Ok so today the 15th of June 2017 and a Skype with Patrick I sent him my 15th of June Patrick meeting PowerPoint presentation that summarises a lot of our previous conversations but also where I'm currently at.

I started explaining while I was busy implementing my histogram analysis on all my components in order to establish where the peaks lie (to estimate a k) but my code bombs out. I started investigating the bug whereby I reached a point where I can positively say the problem sits with where code where the various components / masks are stored in memory using C + + vectors. I did some research online discovered using C + + vectors one could theoretically run out of memory which makes the program crash. Without further confirmation if the vector component storing is the problem, I immediately started on a new implementation of storing my masks for further processing. My new mask storing implementation is writing every component or mask to a file with a specific naming convention and when I need them I just read that from the file into my code. Since that worked I then started looking at adding more source images or test images to my existing test bass.

This was based on the previous discussion between me and Patrick around having adequate test images to test every facet of my code with every instance of side scan sonar images. While adding more test images to my test based my current implementation bombed out this at the nadir removal implementation, that's based on the following: the biggest component being your background, second and third biggest component are always the left and right trips around the centerline inside my side scan sonar image. If the source images is very noisy then theoretically I can get an instance where connected component bigger than my second and third component exist, but smaller than my background component could exist there by my assumptions around removing nadir based on second and third biggest components or connected components won't work. I suspect my current bug is exactly that and because of the manner I implemented it, by make certain assumptions around the various coordinates of my images (geometric calculations) therefor getting null references. So now the logical solution is to Implement Patrick's suggestion around nadir removal.

Patrick said I must just be aware of having a proper criterion with his proposal. When walking from the centerline of the side scan sonar one pixel at a time, to the left and to the right until the pixel intensity drops or climbs over certain threshold that's where I will find the edge of the nadir. Just be aware that you might move too far to the left or to the right and fall off your image which can cause errors in your code.

The next thing Patrick spoke about is this image preprocessing stage where by he referred to another student of his having a very good speaker removal filter or median or mean filter that showed very good results. I will go and see if I can find that reference from Patrick will inform him as he will also look go look and find out. We both agree having a good image preprocessing filter before start processing the image is an important aspect. Patrick also degrees having an adequate range of test images to test every facet of my implementation is a priority. Not having this might cause issues further on into my implementations so I need to sort them out as early as possible. The end stop recording Google