plan to do with the 3D data.

Work on improving segmentation

Qualitatively, my results seemed good and the ophthalmologists seemed to think they looked nice, but there is definitely room to improve the segmentation. When evaluating a clustering, I ended up relying on my subjective comparison between the clustered image and the original image to decide how well the fibers were

brought out. It would be great if I could get a few preliminarily segmented images to compare my results with so that I can have a quantitative evaluation of how well the clustering performed. I could then compare the error rates that the various methods produce and use whichever method has the least error overall.

The other problem was that I did not have a great template of what the fibers should look like. Without a good template, it was difficult to do object detection as a way to detect the fibers. As it turns out, the fibers are long thin strips but they vary in thickness. If I incorporate this information into an object detector, then I might end up with a very well segmented image.

It has come to our attention that when there are spots that are lined up in the image, there is likely a fiber there. We will use this and make sure to use a clustering method that favors cases where that happens.

Work on 3D reconstruction

The Ophthalmology department really liked seeing the fibers in the data set and their concern lies with analyzing these fibers. They want to know how many fibers there are and how they branch off each other as well as know their shape and size. In order to enable this type of analysis, the next step then will be to take the 3D data set and generate a series of meshes for the isosurface that corresponds to a voxel value of 1. This will give us a mesh for each of the fibers that we can then move around, analyze, and run simulations on.