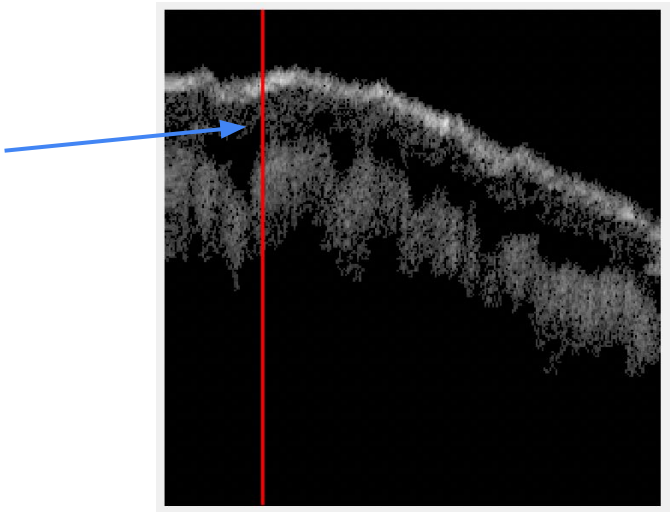
Layer Segmentation

By Andy Zhao, Bryan Zhang, and Joseph Li

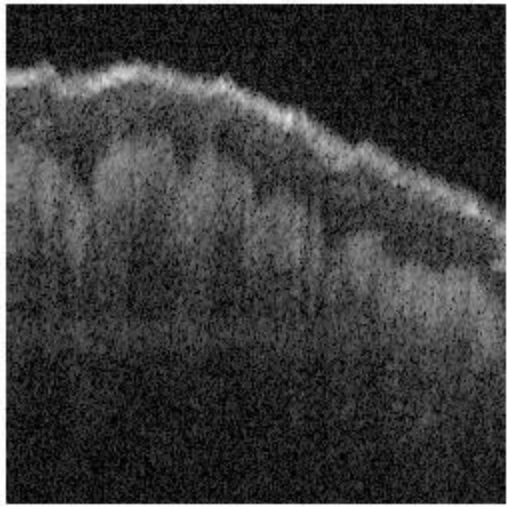
Introduction

- We want to segment a skin surface based on intensity readings of a vertical slice of the skin sample
  - Lighter colours have higher readings
- We believe that a sharp rise (large first derivative) in intensity indicates the beginning of a skin layer
- A hybrid of Sina's image processing and Xin's surface detection & smoothing algorithm.

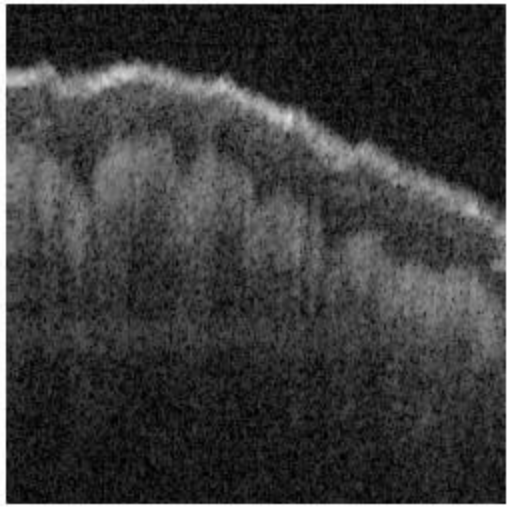


Preliminary Image filtering

```
%image filtering
I = imgaussfilt(I);
I = imadjust(I, [0.15,1]);
imbw = imbinarize(I);
im_opened = bwareaopen(imbw, 100,8); %600 default
I = im_opened .* im2double(I);
```

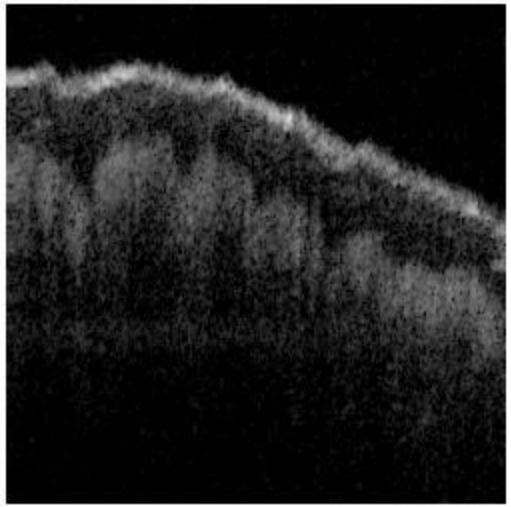


1. Original sample image



`imgaussfilt()`

2. Gauss filtering

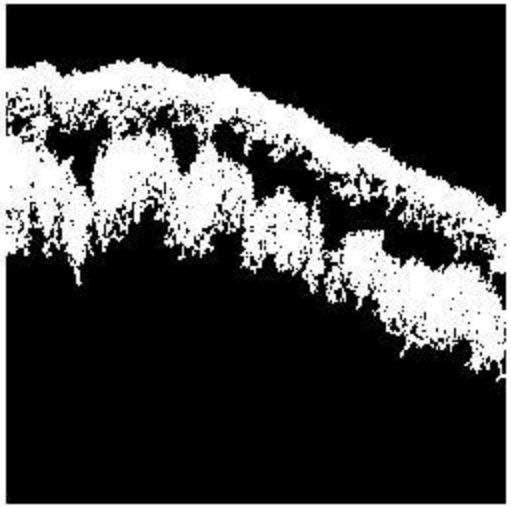


```
imadjust(I, [0.15,1])
```

3. Intensity adjustment

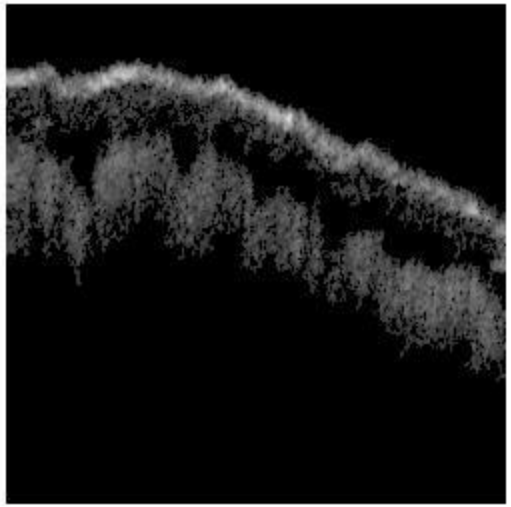
`imbinarize()`

4. binarization



```
bwareaopen(imbw,800,8)
```

5. Area opening

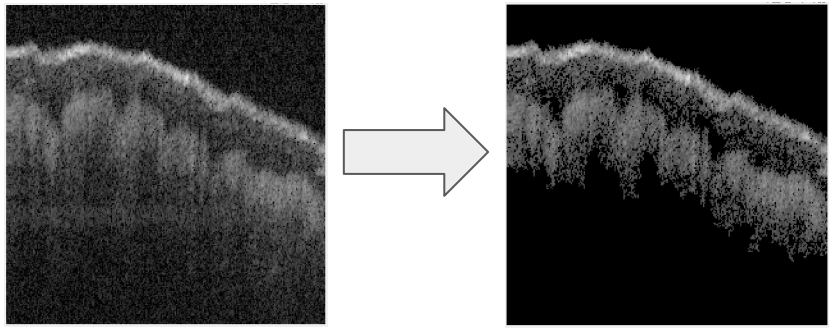


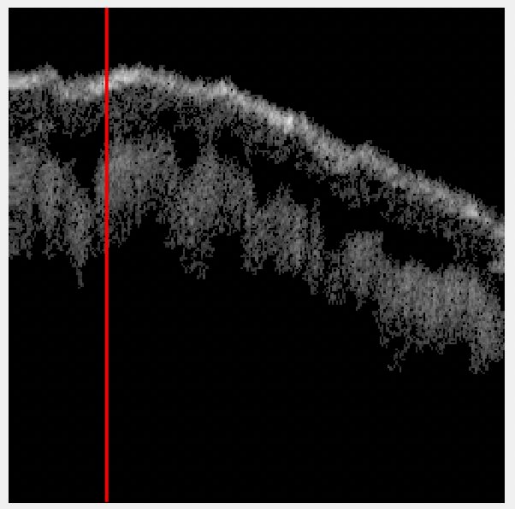
`im_opened .* im2double(I)`

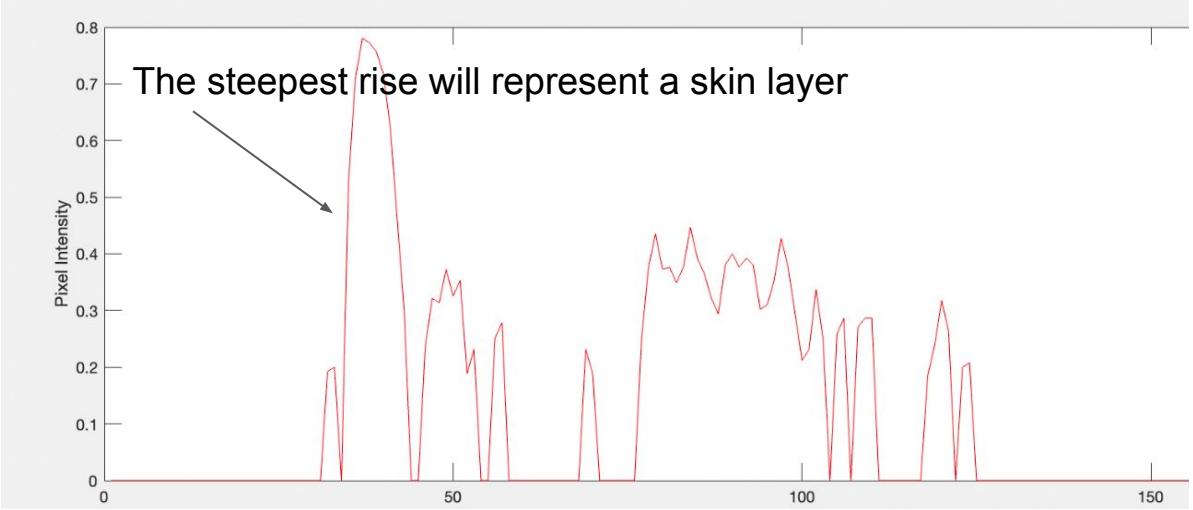
6. Final filtered image

Method:

- Filter image
- Plot pixel intensity vs pixel number for one “slice” of an image



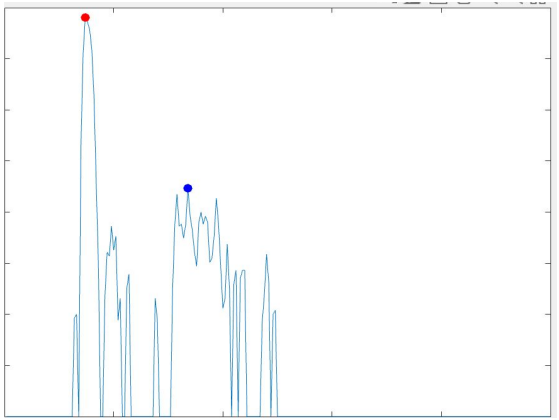




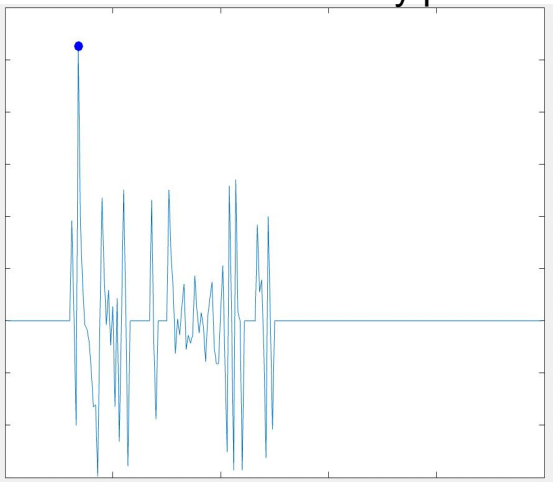
Method:

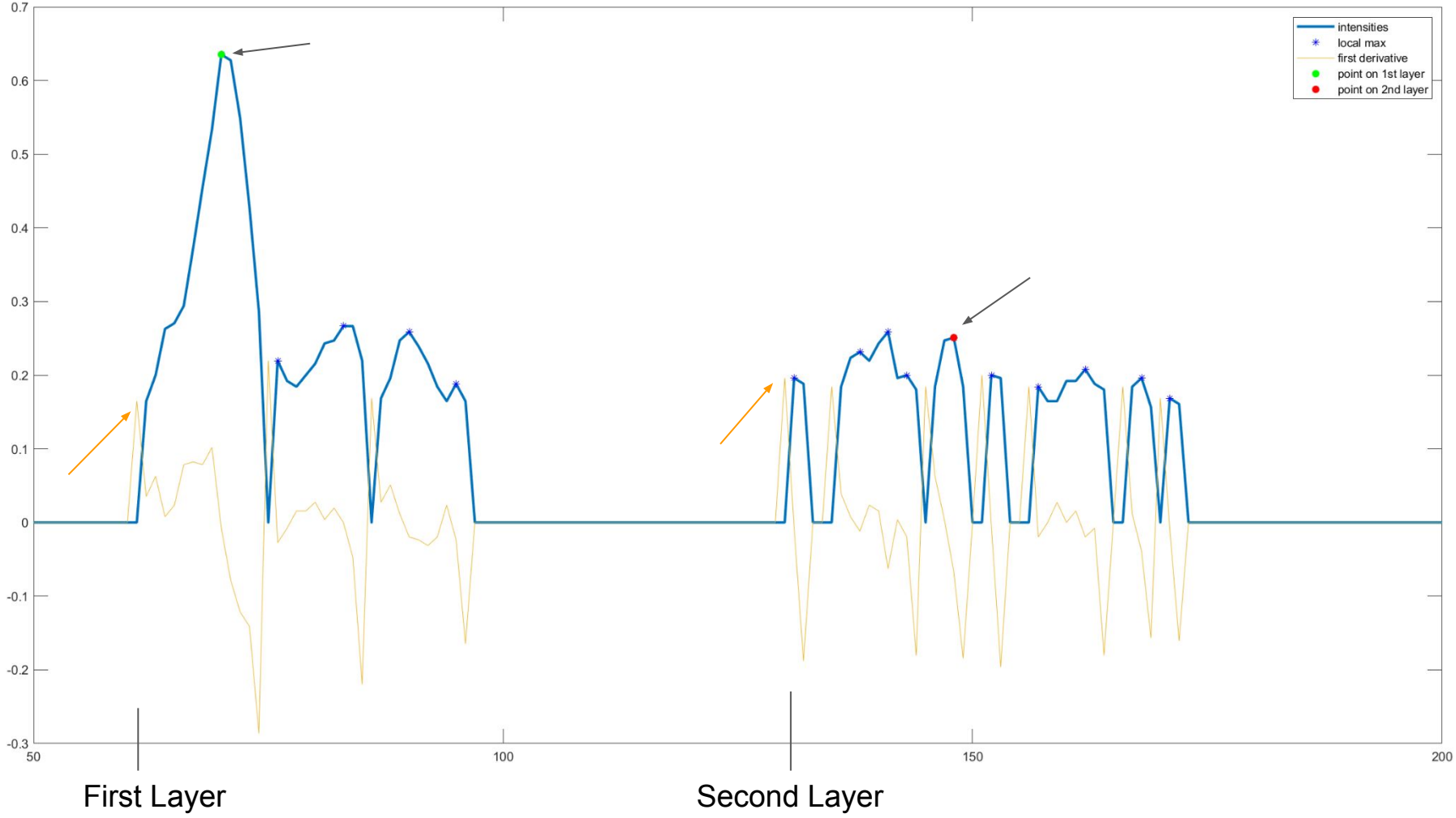
- Find the a peak in each layer.
- Search 25 pixels in front of the peak (red and blue dot in picture) to find the largest derivatives
- The x-value of the derivatives will be the start of the layers.
- Repeat the entire process for all slices in an image.
    a. About 501 times per image

Intensity vs Pixel number

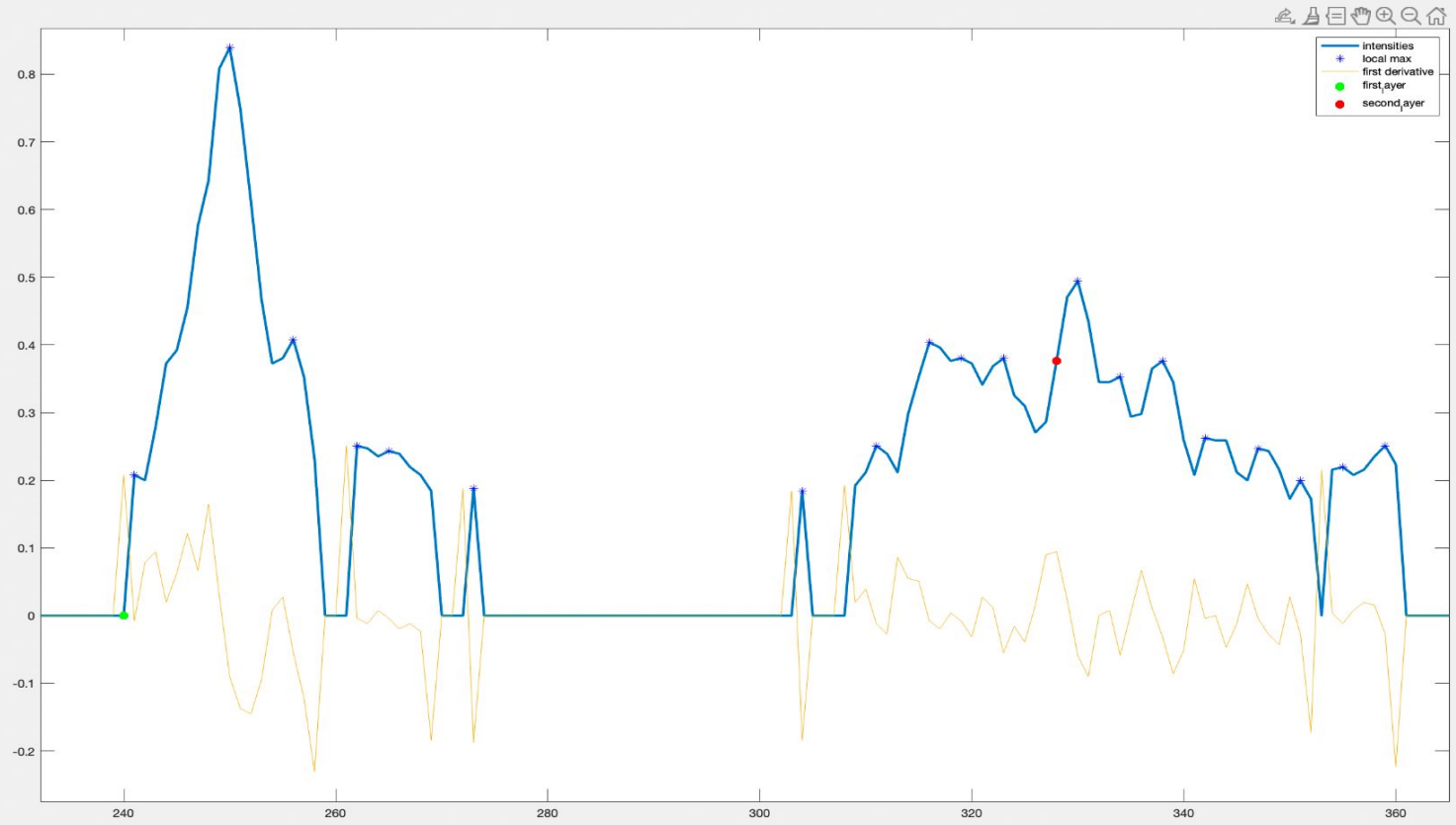


Derivative of intensity plot

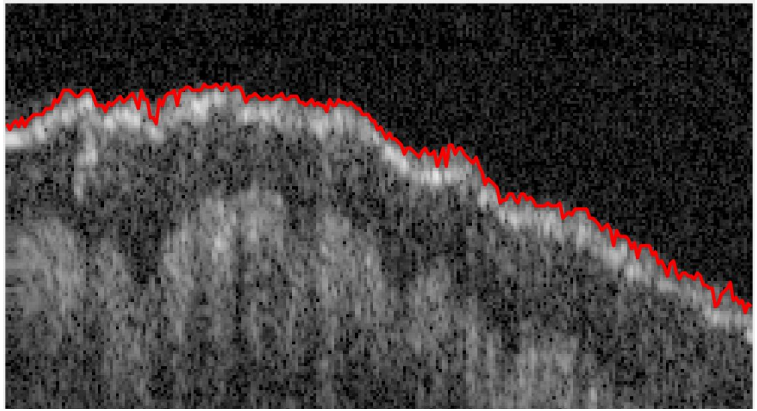


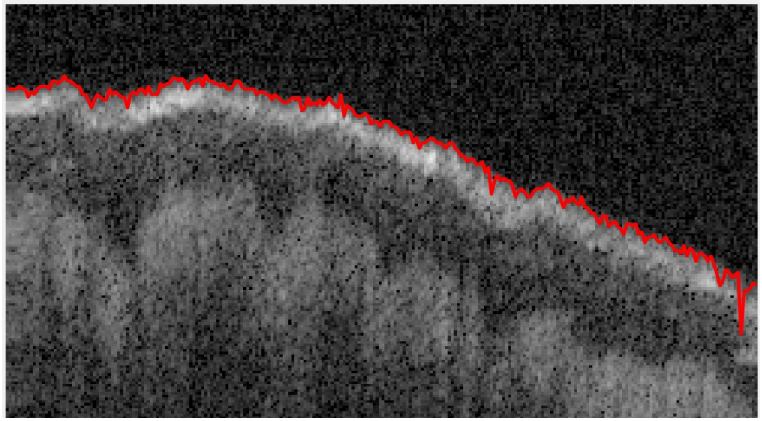

intensities
local max
first derivative
point on 1st layer
point on 2nd layer
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
50
100
150
200
First Layer
Second Layer

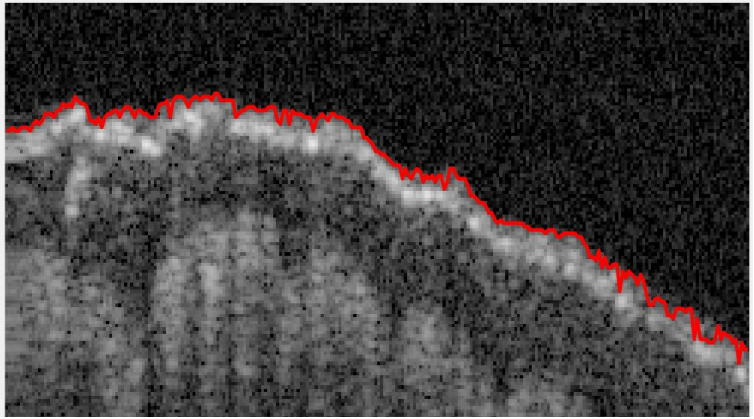
File Edit View Insert Tools Desktop Window Help

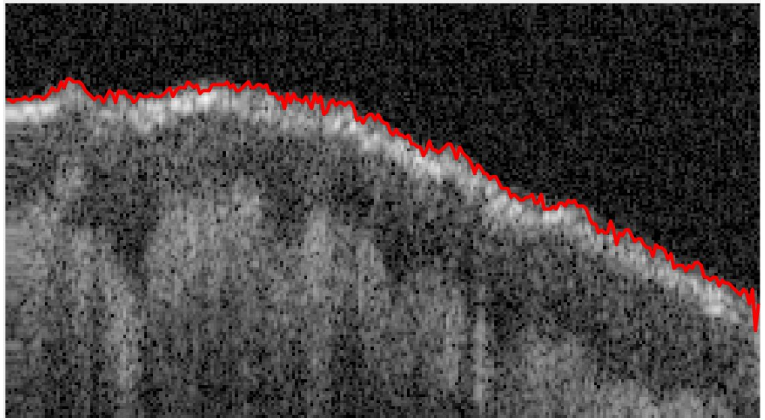


First Layer Results:

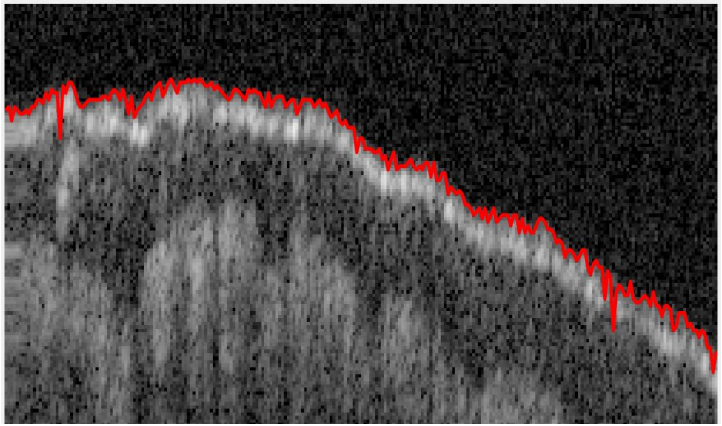


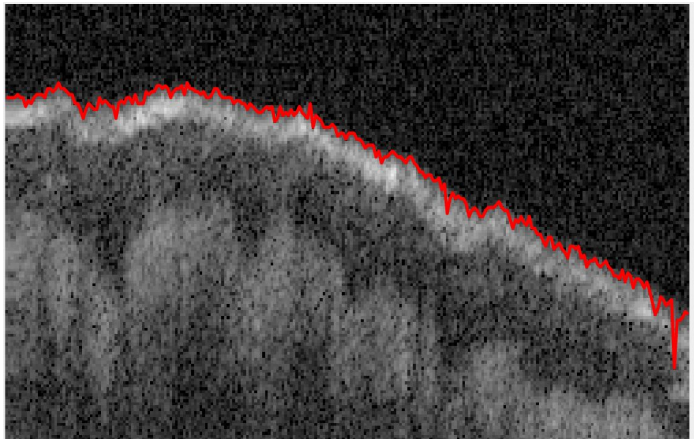


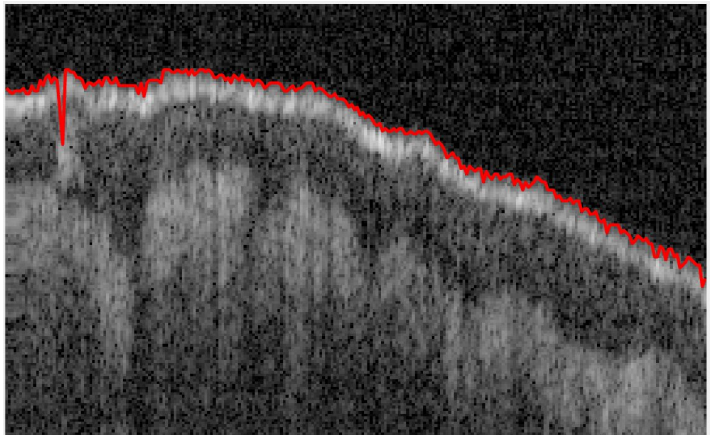


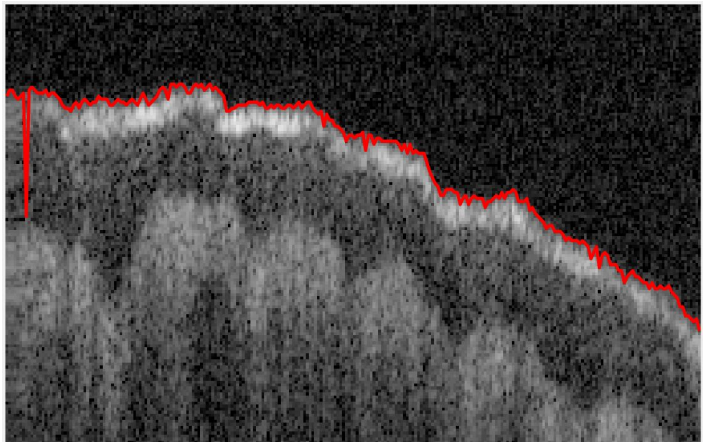


First Layer Results:



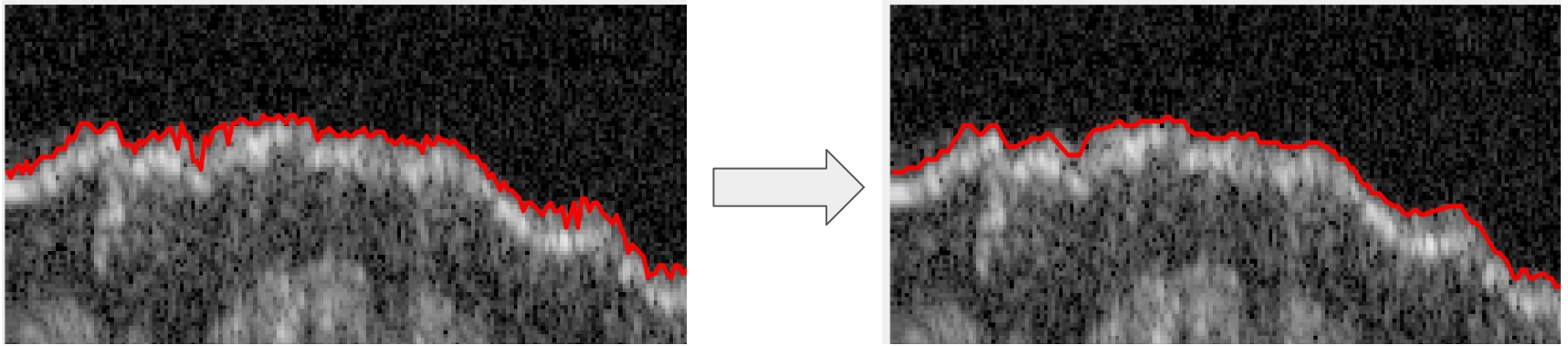




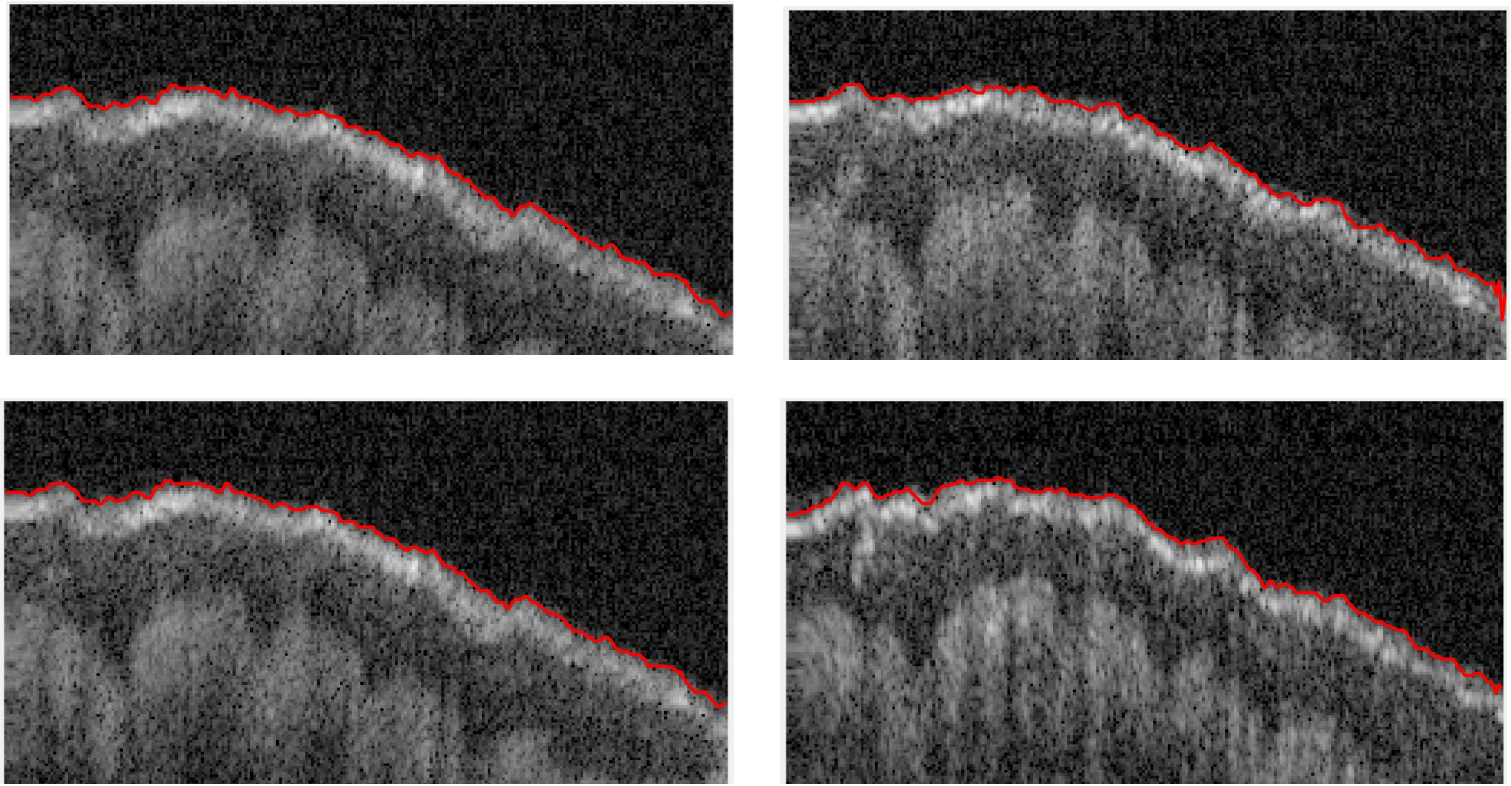


Smoothing the Surface

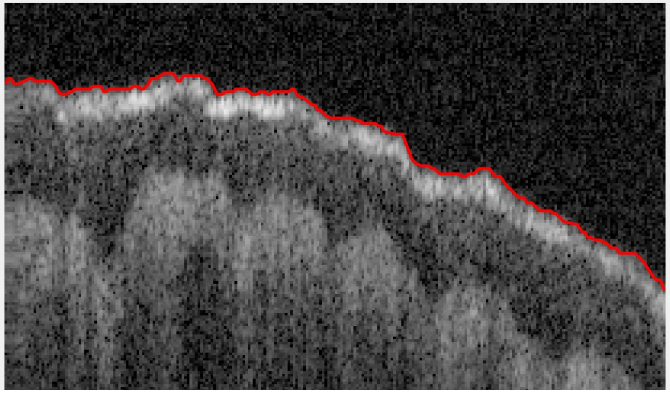
- Smooth the surface using Xin's Smoothing Algorithm
- The method is based on looking at the second order difference between points on the surface.

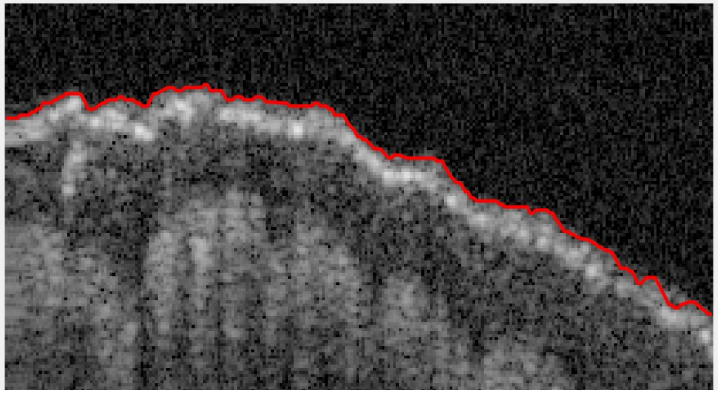


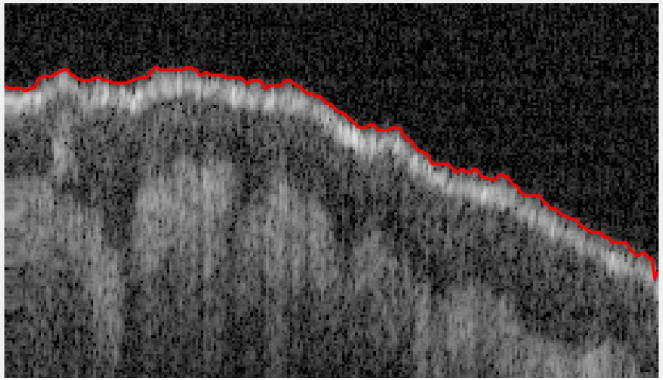
Results: (smoothened)

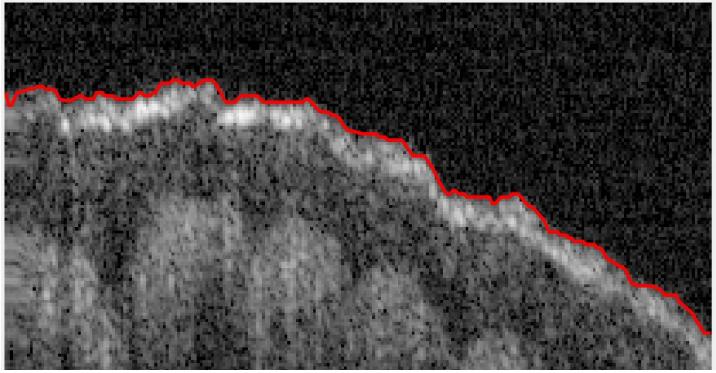


Results: (smoothened)

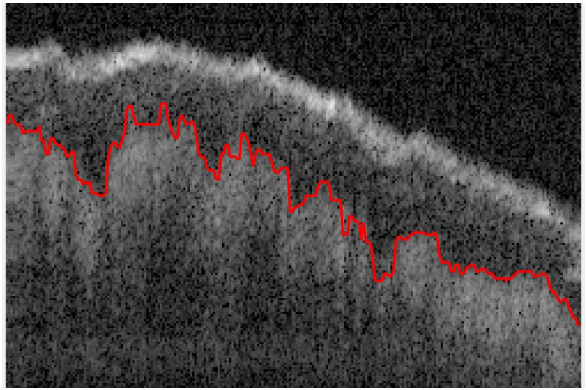


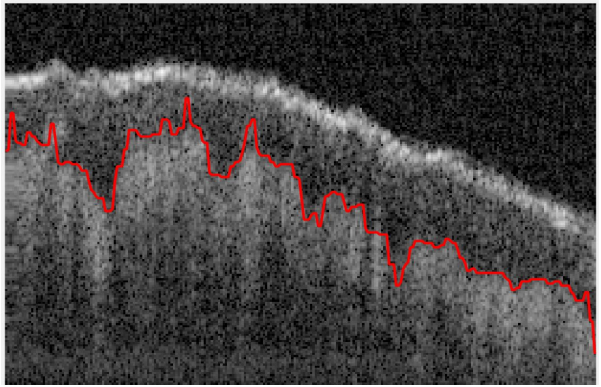


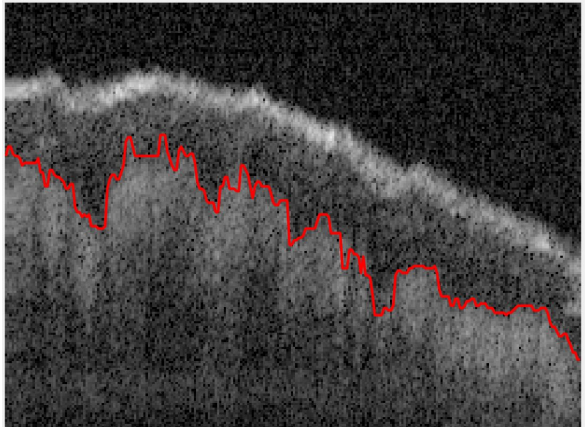


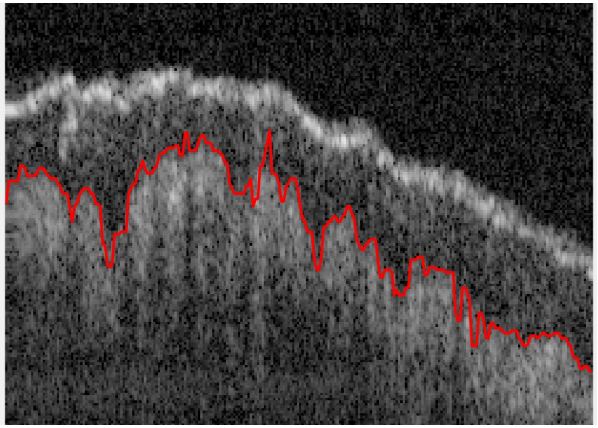


Second Layer Results:

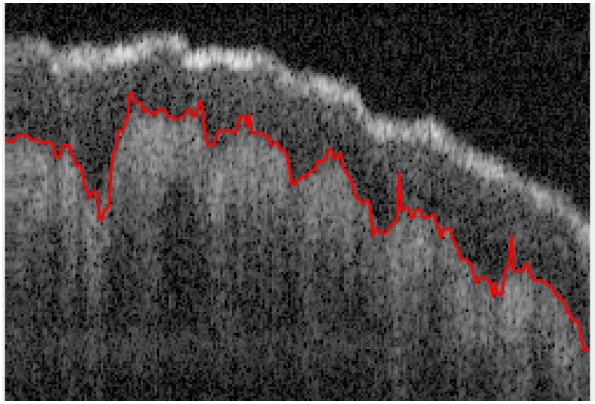


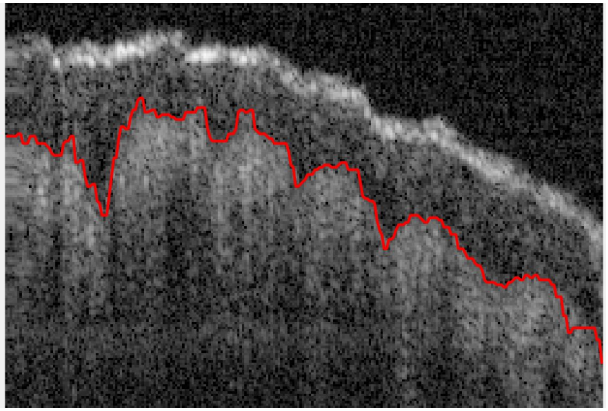


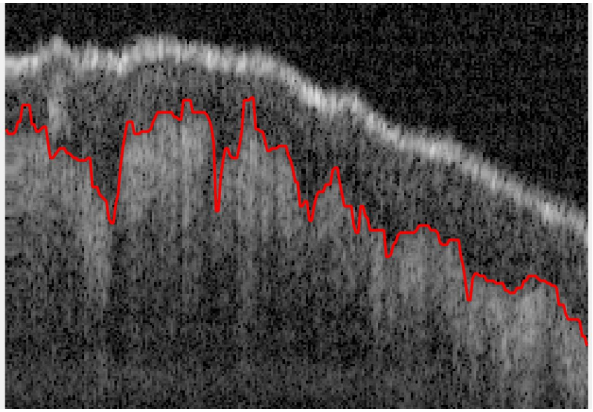


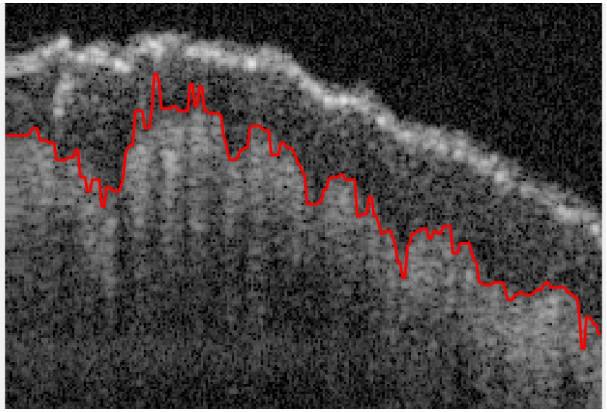


Second Layer Results:



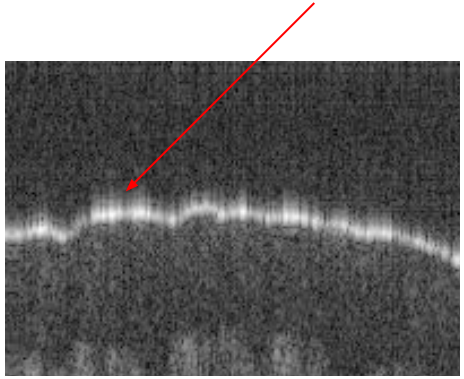






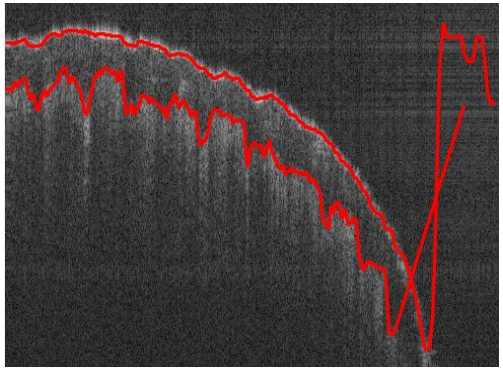
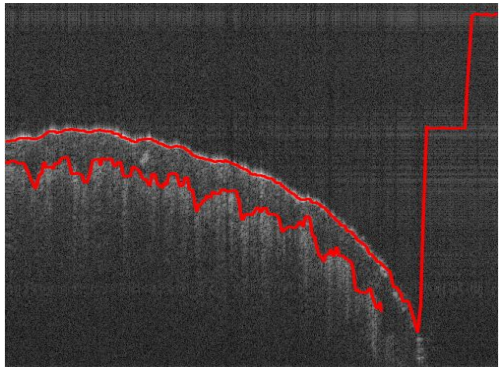
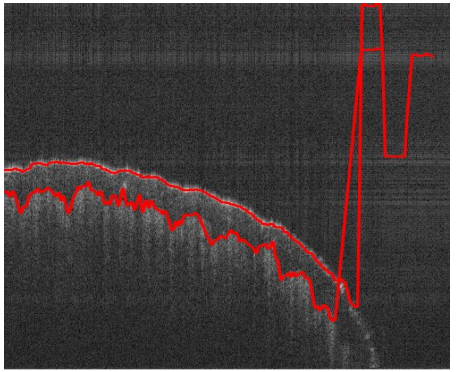
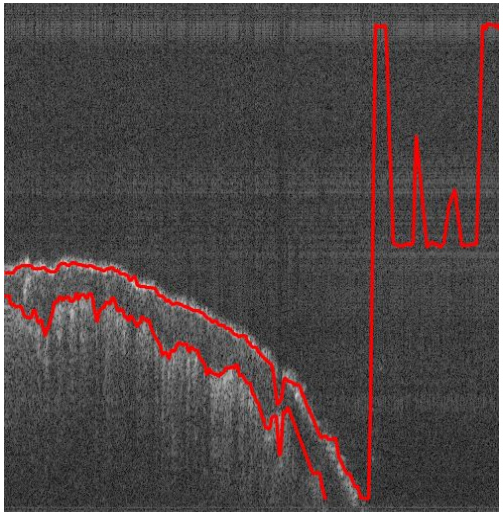
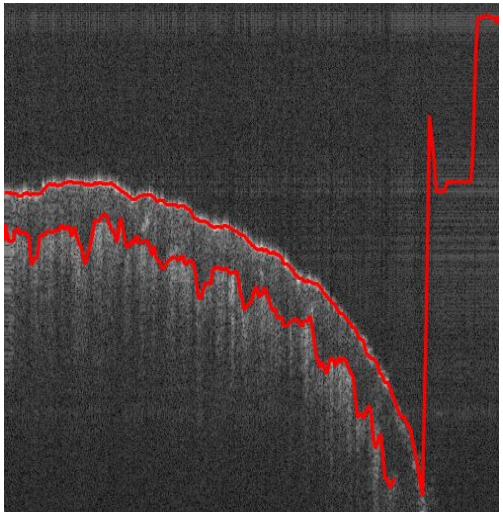
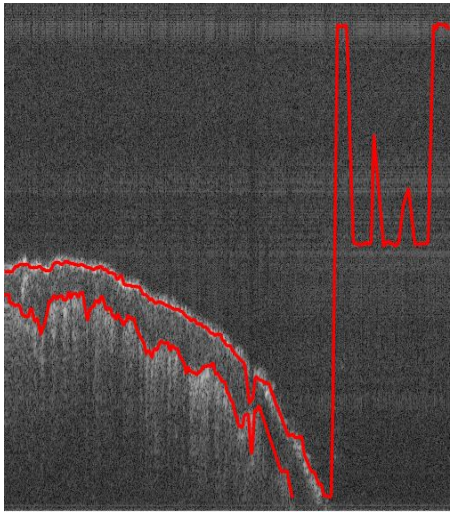
Bug fixes

1. Avoiding glare of first layer
    a. Found the highest negative derivative (end of first layer) and added the estimated length of the first layer (about 10 px)
    b. Reduce interval size when searching for minimum derivatives
2. Determine the number of layers
3. Attempt to detect whether no layers are present
4. Optimizing code speed
    a. Moved image filtering to main function
    b. Used preallocation

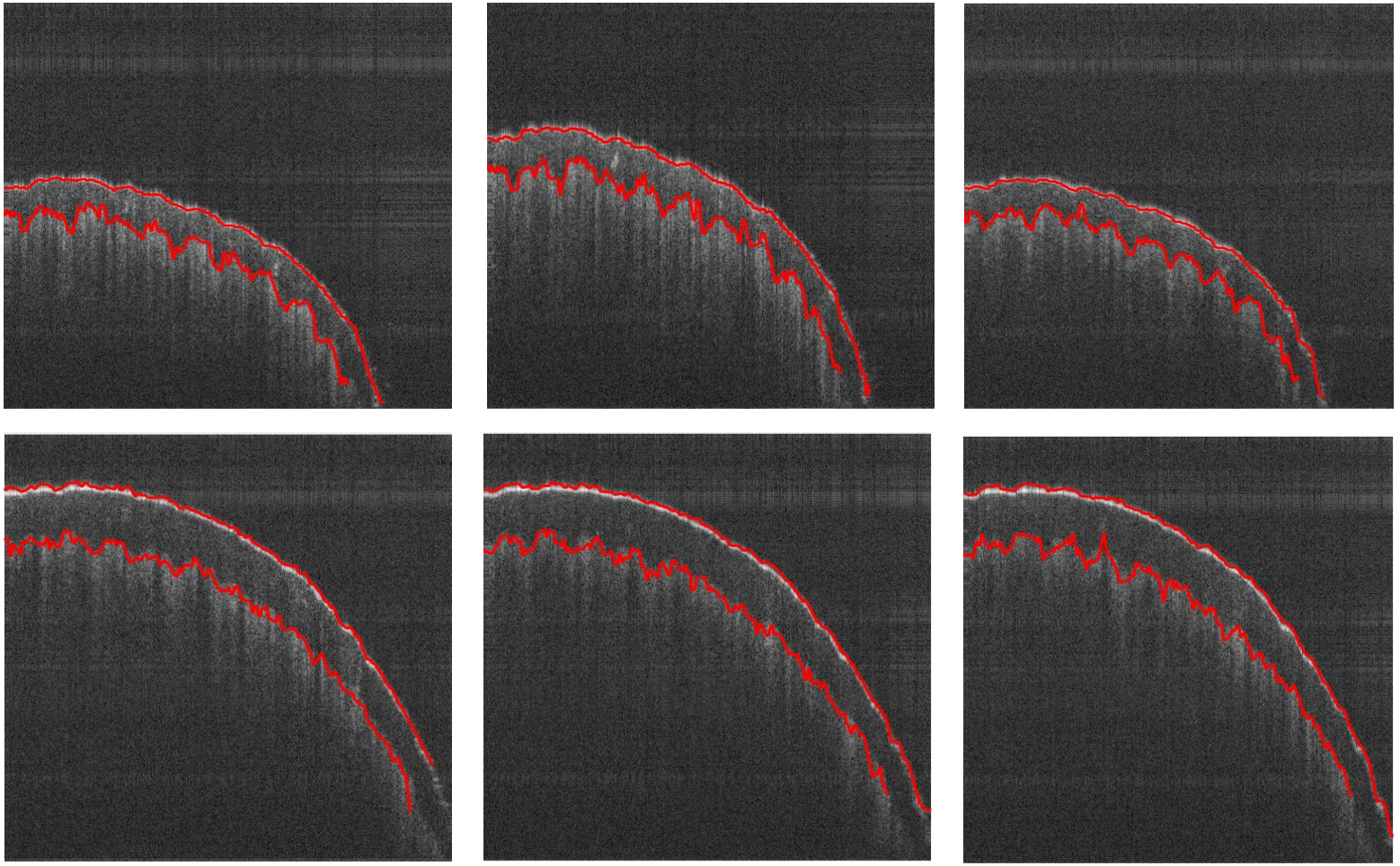


New Results

These images are considered successful

These images are perfect



Profiler

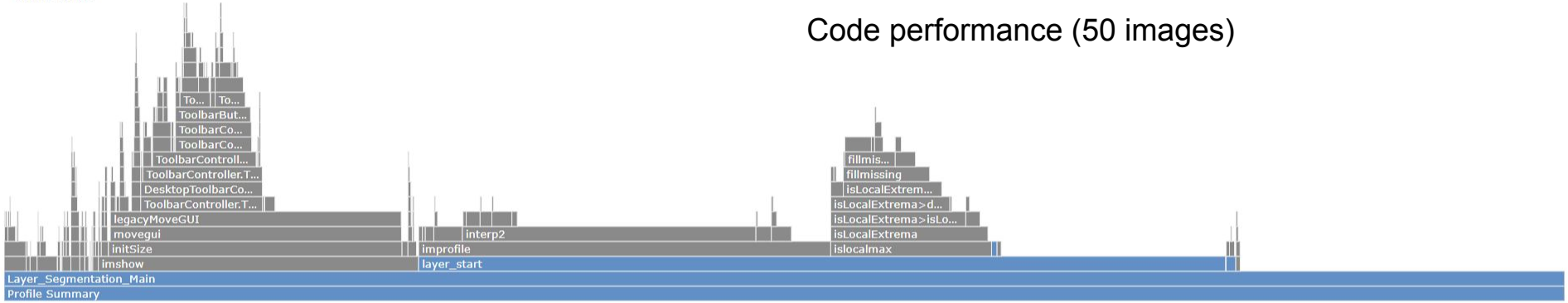
PROFILER

Print | Profile Summary | Back | Forward | Find... | Highlight | Layer_Segmentation_Main | Run and Time

FILE | NAVIGATE | SEARCH | VIEW | PROFILE

Profile Summary (Total time: 101.905 s)

Flame Graph



Generated 08-Aug-2022 14:55:57 using performance time.

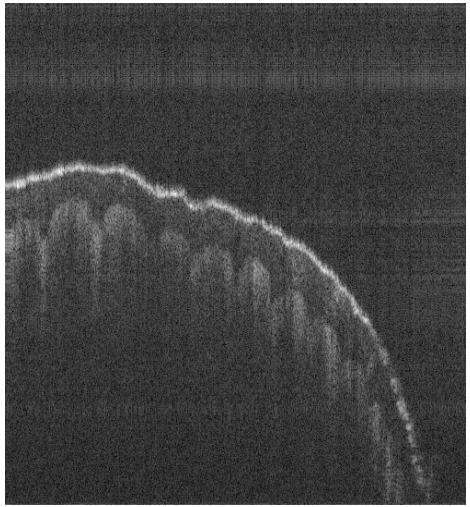
Function Name	Calls	Total Time (s)	Self Time* (s)	Total Time Plot (dark band = self time)
Layer_Segmentation_Main	1	101.061	21.191	
layer_start	25006	52.669	14.198	
improfile	25006	27.042	2.622	
imshow	49	20.069	0.123	
interp2	25006	18.983	15.353	
initSize	49	18.325	0.037	
movegui	49	18.174	0.007	
legacyMoveGUI	49	18.154	8.209	
islocalmax	24567	10.842	0.261	
isLocalExtrema	24567	10.581	0.491	
isLocalExtrema>isLocalExtremaArray	24567	9.338	0.915	
isLocalExtrema>doLocalMaxSearch	24567	8.319	0.561	
ToolbarController ToolbarController>@(e,d)obj.handleMouseMotion(e,d)	104	7.464	0.003	

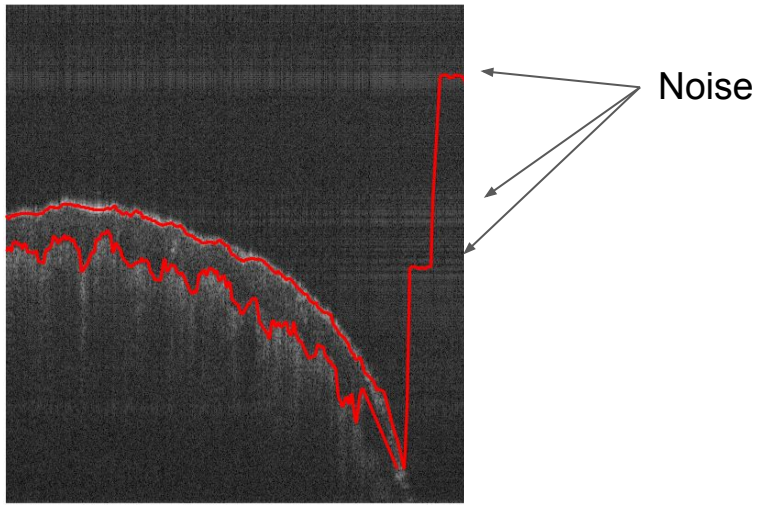
Statistics about our program

- Takes about 1 seconds to compute the two layers
- takes 1.5 to 2.5s to show image with plotted layers
- Out of 255 images:
  - 21 images had badly drawn layers
  - 2 images couldn’t be computed / had some errors
  - Rest of them were all good / had very minor issues at the end

Challenges:

- Only one dataset, so some parts are hard coded
  - Need to find a method that works autonomously for all data sets
- The second layer intensity is a bit low
  - Almost blends in with background
- Need to eliminate background noise
- current algorithm may not be optimal in terms of speed and resources wise





Solutions

-multithreading to improve speed

-reducing noise of machine and use better images

-More general thresholding instead of hard-coding (maybe via machine learning)

    - i.e. interval size, image filter, # of pixels between layers.

-Create two different image filter for each layer

Conclusion

- This method of skin segmentation is relatively easy to implement and can segment a layer very accurately, with only a few minor bugs
- Overall, the success rate is about 91% and runs at a reasonable speed
- Source code: UBC-PS-OCT/surface-segmentation (github.com)

New results with z smoothing