

Progress Report 1

Jan. 25, 2017

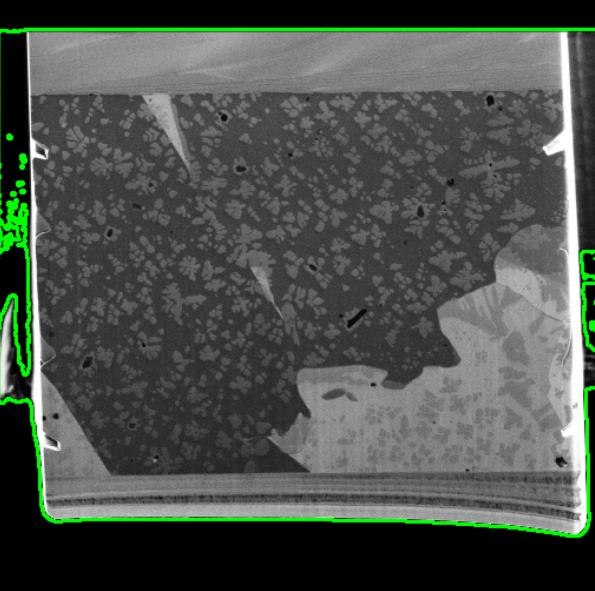
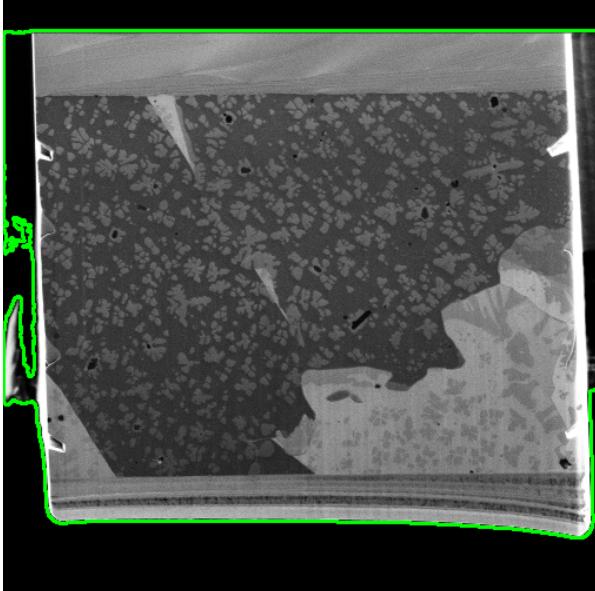
Goal: Aligning different frames, for later pair-wise comparison;

Methods: pick out outstanding features in each frame, and align these features.

Trials:

- 1) use cv2.**findContours**, to draw all contours;

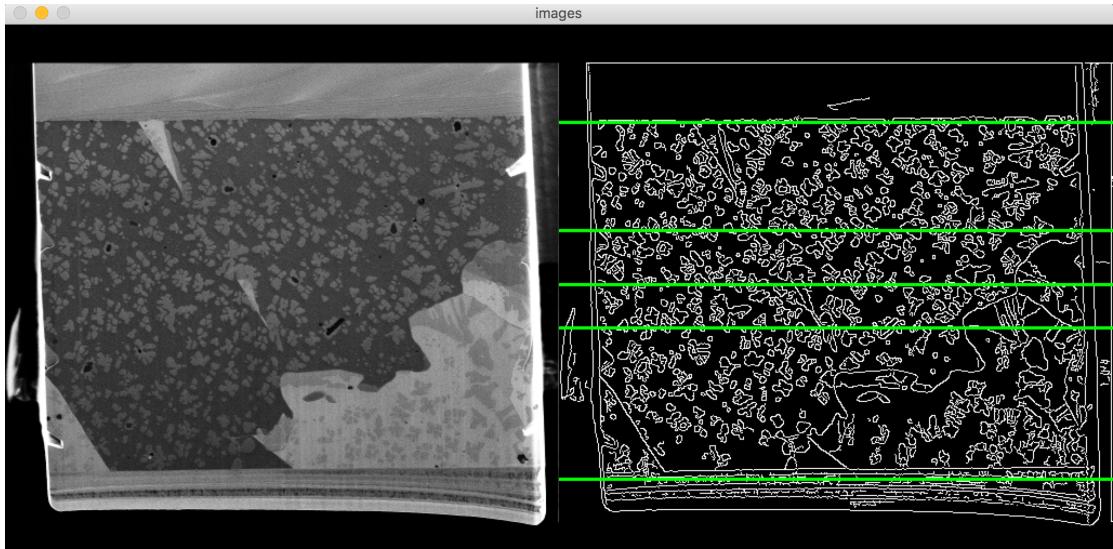
Conclusion: pick up unnecessary edges and contours;

	
1A. Drawing all the contours found in the original image; Perhaps image processing would reduce the unnecessary contours.	1B. Drawing contours of hierarchy = 1, reduced some noises, but still not ideal.

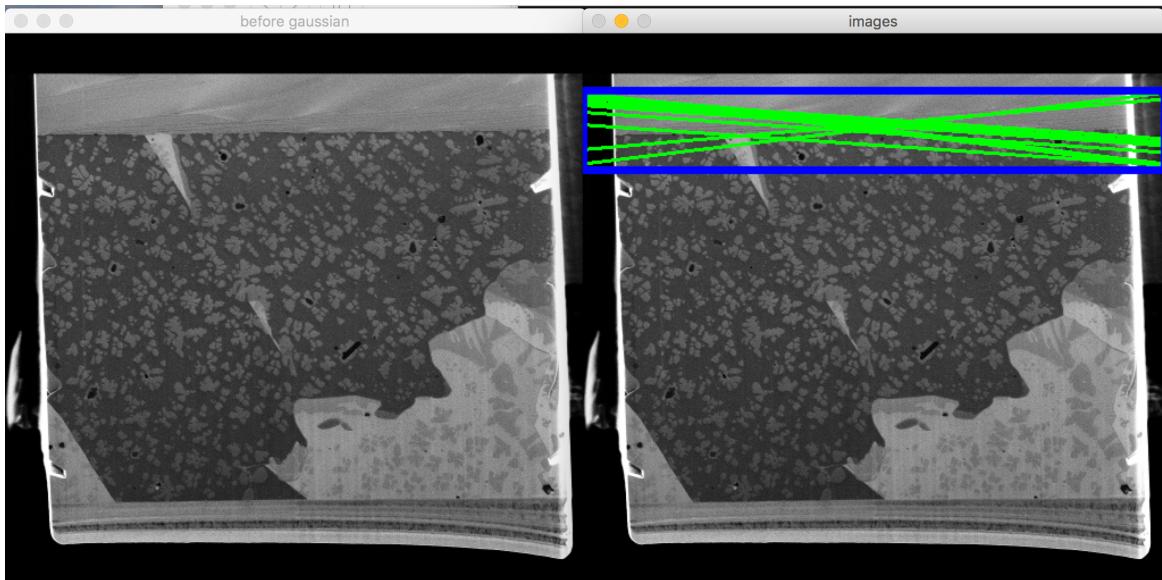
- 2) tried to draw **bounding rectangle**

Wasn't successful either. Forgot to take screenshots.

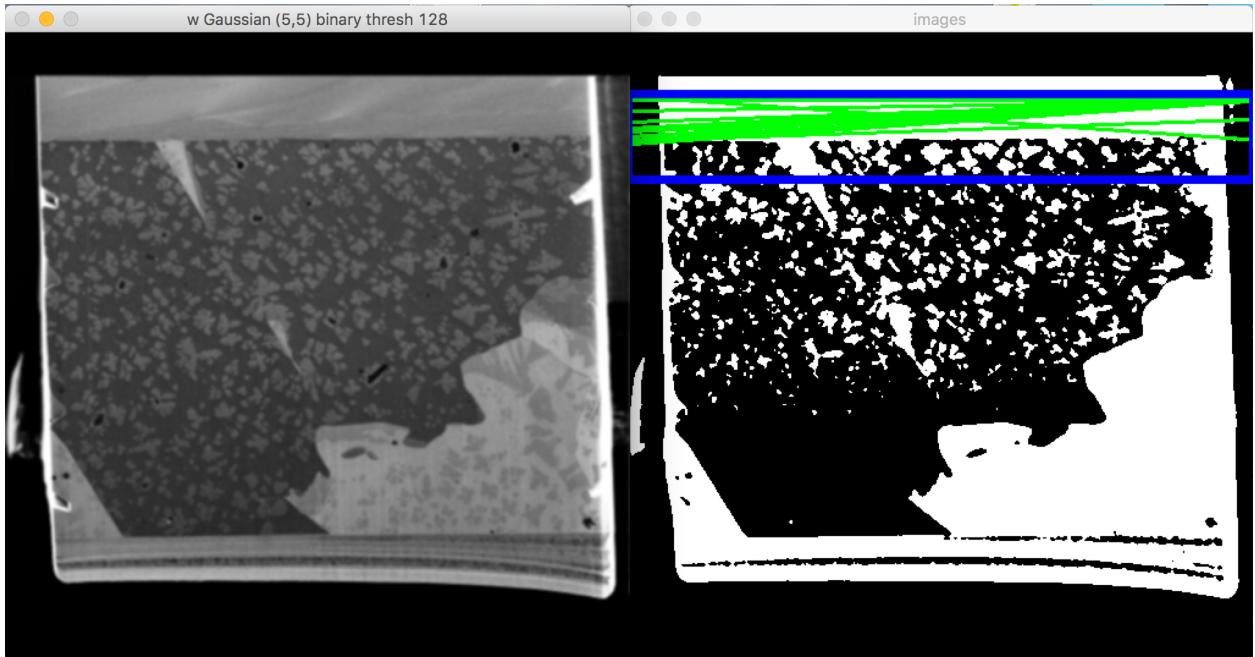
- 3) Use **Houghline Transform** to pick out lines, after certain filters / image processing;
With Sobel filter to get outlines for the entire image;
Below: left: original image; right; noises lead to unnecessary and unexplainable lines



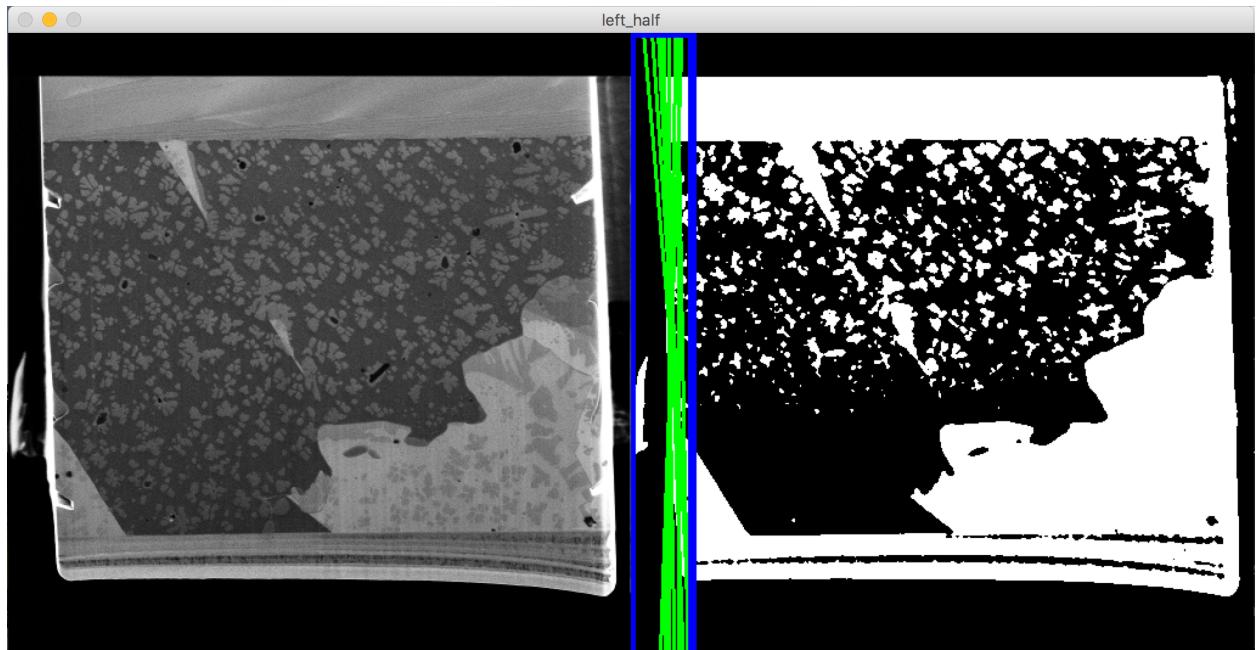
- 4) Manually select a **Region of Region of Interests (ROI)** - (shown as blue box below),
and run line detection in this region;
Aim to detect the line where the surface of interests start, right below the gray region;
Below: images processed only with Gaussian Blur and grayscale;

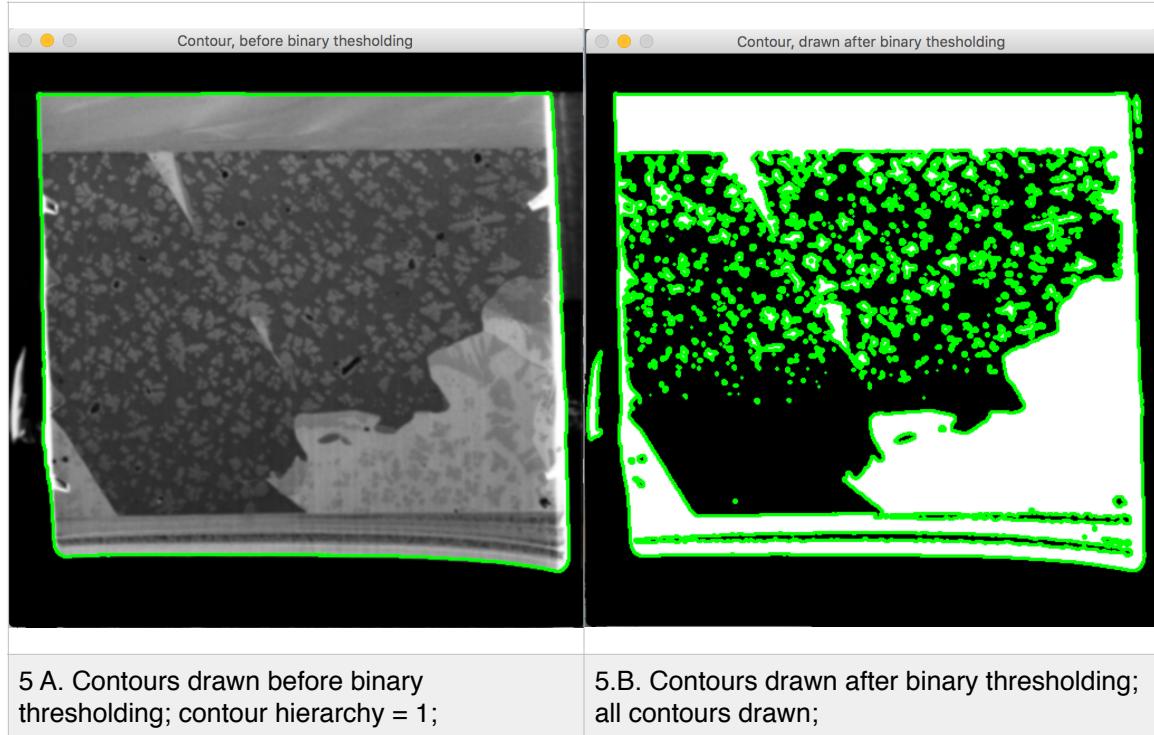


Used Binary threshold on top of Gaussian blur; contrasts is enhanced for sure, but perhaps edge wasn't coherent, and thus not detected by Houghline Transform.



5) Change the ROI to left of sample instead, hope to detect line between sample surface, and the pitch dark background. Further thoughts: lines not accurate enough; perhaps next step combine houghline results, with contours?





Next step and further thoughts:

- Methods using the contours:
 - First draw all contours in green, then use CV to selectively detect green lines using Houghline Transform; hopefully will pick out lines tangent to left and bottom contours -> alignment?
 - Can we align these contours directly between frames, without doing another layer of line detection?
- Methods using Houghlines alone;
 - Perhaps find 100 lines in boxed region, and pick out one final line based on:
 - sort them based on gradient, and consistently pick either the steepest or least steep one;
 - Merge all these lines and got a final merged line.

-> Although even slight variation would cause pair-wise comparison to fail later