CSE472 Assignment 2

Yunhoe, Ku

Due date: November 11st, 2021, before the class

1 Feature Detection

1.1 Task 1

In this problem, I use some parameters specified in our documents. The result is like this:

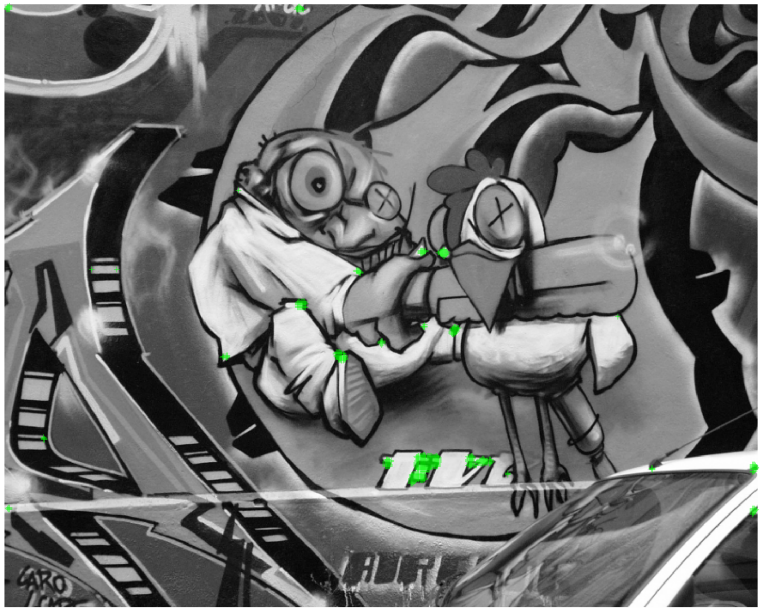


Figure 1: Harris operator result without non-local maxima suppression

Because there is no processing, we can see the **clustered** dots within some points like knee, shoe nose, corner of letters. They have similar features: there are some areas or near the areas that the amount of changes of gradient is greater than the others.

1.2 Task 2

Because above result is so clustered, we can't use it directly because we should make more corner points to match two images. To do make that, we should remove all the things except for **the local maximum**. So I take 21-by-21 window to check the value of center is whether maximum or not. The result is like this:

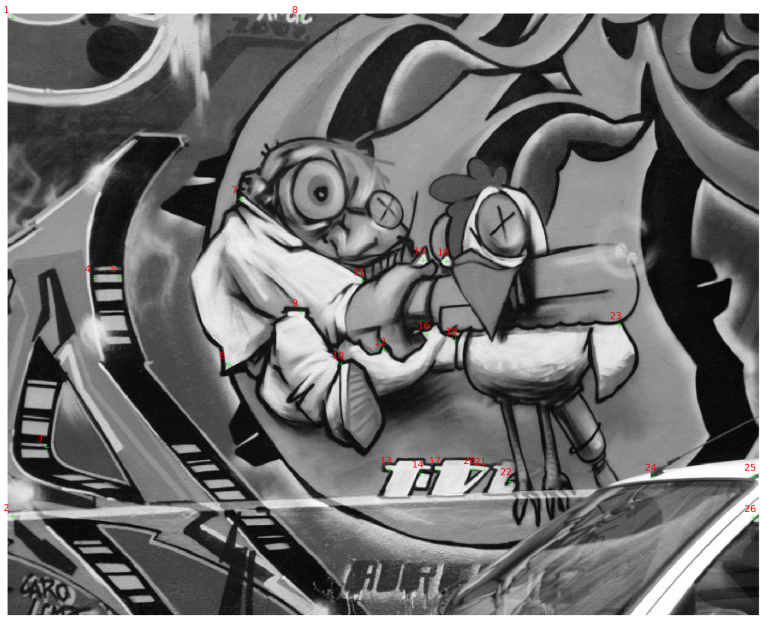
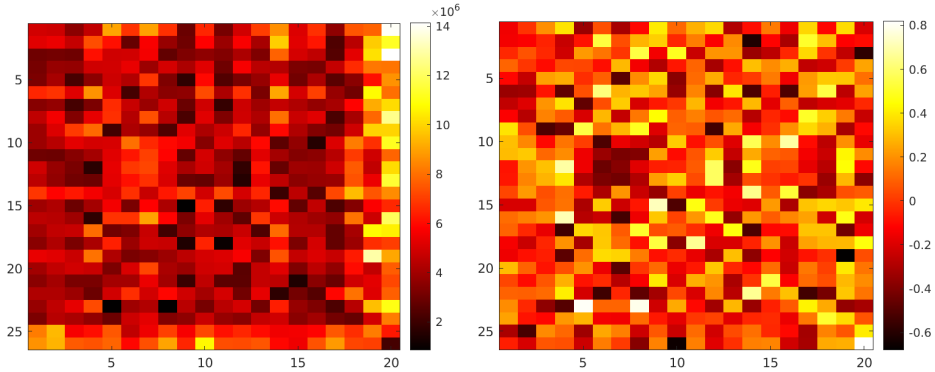


Figure 2: Harris operator result with non-local maxima suppression

1.3 Task 3

With those result, we can extract some features from images. Now, we can take *SSD(Sum of Square* or *NCC(Normalized Cross-Correlation* to check the errors between two images so that we can find that which one is the best matching with someone. Anyway, here is the result of using SSD between feature between im1 and im2:



(a) Result with SSD (b) Result with NCC

Figure 3: Heatmap of SSD between features of im1 and im2.

In this figure, we can see some colorful points. On the right side of each figure we can see some values with color matching. Based on that fact, if a point of matrix has darker(close to black) color we can say it is lower values compared with others. The objective of this figure is one, we just want to see some relationship between features from two images. But what can I talk about this figure is this result so confused that we can't make right decision on that figure immediately. Say it again, it is **NOT** meaning we **CAN'T** make a right decision. But there is a good way to express those result. Only we consider about that figure is just a pair which has **minimum value**. So we just change it another form with some numbers:

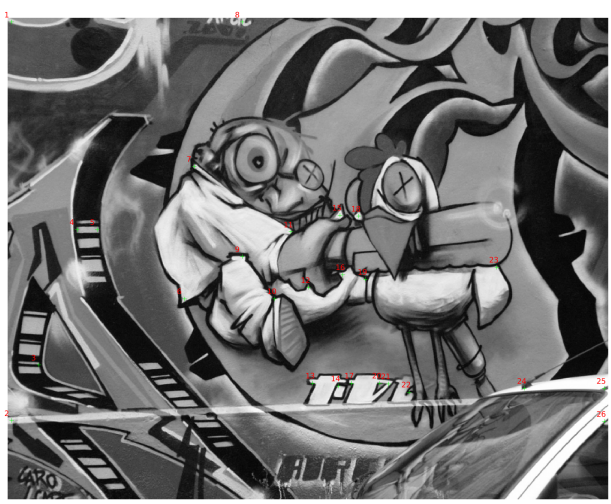


Figure 4: Harris operator result of im1 with numbers.

Figure 5: Feature matching result with above result. You can see some numbers near some points. That means this points is related with points which has same number in the above figure.

Now we can see, which points are correlated with each other. Obviously, there are some points which has no correlation with any others. But we can see that a certain point that is probably matched with other points but the result is not with our expectation. The reason why this situation can be from is I think there is no idea about feature matching, just comparing SSD or NCC between two points. Although it seems like they are similar but they can be matched with another points. I think that this kinds of problem can be resolved by applying some processing like computing 8-way gradients to compare exactly, using another descriptors which can be more accurate than our rotational-invariant descriptor and so on.

2 Image Stitching

2.1 Task 4

2.1.1 Homography transformation

In this task, I use parameters like below:

$$parameters = \begin{cases} error_metric : SSD \\ threshold : 1000000 \\ number_of_inlier : 21 \end{cases}$$

First, I check the distance matrix which was achieved from **task 3**. With some speculation, I choose the value of threshold as 10^6 using a heuristic. The reason why I use error metric like that is SSD is easy to compare than NCC

because there is nothing to do to compare two values. Just pick a lower value. So based on this parameters and my codes, the result of homography transformation is like this:

Figure 6: Result of homography transformation. We can see quite good alignment result. Becuase I can't make a method to crop figure neatly, so please understand this problem.

Using the advantages of RANSAC, we can make good image alignment. But also we can make a bad quality of homography transformation. I think this problem comes from my code which drop the underscored pairs not before the RANSAC but do the progressive of RANSAC. The bad case of homography transformation is below. But despite of this problem, the codes mostly can make quite good result so I think my code is enough to make fine result like this simple example.

So as a result, the number of iteration to get homography matrix with a inlier chance of 0.99 is $log(1-0.99)/log(1-(21/149)^4) = 1.1669 * 10^4$.

Figure 7: Result of homography transformation. We can see some disalignment on the top side of image

2.1.2 Affine transformation

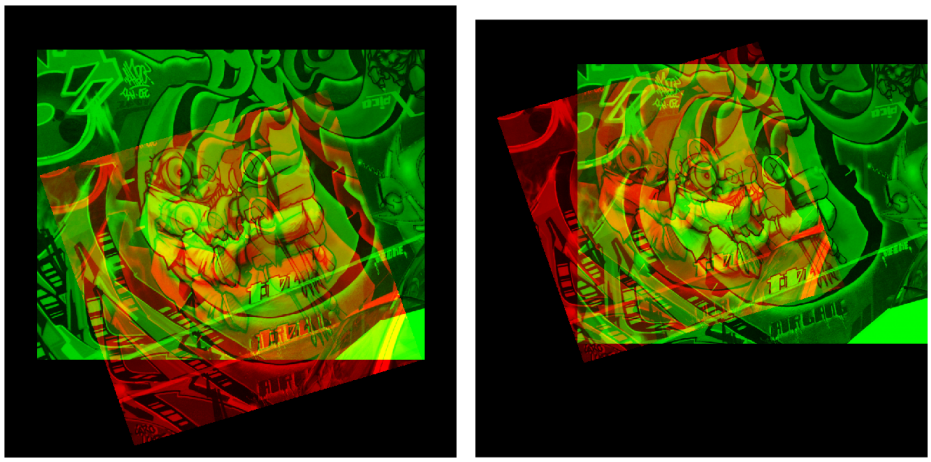
In this task, I use parameters like below:

$$parameters = \begin{cases} error_metric : SSD \\ threshold : 1000000 \\ number_of_inlier : 11 \end{cases}$$

Contrary to calculate homography matrix, my codes can't make good quality of affine matrix. The result is like this:

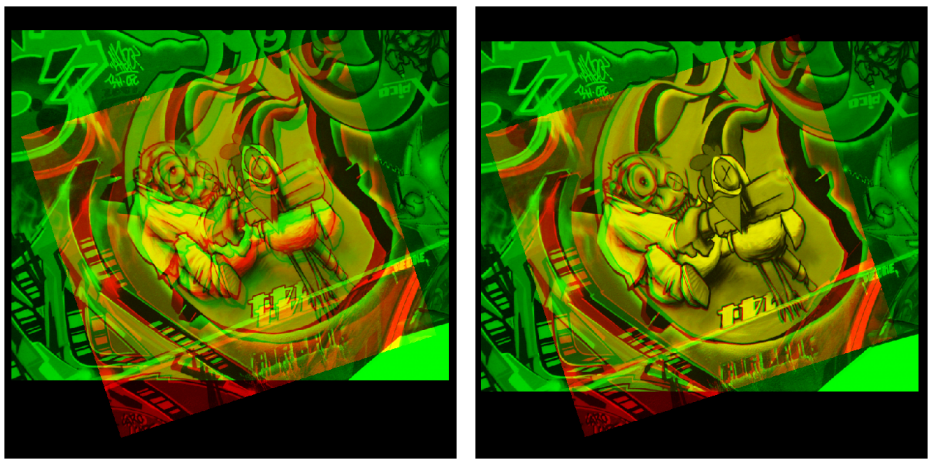
I think the reason of why this affine transforamtion can't successfully worked as a high probabilty is limitation of affine transformation. Logically, homography transformation contains affine transformation so that affine transformation can't make same result with respect with homography transformation. However, despite of this limitation, sometimes it can work like this:

Finally, the number of iteration to get homography matrix with a inlier chance of 0.99 is $log(1 - 0.99)/log(1 - (11/149)^4) = 1.5503 * 10^5$.



(a) Bad case of affine transformation 1 (b) Bad case of affine transformation 2

Figure 8: The bad result when im1 is transformed to im2 with calculated affine matrix.

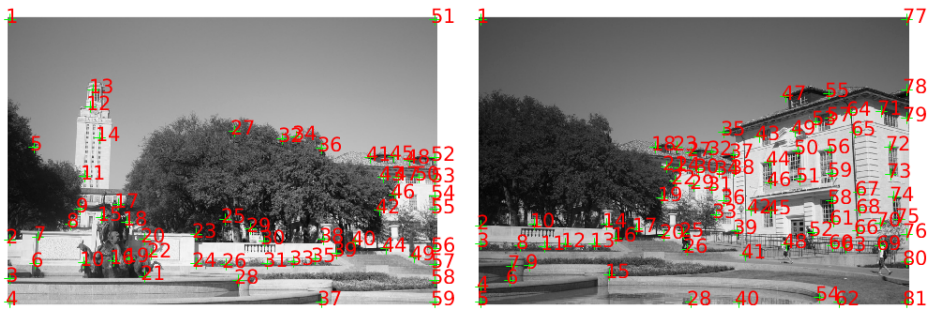


(a) Good case of affine transformation 1 (b) Good case of affine transformation 2

Figure 9: The good result when im1 is transformed to im2 with calculated affine matrix.

2.2 Task 5

I tried to my best, but I can't make image panorama. First I can't make feature matching using same method. Obviously, I tried to find the optimal value of parameters. But with my knowledge, I can't make feature matching so I tried to do some trick like this. First I show you my feature extraction result:



(a) Left side image (b) Right side image

Figure 10: Feature extraction result of each side image. It seems like feature extractions work well.

And this is a result of matlab built-in feature matching method. We can see nice result, and I see a tendency that the candidate features of matching are skewed. The meaning of this is matched features of left image are clustered to right side of picture and vice versa. So I think that if matching pairs can't work well with all the points then the matching will be successfully worked with points with close to the baseline.

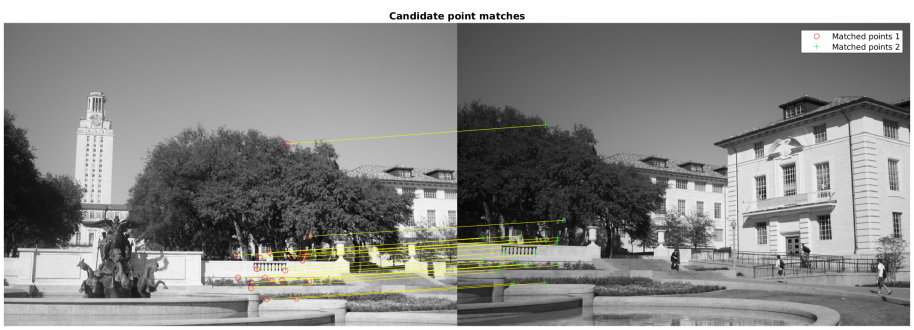
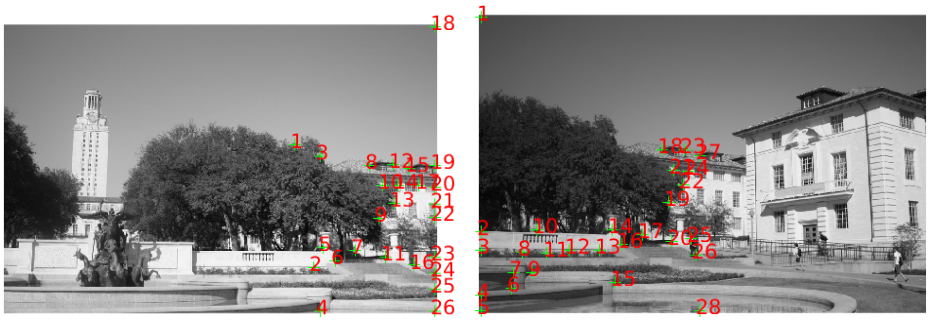


Figure 11: Feature matching result with using built-in function of matlab

So I detect some corners using harris operator and filtered them in respect with above strategy. The result is below. But the result is unlikely to my idea. It doesn't matter I used some trick or not. Sadly, feature matching doesn't still work. So if I have a time to do this problem later, maybe I want to figure out why this trick can't work.



(a) Left side image (b) Right side image

Figure 12: Feature extraction result with trick. It seems like feature extractions work well.

3 Appendix

1. Here is my github link: https://github.com/samsara-ku. I will open my private repo as soon as possible.