

CS543/ECE549 Assignment 1

Name: Hanwen Zhang

NetId: hanwen6

Part 1 : Implementation Description

Provide a brief description of your implemented solution, focusing especially on the more "non-trivial" or interesting parts of the solution.

- What implementation choices did you make, and how did they affect the quality of the result and the speed of computation?

For both basic and multiscale alignments, I tried the sum of squared differences (SSD) and normalized cross-correlation (NCC) as two kind of metrics to score how well the images match. And find that the processing speed of SSD is much faster than NCC in both basic alignment and multiscale alignment; in basic alignment for jpg images, SSD's alignment speed is around 4 times faster than NCC with same output quality; in multiscale alignment, SSD's alignment speed is around 4.8 times faster than NCC with same output quality. However, the test images provide in homework 1 do not actually have the same brightness values. It is necessary to preprocess the images that need be aligned before the processing SSD alignment. I used canny edge detection filter to convert the B, G, R channel images to black/white edge images that do not contain brightness value. And use SSD to align the edge images and get the offsets. And flowchart 1 below is for basic alignment, and flowchart 2 below is for multiscale alignment.



Fig 1. The flowchart of basic SSD alignment

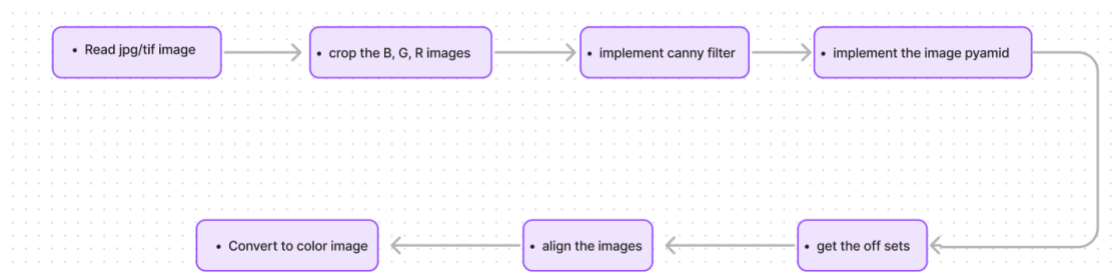


Fig 2. The flowchart of multiscale SSD alignment

- What are some artifacts and/or limitations of your implementation, and what are possible reasons for them?

Use SSD as the metric to score how well the images match is much faster than NCC. However, it is different for SSD to deal with the images that contain brightness values. An image preprocessing for image normalization before run SSD alignment is required. Otherwise, the alignment offsets will not be accuracy for align the B, G, R images, and the output images will not be expected.

Part 2: Basic Alignment Outputs

For each of the 6 images, include channel offsets and output images. Replace <C1>, <C2>, <C3> appropriately with B, G, R depending on which you use as the base channel.

A: Channel Offsets

Using channel as base channel:

Image	<G> (h,w) offset	<R> (h,w) offset
00125v.jpg	<G> (2, 5) offset	<R> (1, 10) offset
00149v.jpg	<G> (2, 4) offset	<R> (2, 9) offset
00153v.jpg	<G> (3, 7) offset	<R> (4, 14) offset
00351v.jpg	<G> (1, 4) offset	<R> (1, 13) offset
00398v.jpg	<G> (3, 5) offset	<R> (4, 11) offset
01112v.jpg	<G> (0, 0) offset	<R> (1, 5) offset

B: Output Images

Insert the aligned colorized outputs for each image below (in compressed jpeg format):



Fig 3. 00125v.jpeg



Fig 4. 00149v.jpeg



Fig 5. 00153v.jpeg



Fig 6. 00351v.jpeg



Fig 7. 00398v.jpeg



Fig 8. 01112v.jpeg

Part 3: Multiscale Alignment Outputs

For each of the 3 high resolution images, include channel offsets and output images. Replace <C1>, <C2>, <C3> appropriately with B, G, R depending on which you use as the base channel. You will also need to provide an estimate of running time improvement using this solution.

A: Channel Offsets

Using channel as base channel:

Image	<G> (h,w) offset	<R> (h,w) offset
01047u.tif	<G> (24, 26) offset	<R> (38, 74) offset
01657u.tif	<G> (8, 56) offset	<R> (12, 118) offset
01861a.tif	<G> (36, 72) offset	<R> (60, 148) offset

B: Output Images

Insert the aligned colorized outputs for each image below (in compressed jpeg format):



Fig 9. 01047u.jpeg



Fig 10. 01657u.jpeg



Fig 1861a.jpeg

C: Multiscale Running Time improvement

Report improvement for the multiscale solution in terms of running time (feel free to use an estimate if the single-scale solution takes too long to run). For timing, you can use the python `time` module, as described in the assignment instructions.

The running time of multiscale solution for tif image is around 2.4 times faster than basic alignment solution. When using SSD basic alignment method, it will take around 66 second to colorize each image, and it will only take around 27 second to colorize the image when running SSD multiscale alignment method.

Part 4 : Bonus Improvements

Post any extra credit details with outputs here.