

Homework3

0856144邱賢祐
0856095黃柏豪
0856148陳奕遠

1. Introduction

In this homework, we practice how to implement automatic Panoramic image stitching process. This approach contains four steps including interesting points detection and feature description, feature matching, compute homography matrix with RANSAC, and wrap two images with the homography matrix.

For step 1, we use Scale-Invariant Feature Transform (SIFT) as interest points features. SIFT feature is invariant to image rotation, scale and noise because it uses the edge orientation histogram to describe the key points.

In step 2, feature matching is done by finding the nearest feature with L2-norm distance and ratio distance. Ratio distance is the closest distance divided by the second closest distance. Thus, if the ratio distance is close to 1, it means there are 2 ambiguous matching at least, and we should avoid this kind of matching.

In step 3, RANSAC implemented here is to find the best homography matrix. The key idea is to update the homography matrix in each iteration if the number of inliers is higher than the previous one that is derived via the previous homography matrix.

In step 4, we first implemented image stitching, and we utilize linear blending and pyramid blending to optimize the blending result.

2. Implementation Procedure

2.1. Interest points detection & feature description by STFT

SIFT takes a 16x16 square window around the detected feature and divides it into 4x4 grid cells. Each cell has a 8 bins orientation histogram. So, the SIFT feature has 128 dimensions.

Usually, the steps of construction of SIFT have 4 steps. First, use Difference-of-Gaussian (DoG) to find local extrema over scale and space and these local extrema are the potential key points locations. Second, refine the local extrema and compute where the key points actually are. Third, assign orientation to each key point and an orientation histogram with 36 bins covering 360 degrees is created. Finally, take a 16x16 square window around the key points and divide it into 4x4 grid

cells. Each cell has a 8 bins orientation histogram. So, the SIFT feature has 128 dimensions.

But in our implementation, we directly use the OpenCV function to construct the SIFT descriptor like the code down below.

```
## Find keypoints and compute descriptors in 2 images
sift_detector = cv2.xfeatures2d.SIFT_create()
(kp1, desc1) = sift_detector.detectAndCompute(img1,None) # kp is cv2.KeyPoint object
(kp2, desc2) = sift_detector.detectAndCompute(img2,None) # desc is numpy ndarray
```

2.2. Feature matching by SIFT features

We use the L2-norm distance metric as the similarity between two descriptors. Once all the descriptors are constructed, we compute the distance for all combinations of key points between two images. And try to find out which key points in image 2 are the closest and second closest to each key point in image 1.

In the end, compute the ratio between closest distance and second closed distance. If the ratio is larger than the threshold, we ignore it. We set the ratio as 0.5.

2.3. RANSAC to find homography matrix H

Due to mismatching and impreciseness of keypoint detection may happen through SIFT, RANSAC implemented here is to find the best homography matrix. Before discussing the procedure of finding the homography matrix, we would like to mention the idea of RANSAC beforehand. The RANSAC formula is shown below:

$$P_{\text{model}} = 1 - (1 - P_{\text{inlier}})^N$$

Where P_{model} is the probability of getting a model that a set has all inliers. P_{inlier} is the probability of picking an inlier. N is the number of elements randomly selected to a set. **Iteration** is the repeating times of the selection.

The goal of RANSAC is to have a chance of getting all inliers in a set as high as possible. Practically, we do not know the value of P_{inlier} , however, we can control N and iteration to reach $P_{\text{model}} > 99\%$. The formula calculating iteration times is shown below:

$$\text{Iteration} = \frac{\log(1 - P_{\text{model}})}{\log(1 - P_{\text{inlier}}^N)}$$

For example, suppose we want to have a 99% chance of getting the best homography. Assume probability of getting an inlier is fixed to 0.5. If N is set 30, the iteration should be $\log(1-0.99)/\log(1-0.5^{30}) = 4,944,766,884$. In addition, if probability of getting an inlier is changed to 0.9, $\log(1-0.99)/\log(1-0.9^{30}) \approx 106.32$. To sum up, times of Iteration to reach the

highest P_{model} grows while N gets higher or P_{inlier} gets lower. In addition, iteration times are uncertain which could be extremely big.

Returning to the subject of finding the best homography matrix. The procedure is listed below:

1. derive homography matrix via numbers of matches selected randomly.
2. calculate inliers.
3. update matrix if number of inliers is more than the previous one.

We randomly select matches considered as good matches via the last procedure to a set for deriving homography matrix. The homography matrix giving the highest number of the inliers is considered the best transformation matrix. To evaluate the number of inliers, a threshold of the distance between two corresponding key points is needed. In our implementation, it would be determined as an inlier if the deviation (distance) of the $img1_kp$ and the projection point (projection point = $H \times img2_kp$. H is a homography matrix, “ \times ” is matrix multiplication) is within 5 pixels. Notably, in a keypoint match of two images, there is one key point in each. One is the keypoint in $image1$, $img1_kp$, another is the keypoint in $image2$, $img2_kp$.

2.4. Warp image to create panoramic image

In this section, we first wrap the $image2$ with the homography matrix we get from section 2.3. Second, we synthesize $image1$ and $image2$ by utilizing pyramid blending and linear blending.

In pyramid blending method, the procedure is listed below:

1. Generate Gaussian pyramid for $image1$ and $image2$
2. Generate Laplacian pyramid with the Gaussian pyramids
3. Combine the left part ($image1$) and right part ($image2$) in each level
4. Reconstruct the result image

In linear blending method, we use a linear mask to blend the boundary between $image1$ and $image2$.

3. Experimental Result

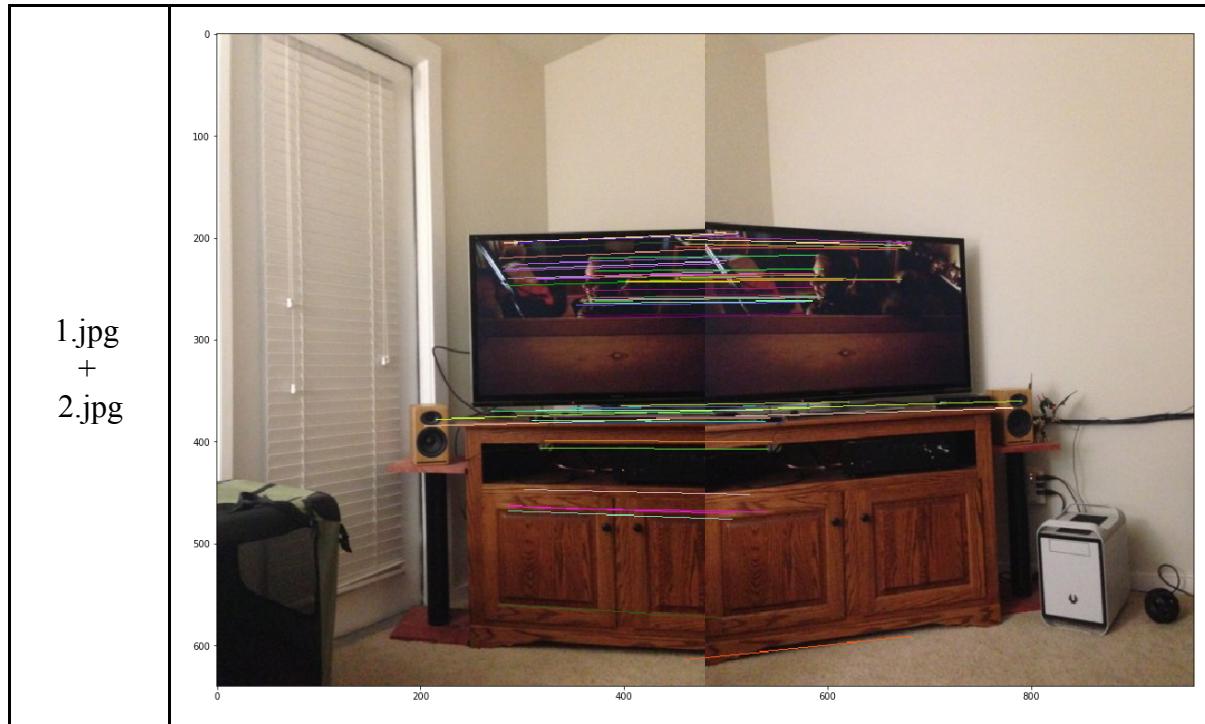
3.1. Interest points detection & feature description by STFT

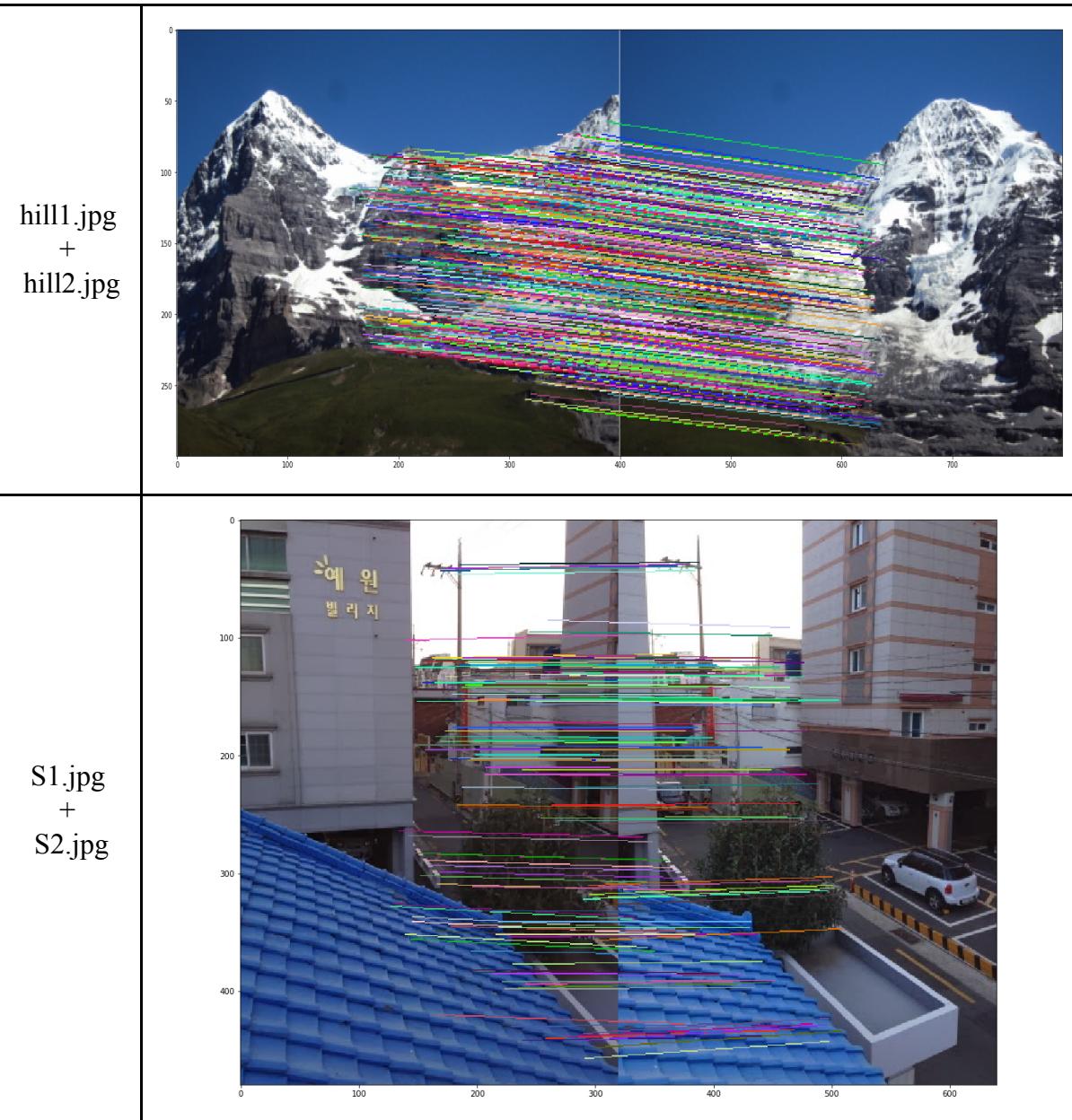
The following table shows the number of keypoints in each image.

image	1.jpg	2.jpg	hill1.jpg	hill2.jpg	S1.jpg	S2.jpg
#keypoints	410	424	1217	1494	972	1047

3.2. Feature matching by SIFT features

The following table shows the matching result of each pair of images.

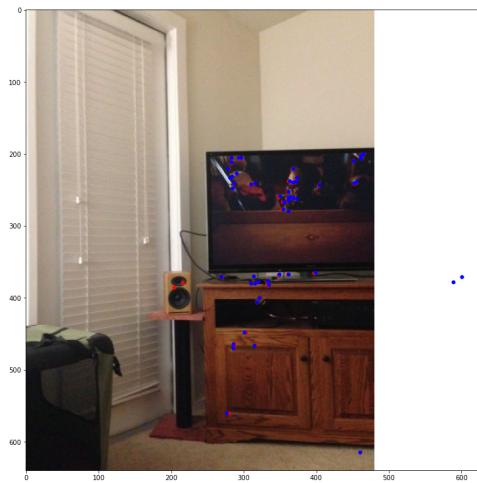




3.3. RANSAC to find homography matrix H

To compromise the issue of RANSAC aforementioned, we set $N = 4$. Under the condition that we do not know the probability of getting best key points and we believe the precision of key point detection and match, we set iteration = 10,000. The result of the best homography matrix is shown below. Red dots are key points of image1, blue dots are key points of image2. The picture 1+2.jpg shown below has such a result due to mismatching that leads to two blue key points projected to outside of image1.

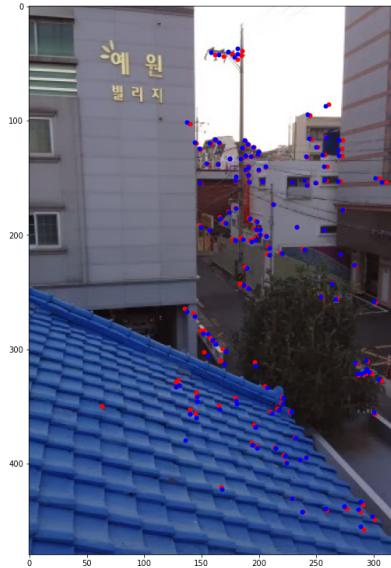
1+2.jpg



hill1+2.jpg

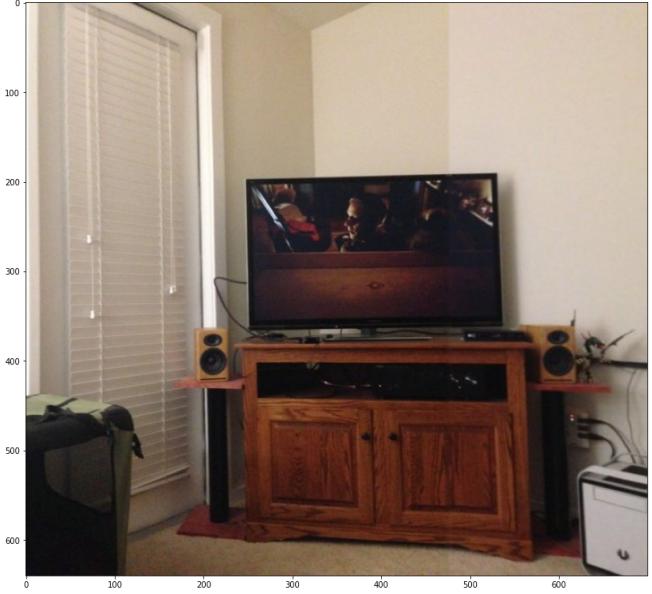


S1+2.jpg

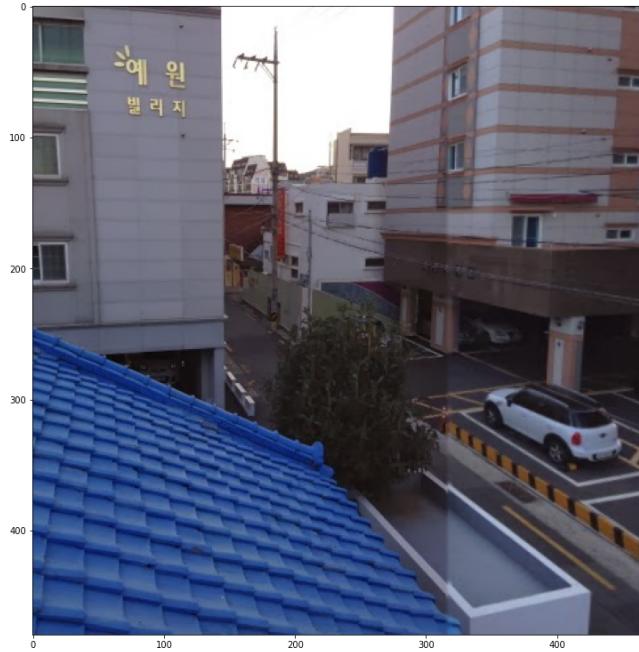


3.4. Warp image to create panoramic image

a. Results of Pyramid blending

1+2.jpg	
hill1+2.jpg	

S1+2.jpg



b. Linear blending

1+2.jpg



hill1+2.jpg

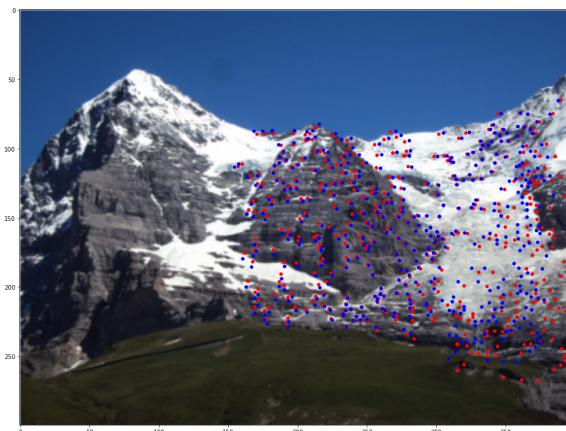


S1+2.jpg



4. Discussion

We did an experiment of getting a homotrophy matrix with all matches via feature matching by SIFT descriptor. The result is shown below. It is easy to observe that the result is worse than RANSAC with set size=4. We assume that the worse result is due to the noise of mismatch and key points precision. Example of mismatch is shown 1+2.jpg in section 3.3.



5. Conclusion

In this assignment, we implement image stitching including Interest points detection, feature description by STFT, feature matching, homography matrix selecting via RANSAC algorithm, pyramid blending. In key points matching, we use feature matching with SIFT feature descriptor. In the homography matrix, we

implement RANSAC for selecting the best one. In addition, we illustrate the purpose of RANSAC and the parameters of it. In image stitching, we make use of the homography matrix for warping. Then, we implement two blending methods, pyramid blending and linear blending for blurring the line between two images in stitching.

6. Group work assignment

0856144邱賢祐: step 4. Warp image to create panoramic image

0856148陳奕遠: step 3. RANSAC to find homography matrix H

0856095黃柏豪: step 1&2. KP detection & matching