

COL780:A2: Corner Detection And Matching

Sidharth Agarwal 2019CS50661

March 7, 2023

Libraries Used:

OpenCV, tqdm and standard python libraries

Run:

To run : **python 2019CS50661_A2.py {input_dir}**

You can set the flags of whether to plot corners and randomly matched corners using the global variables mentioned on the top. You can also change the parameters of filtering to vary speed of processing based on different datasets. Final panorama is saved at **output.png**

Q1

I implemented the hessian corner detector by taking the hessian of the gaussian smoothed image using opencv derivative and gaussian kernel functions. I later rendered a circle of yellow color on each of the detected corners.

Q2

I first detected corners in both the adjacent images. Now I iterated over corners of the first image and took the corner closest to this corner in the second image and calculated the ssd distance between patches of size 5 surrounding these two corners. If this is less than some threshold then I added this pair to matched set of tuples.

Q3

I first calculated the homography matrices based on the source and destination of matched corner pairs between adjacent images. Then using these homography matrices I calculated the affine matrices of each image with respect to the first image. Then I calculated the dimensions of the final panorama using the affine transformations over the corners of each image. Now at last I calculated the wrapper images using wrapAffine function which after stitching gave me the final panorama.

Figure 1: Corners detected for Q1



Figure 2: Example of randomly matched corners

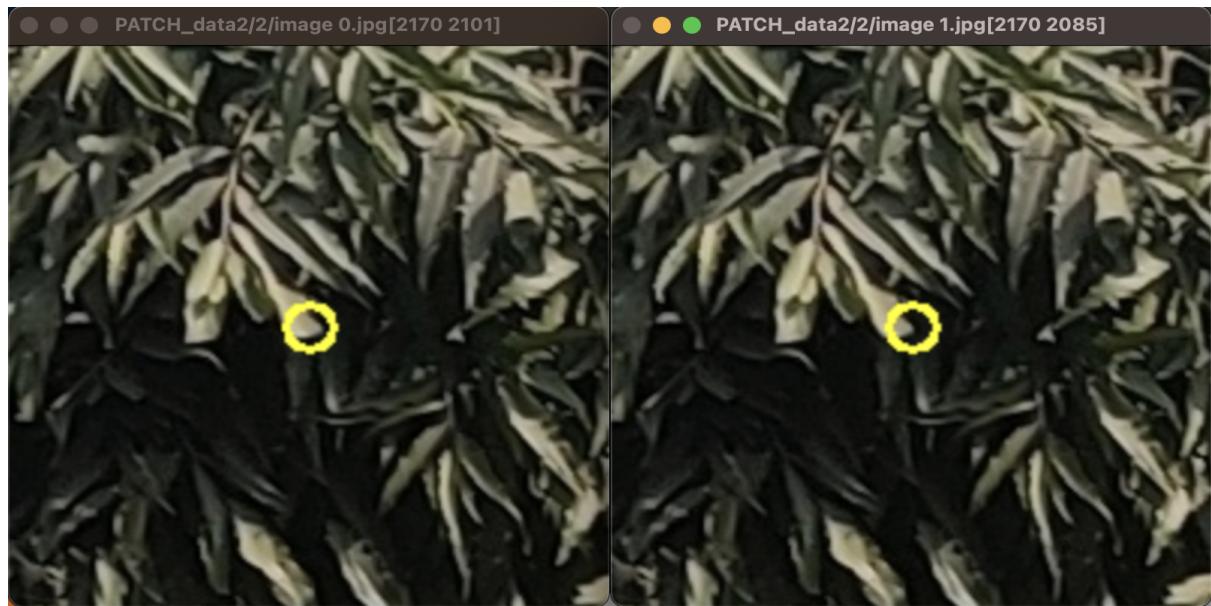


Figure 3: image0 – image 14 – panorama

