Cesidio Lucente CSE468-Lab 3

For the SSD alignment algorithm, what I did was use pixelwindow to crop each image channel out of the main image, storing each into its own matrix. Next I created a template out of the center of the blue image/channel, with a window of 120x120. Then I had a double loop that iterated over y then x from around 80 to 260 so the window wouldn’t touch out of bounds. At each iteration it took the SSD of first the green image/channel then the red image/channel and stored the lowest SSD and location for both. Once I had the location of the lowest SSD, I adjusted the values so it could be converted into translation x and y values for each. With this information I created a new matrix starting with the blue image and some padding. Then I added green and red in the proper location and layer to create the final image matrix, then wrote it to a file with the proper naming and printing the translations to the command window.

For NCC it’s the same as SSD until the double loop. Instead what I did was call the normxcorr2 function which takes a template (non blue image) and compares it over the entire base image(blue image) and records each NCC value and its location and stores it into a matrix. I search the matrix for the highest NCC value and its location for both the green and red. I modify the x and y vales slightly so that they will align with the x and y of the blue image with the padding, and just like in SSD I add them all into the same matrix. The matrix is written to a file and the translations are printed to the command window.

The total runtime of the program is around 1 minute and 40 seconds, and the translations are as follows (note: image 3 would never be quite right but trying to adjust for it would break the algorithms for each other image):

SSD alignments

image1.jpg x:5 y:2

image2.jpg x:4 y:2

image3.jpg x:-64 y:-59

image4.jpg x:4 y:1

image5.jpg x:5 y:2

image6.jpg x:0 y:0

NCC alignments

image1.jpg x:3 y:0

image2.jpg x:2 y:0

image3.jpg x:5 y:1

image4.jpg x:2 y:-1

image5.jpg x:3 y:0

image6.jpg x:-2 y:-2