Ryan McCoppin

Computer Vision

Project 4

My code runs. There is some bug as far as I can tell. It can work with more than three images, although the bug prevents correct results. I’ve spent hours trying to figure out what it is and have yet to prevail.

I would like to discuss in this report what the software *should* do as well as look at my, albeit bad, results. Hopefully this discussion will help me figure out what is exactly wrong.

The purpose of this software is to register two images together to be one overlapped image by averaging their intensity. First assign one image to be the reference image and one image to be the sensed or target image. Select k points in the reference image. Choose a window around each of k points and find the mean. Under the same window in the target image, look at each point and compute the window mean. Using these average values and windows, compute the correlation of the k points. Save the points with the highest correlation value corresponding to one of the chosen k values.

Using these correlated points, compute a, b, c, d, e, and f using the affine transformation by eliminating variables. After solving for these points, you can write a mapping that maps I(X, Y) to I(x,y). I(x,y) becomes:

I am fairly certain that my mistake is somewhere in computing the correlation. I wrote output that shows the mapping between points. And it doesn’t seem correct. Also, the correlation values are not as high as I would prefer. An example result on the 21 and 22 images:

\*\*\*\*\* Registration begin \*\*\*\*\*\*\*\*

point: 161, 194, I: 101, area avg: 92.445

area: 1600, radx: 20, rady: 20

point: 196, 49, I: 98, area avg: 100.806

area: 1600, radx: 20, rady: 20

point: 69, 100, I: 85, area avg: 97.7781

area: 1600, radx: 20, rady: 20

(161, 194) => (134, 248), r: 0.341668 🡸 These point mappings are not correct. Not sure of the reason.

(196, 49) => (223, 197), r: 0.506294

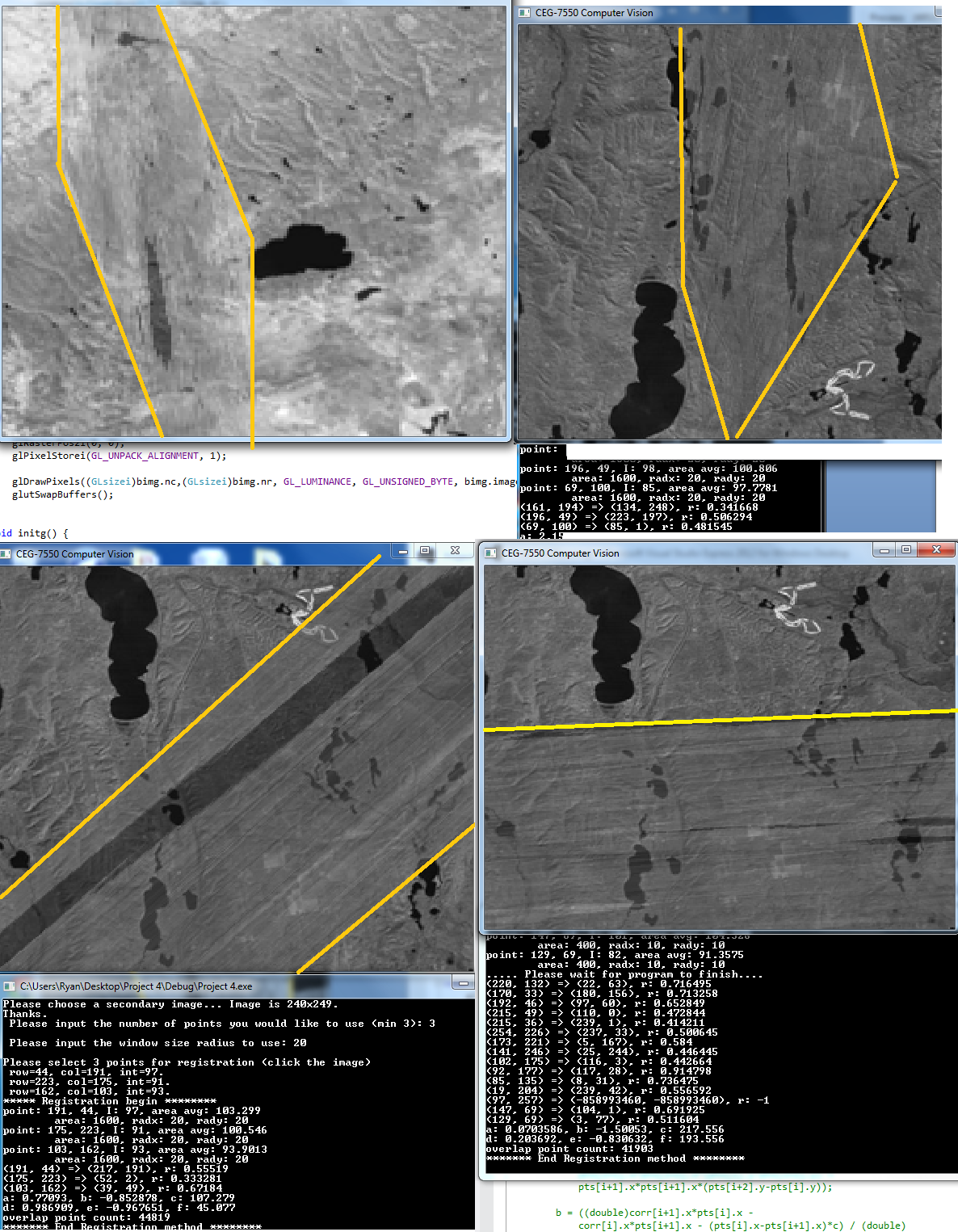
(69, 100) => (85, 1), r: 0.481545

a: 2.15737, b: -0.0930494, c: -195.284

d: 1.29027, e: 0.663169, f: -88.3882

overlap point count: 22461

\*\*\*\*\*\*\* End Registration method \*\*\*\*\*\*\*\*



The present software version is only using three points until these errors are fixed. Multipoint code is commented out. ☹ I wrote a “simpler” multi-point variation by simply sorting the points by their correlation value and using the highest correlated values for registration for temporarily until other bugs are fixed. There is an also a loop that looks at the error to decide which set of points should be used. Presently, they are only consecutive points.

The biggest known problem, that I am sure you will find inconvenient is that you can only run the program once. When you try to select points on the second run, it does not let you. I tried placing the various glut lines in various educated places to fix it, it didn’t fix it.

In conclusion, this project proved more challenging than I originally planned. Although it seems simple from a theory perspective, coding mistakes are all too simple to make. Correlation between two points is never “good enough” for my taste. After starring at my code for countless hours, I contribute the bad results to poor correlation between points. The highest correlation I have seen was 0.94, and a visual match-up never seemed like the correlations were correct.

