國立陽明交通大學

113-1 電腦視覺 HW-3

Group 9 313551130 鍾名捷/313551133 陳軒宇/313553036 葉軒宇

1. Introduction

In this homework, our goal is to implement an automatic panoramic image stitching system. This system will takes two image as input, and consider the part which are overlap by two image to produce a seamlessly stitched panoramic image as the output. There are several important details in the process of implementing, including feature detection, homography estimation using RANSAC and image blending.

2. Implementation procedure

Step1: Feature detection and matching using SIFT

First we using OpenCV's SIFT detection to find the keypoints and descriptors. Then for each feature in source1 compare with all the features in source2 to calculate the distance using Sum of Squared Differences (SSD):

$$SSD = \sum (des1[i] - des2[i])^2$$

For each feature in image1, we find its two closest matches in image2 and apply the ratio test

$$\frac{\|f_1 - f_2\|}{\|f_1 - f'_2\|} < ratio$$

Where f_2 is the closest match and f_2' is the second closest match. This helps to ensure the distinctiveness of the matches.

After that, create combine image showing both input image and create color line to connect matching points.

Step2: RANSAC for homography estimation

When we get a lot of matching points, we can calculate the corresponding homography **H**. Assume the points in image1 be **P,** points in image2 be **P'**, we can get relationship like this.

$$wP' = HP, \quad where\ w\ is\ H's\ parameter$$

Cause we already have lots of character pairs, so we can calculate homography by these poings. In order to calculate it, separate upon function to

$$\begin{bmatrix} wx'_i \\ wy'_i \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

It equals to

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i y_i & -y'_i \end{bmatrix} = \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

It can simplify into

$$A = h0$$

While theoretically a minimum of 4 points correspondences is sufficient to compute the homography matrix **H**, this minimal solution can be sensitive to noise and potential matching errors. Therefore, in our implementation, we utilize 32 sample points in the RANSAC framework to enhance the robustness and accuracy of the estimation. This over-determined system helps minimize the impact of outliers and provides a more stable solution for the homography transformation between the two images.

Here are the RANSAC steps:

1. Samples a subset of 32 point pairs from our matched features
2. Calculate the Homography matrix H
3. Compute the projection error for all point pairs
4. Count the number of inliers within the error threshold
5. Repeat steps 1-4 for N iterations, keeping the H matrix with the most inliers

Step3: Image warping and blending

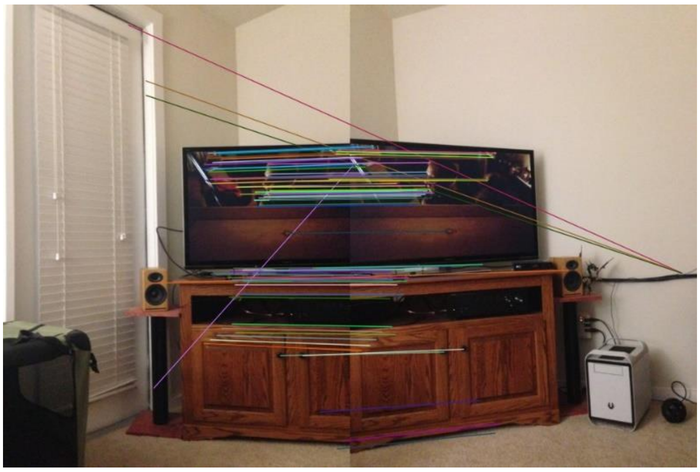
After obtaining the homography matrix **H**, we need to wrap one image onto another and blend them together. First, calculating the boundaries of warped image, then create output canvas with appropriate size to accommodate both image. Last, implement linear blending in overlapping regions using the formula.

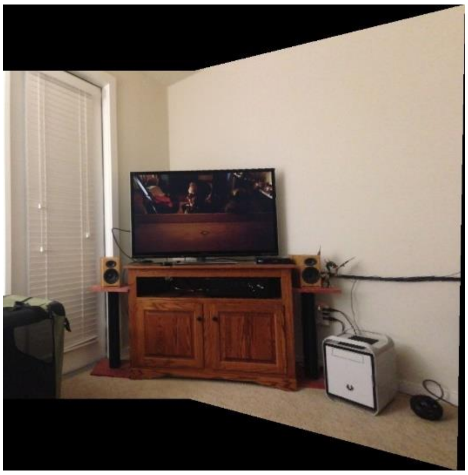
$$weight = \frac{(x - overlap_{start})}{overlap_{width}}$$

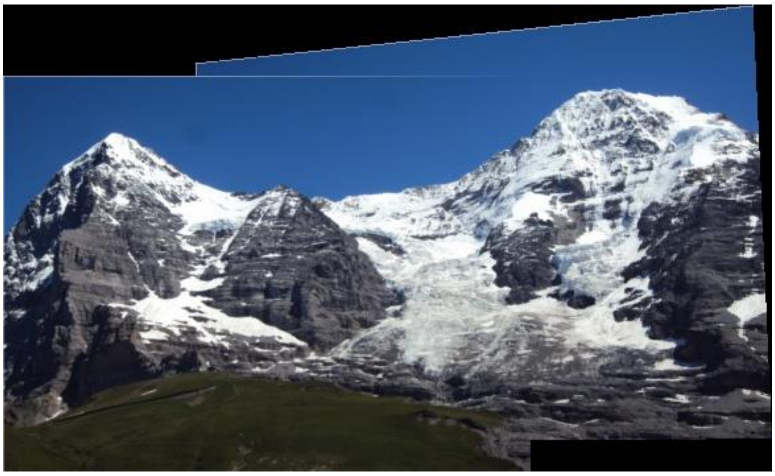
$$result = (1 - weight) * image1 + weight * image2$$

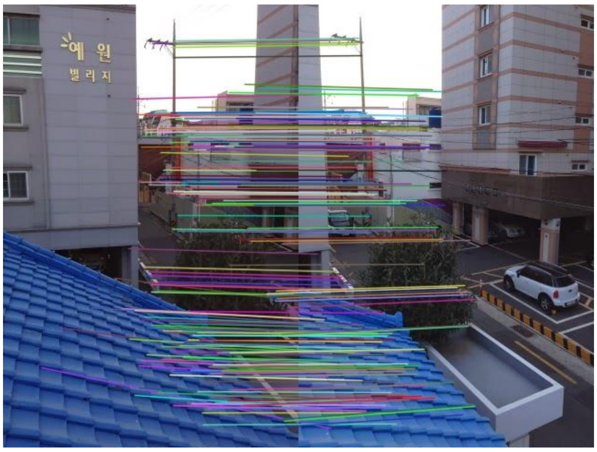
3. Experimental results

1. Results of the picture provide by teacher

Original image	
Match points	

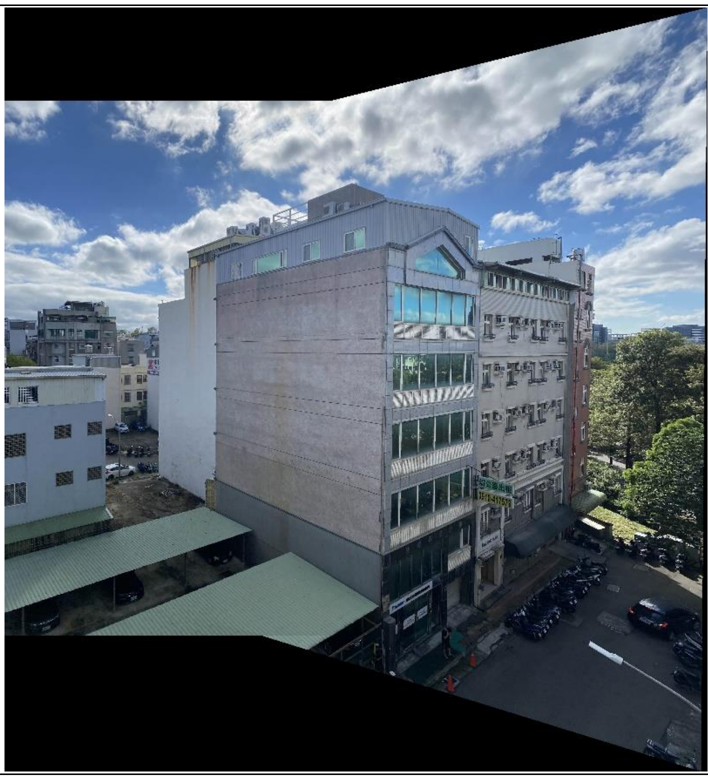
result	

Original image	
Match points	
result	

Original image	
Match points	
result	

2. Results of the picture take by us

Original image	
Match points	
Result	

Original image	 
Match points	
Result	

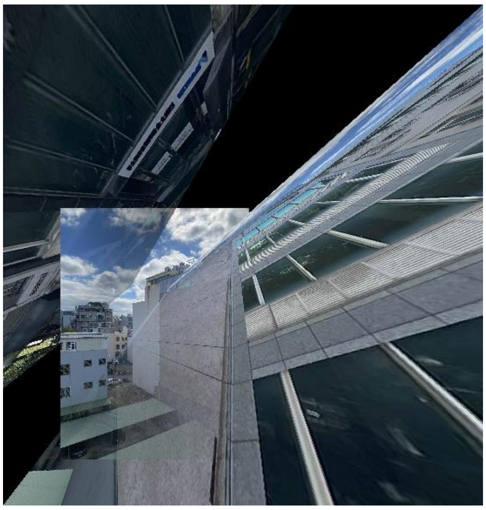
4. Discussion

There are two main challenges during this homework. First, the picture size of the picture taken by us are much much more larger than the picture which are provided by the teacher. This cause the number of keypoints when we are matching are really huge

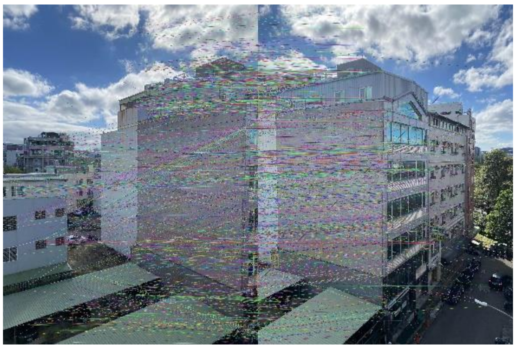
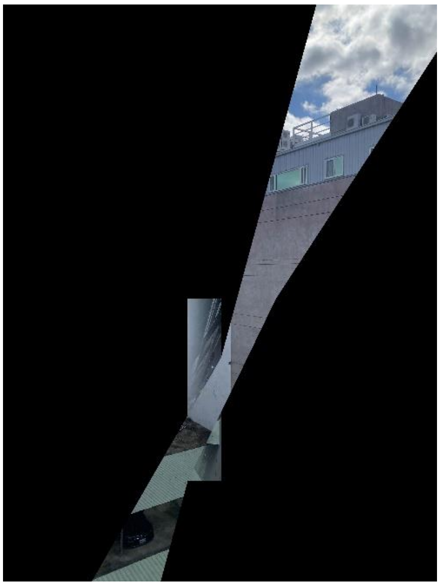
Picture by us	read images, (result1.JPG, result2.JPG) # of keypoints, (114296, 81925)
Picture by TA	read images, (TV1.JPG, TV2.JPG) # of keypoints, (422, 429)

We've try to reshape the picture in order to reduce time, but the result are not that good as we expect, so we keep the original size and wait be like 2~3 hours for the results.

Another challenge is when we try to stitching the picture provided by TA, all of the feature matching threshold are set by 0.5. But when we want to apply the same parameters to our picture, the results becomes really bad

We think that maybe the threshold are too low, so we increase it into 0.8

But the result become worse, at the end we try threshold = 0.6, and get really well result. By our observations, threshold=0.5 are too low that the matching point are not enough, and threshold=0.8 are too high so get too many wrong matching points.

5. Conclusion

The selection of feature matching threshold demonstrates that successful image stitching requires careful parameter tuning. Besides, Larger images produce more keypoints, significantly increasing computation time While resizing images can improve processing speed, it may compromise stitching quality Trade-off between processing time and result quality needs to be carefully considered.

6. Work assignment plan

This project was completed through collaborative efforts among all team members. First, we discussed about the detail of implementing SIFT and snatching problem. After clearly understanding what we should de, we started to take some picture and modify the parameter to complete the whole task. Finally, we make a conclusion about we've encountered within this project.