Summary of Registration

The homework is to track the template in several frames of a video. The functions are as follow:

1. x1, x2 = find_match(img1, img2)

I use the SIFT in OpenCV to detect feature points and the NearestNeighbors to find the first near match d_1 and the second near match d_2 of these feature points in the template and the target image. d_1 will be kept if $d_1 / d_2 < 0.7$.

2. A = align_image_using_feature (x1, x2, ransac_thr, ransac_iter)

I use the RANSAC method which randomly picks 4 SIFT matches to compute the best affine transformation, I set the ransac_thr as 5 and ransac_iter as 500. The visualization is shown in **FIG.1**, the red matches are good ones, the yellow dots are the affine transformation of the four corners of the template image.

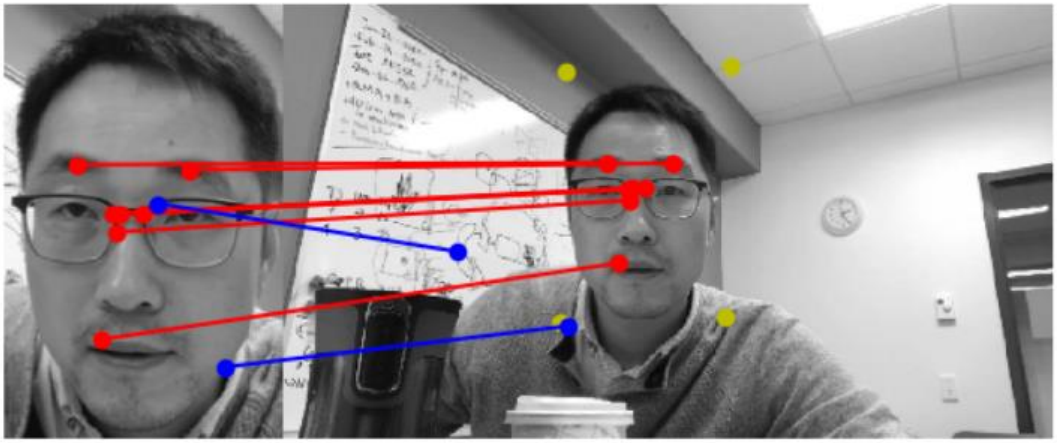

FIG.1 SIFT Matches and RANSAC

FIG.2 warped image

3. img_warped = warp_image(target, A, template.shape)

A is an affine transformation to align the template and the target. **FIG.2** is the warped image of the affine transformation obtained in the RANSAC part.

4. A_refined, errors = align_image(template, target, A)

In this function, A is refined by the inverse compositional image alignment algorithm. The way to compute the gradient of the template is same as the last homework, the steepest descent image is obtained by multiplying the gradient with Jacobian.

Then, warp the target with P (initialized by A), and calculate the error image. After that, I use Numpy.linalg.lstsq to calculate the delta P, the method Numpy.linalg.lstsq(A, B) is to solve the x in Ax = B where A, B are matrices. P is updated by multiplying the inversed W(delta P).

FIG.3 is the visualization of the comparison before and after the image alignment of the template and the first target, **FIG.4** is the error map. I use the Euclidean distance to compute the error image value.

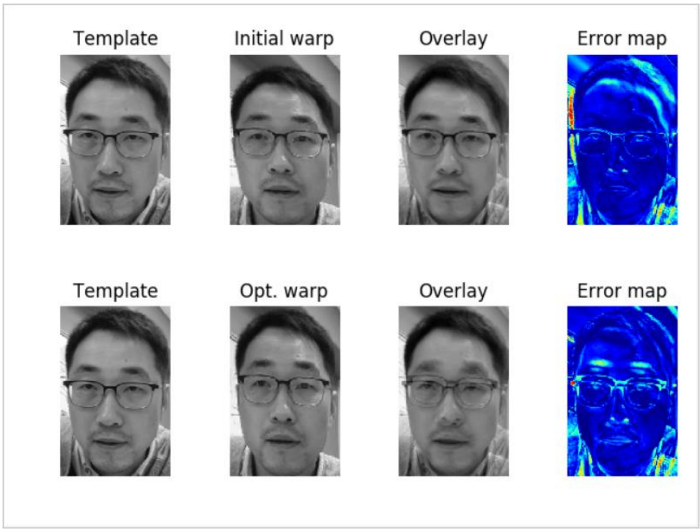

FIG.3 image alignment comparison

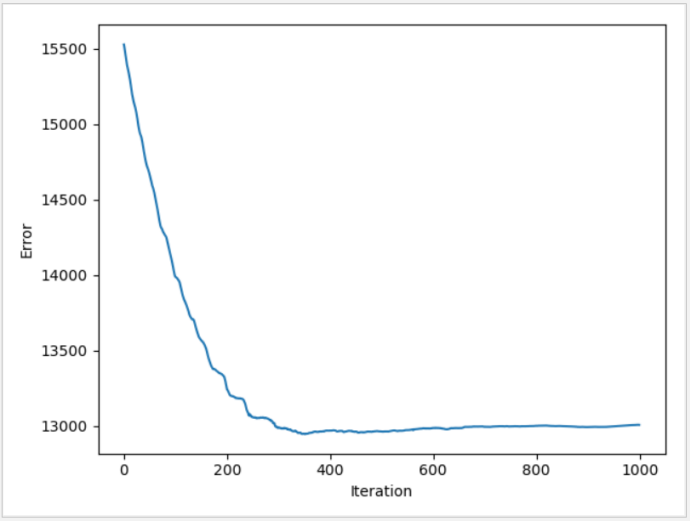

FIG.4 error map

5. A_list = track_multi_frames(template, target_list)

This function is the collection of all the functions above. For the first frame, A is obtained by SIFT and RANSAC with the template and refined by the inverse compositional image alignment algorithm. For the other frames, the template is the warped image of the previous frame, and the initial A is the one refined in the algorithm in the previous frame.

FIG.5 is the visualization of multi-frame tracking, while there are 400 iterations in the function align_image.

FIG.6 is the visualization of multi-frame tracking, while there are 1000 iterations in the function align_image.

It takes about 20 minutes to compute 1000 iterations.

FIG.5 400 iterations in align_image

FIG.6 1000 iterations in align_image