Computer Vision HW3 Report

Group5

1. Introduction

Panoramic Image Stitching is a process of combining multiple photo graphic images with overlapping fields of view to segmented panorama or high-resolution image. It's application has been widely used in our daily life, like the panorama mode of smartphones.

In order to implement image stitching, we simply divided our process into four steps. First, we obtained Interest points detection and feature description by SIFT, SURF or ORB. Second, we match the description features across two images. Third, we applied the RANSAC algorithm to compute the homography matrix. Last but not least, we warped the image with the homography matrix and stitched the images to acquire our final result.

2. Implementation procedure

2.1. Interest points detection & feature description

We convert the images into grayscale and use the opencv function, such as SIFT, SURF or ORB to find the Key points and descriptions of two images

2.2. Feature matching by features

Furthermore, we use Sum of Squared Differences (SSD)

$$SSD = \sum_{i=0}^{127} (des1[i] - des2[i])^2 \qquad \frac{||f_1 - f_2||}{||f_1 - f_2'||} < \text{ratio}$$

to compute the most likely matching keypoints with the ratio to threshold out the bad matches. Lower ratio distance implies better matches. Therefore, we store the better matches and remove the worse matches.

2.3. RANSAC to find homography matrix H

(1) RANSAC algorithm find good pairs

First , we randomly choose 3 key points of img1 and img2 respectively .
Second , we use the pair of key points to calculate the affine matrix .

$$\begin{bmatrix} x1' & x2' & x3' \\ y1' & y2' & y3' \end{bmatrix} = M_{2\times3} \cdot \begin{bmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$M_{2\times3} = \begin{bmatrix} x1' & x2' & x3' \\ y1' & y2' & y3' \end{bmatrix} \cdot \begin{bmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

After we get the affine matrix , we could use this affine matrix to transform all img2 key points to img1 coordinate system .

Third , we calculate the error between img1 key points and transformed img2 key points(MSE) . If error is smaller than certain threshold , we put the corresponding pair of img1 and img2 key points in the array and we called the array “Inner Group”.

By doing these steps lots of time , we save the best “Inner Group” which has the most pairs of key points to find homography matrix H .

(2) Find homography matrix H

In this step , we use hw1 “camera calibration” function to find the homography matrix H . We first choose array P

$$P = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_2x_1 & -x_2y_1 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_2x_1 & -y_2y_1 & -y_2 \end{bmatrix}$$

where x_1, y_1 are the image2 points , x_2, y_2 are the image1 points

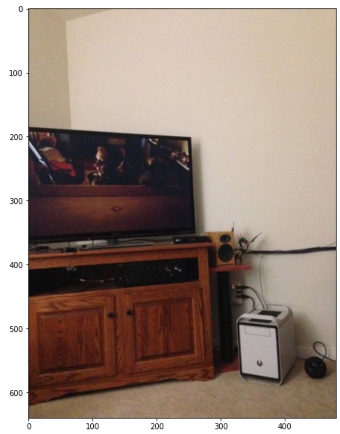
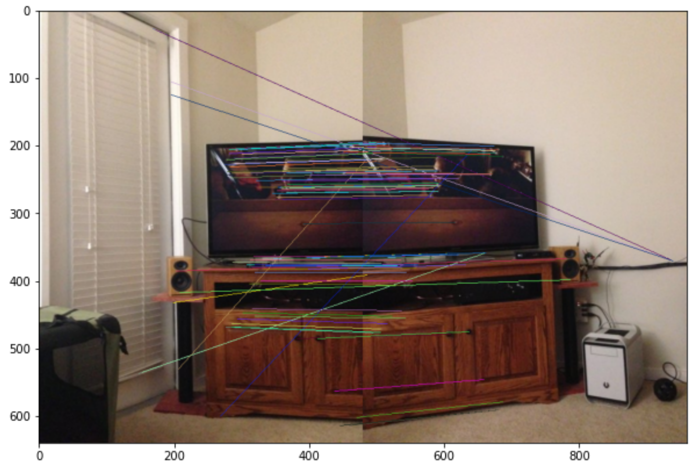
The answer of the homography matrix is the eigenvector of P^TP associated with the smallest eigenvalue and reshape to 3*3 .

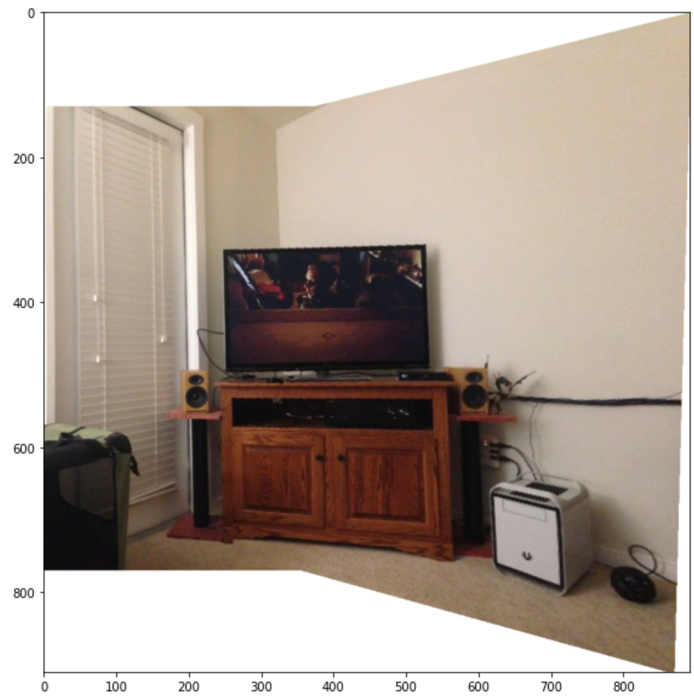
2.4. Warp image to create panoramic image

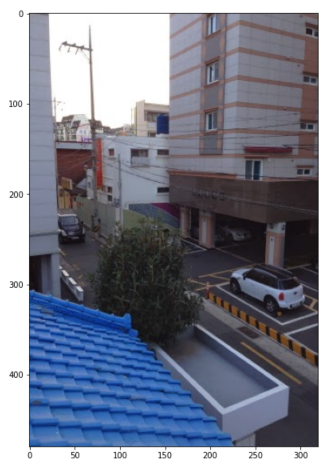
After we find homography matrix H , we could use homography matrix H to convert the coordinate axis between the first image and second image . First we find the four corner vertices of image2 and convert them to the coordinate system of image1 through the homography matrix . Second , we found the minimum and maximum of eight corner vertices (image1 and image2) , and we called “offset_min” and “offset_max” respectively . We use “offset” to understand the shape of the synthesized photo . Finally , we have the shape of the synthesized photo , “offset” and homography matrix , so we can use (1) “offset” to put image1 in the synthesized photo . (2) inverse homography matrix to convert each pixel in the synthesized photo to image2 coordinate system to check the coordinate whether it is in image2 and put image2 in the synthesized photo .

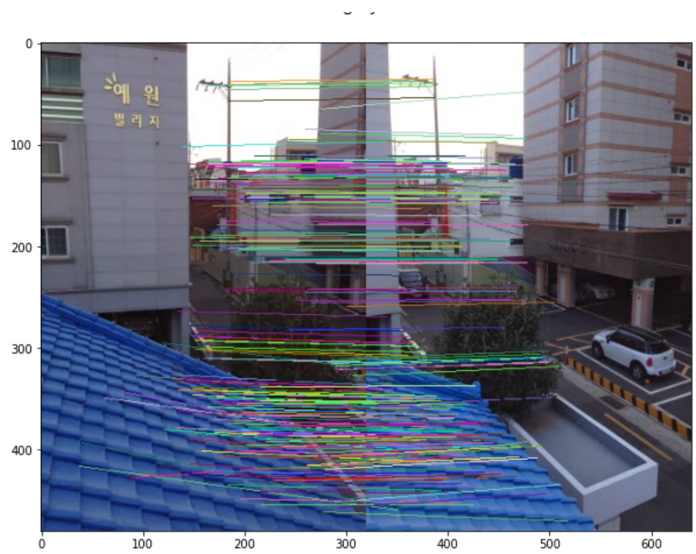
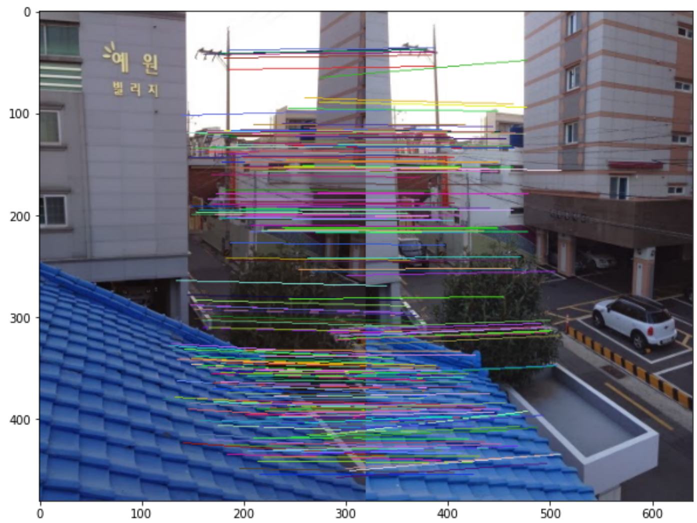
3. Experiment Result

SIFT		

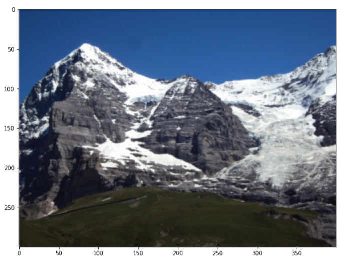
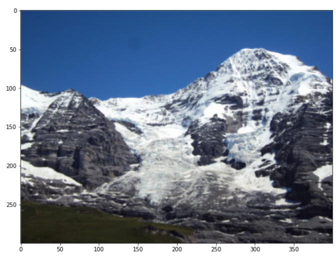
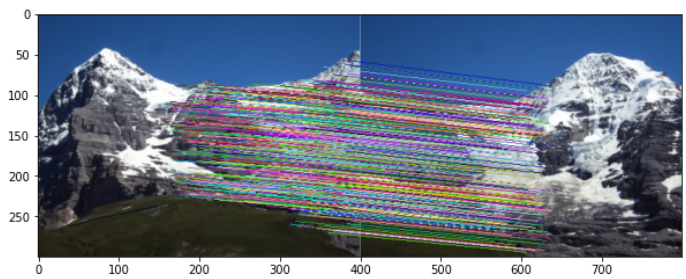
Original image		
Match points without RANSAC		
Matching points with RANSAC		

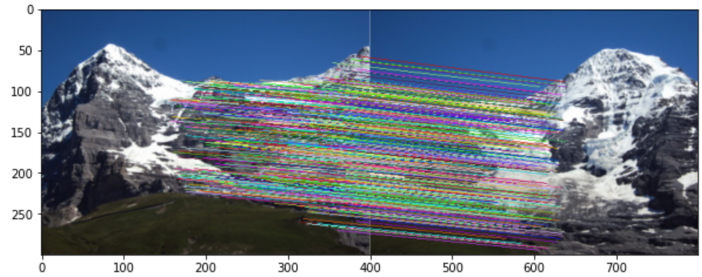
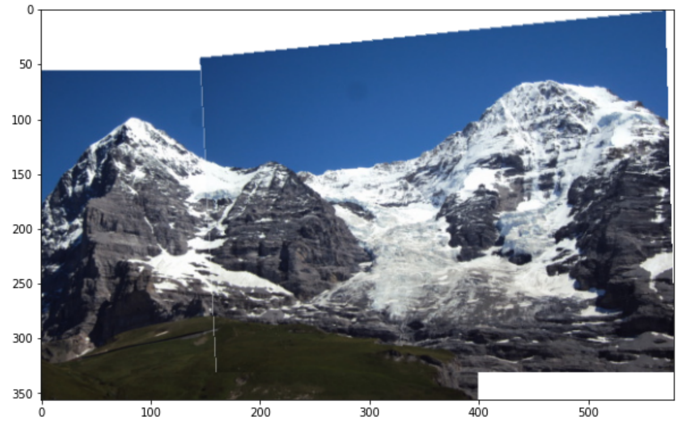
<table>
<tr><td>Warp result</td><td>
</td></tr>
</table>

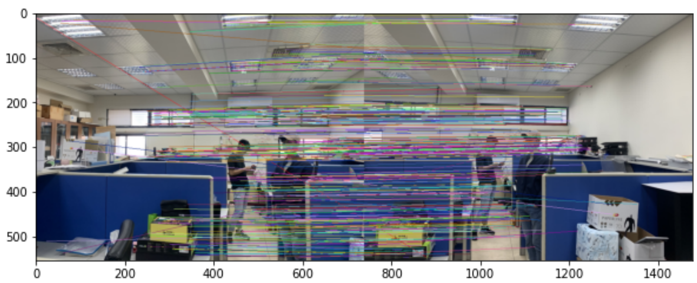
<table>
<tr><td>SIFT</td><td></td><td></td></tr>
<tr><td>Original image</td><td>
</td><td>
</td></tr>
</table>

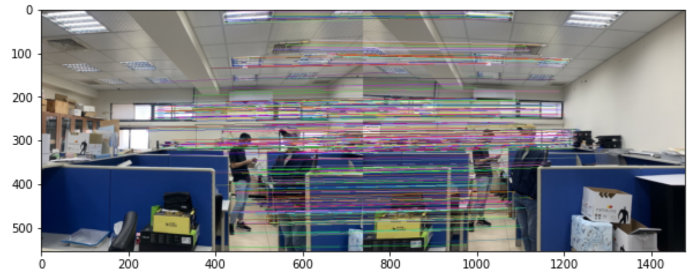
Match points without RANSAC	
Matching points with RANSAC	

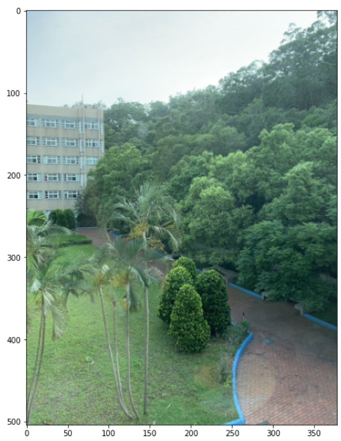
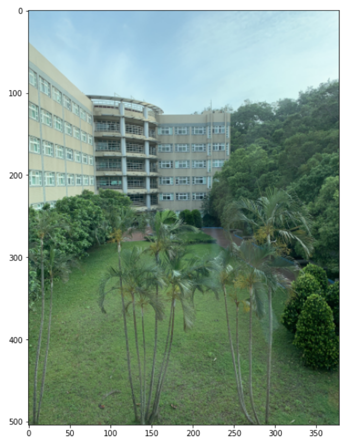
Warp result	

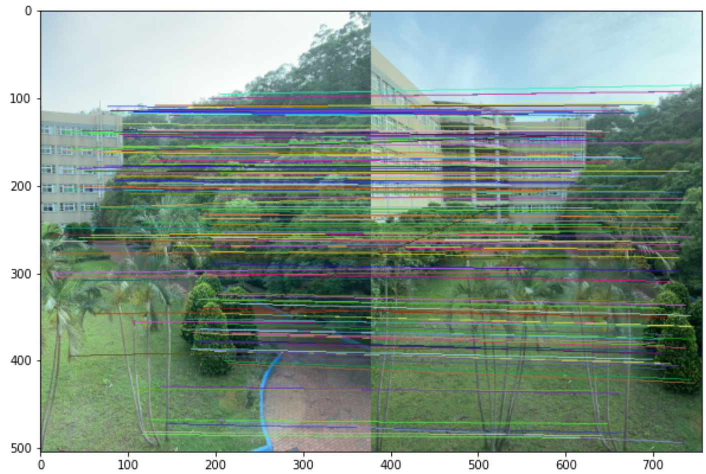
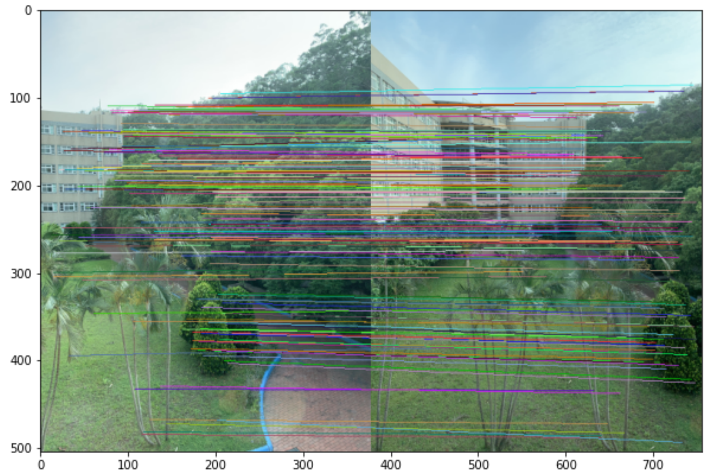
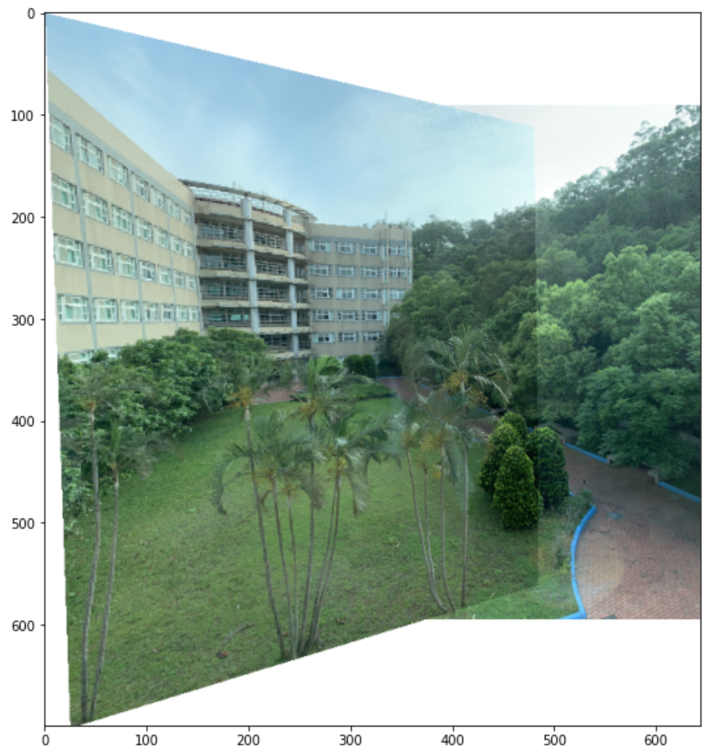
SIFT		
Original image		
Match points without RANSAC		

Matching points with RANSAC	
Warp result	

SIFT		
Original image		
Match points without RANSAC		

Matching points with RANSAC	
Warp result	

SIFT		
Original image		

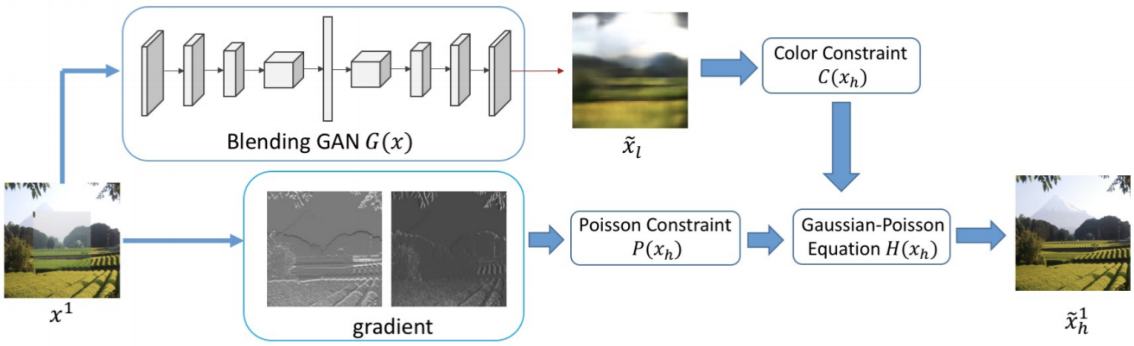
Match points without RANSAC	
Matching points with RANSAC	
Warp result	

4. Discussion

4.1. We try to use a blending function to enhance the result of the stitched image after we do the whole procedure mentioned above.

GP-GAN, the blending approach we choose, contains two parts, color constraint and Poisson constraint. To generate the color information, they propose a Blending GAN to learn the mapping between the composite images and the well-blended ones. Color constraint is applied to make a composite image more realistic and natural. In addition, they obtain gradient information including image details, like textures and edges, via applying gradient filters.

The model architecture shows below.



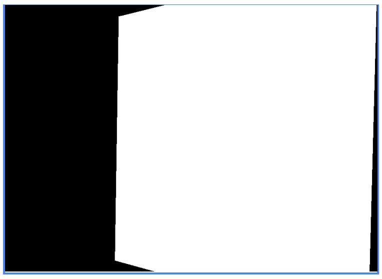
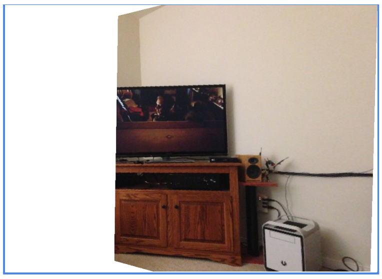
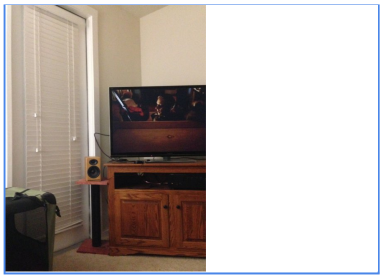
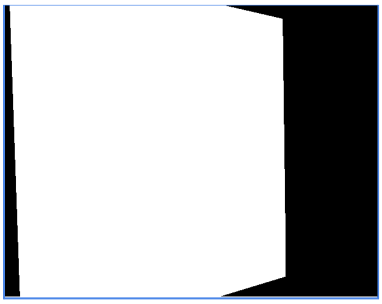
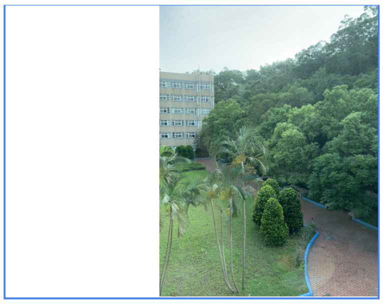
(site: https://arxiv.org/abs/1703.07195)

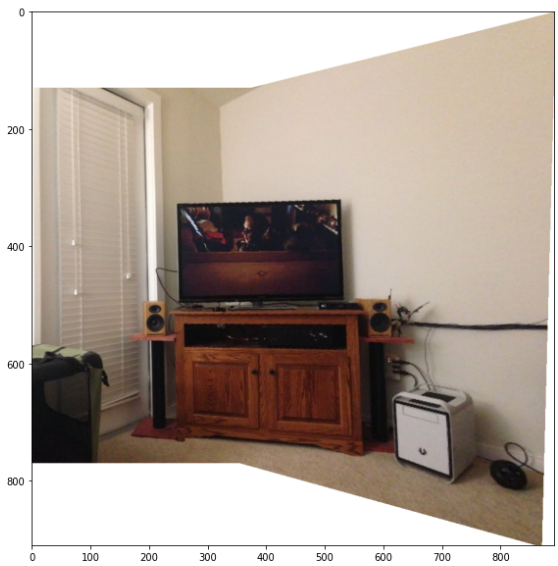
We apply the source code the authors provide on github. However the result becomes worse from our perspective. It generates some shadows on the image, and the color gets lighter. We think the reason is that the brightnesses and textures of two images we want to stitch are very close, so just stitching the two images, it looks fine . The other reason might turn out to be the training dataset and our images exist some domain gaps. Although, the color looks consistent.

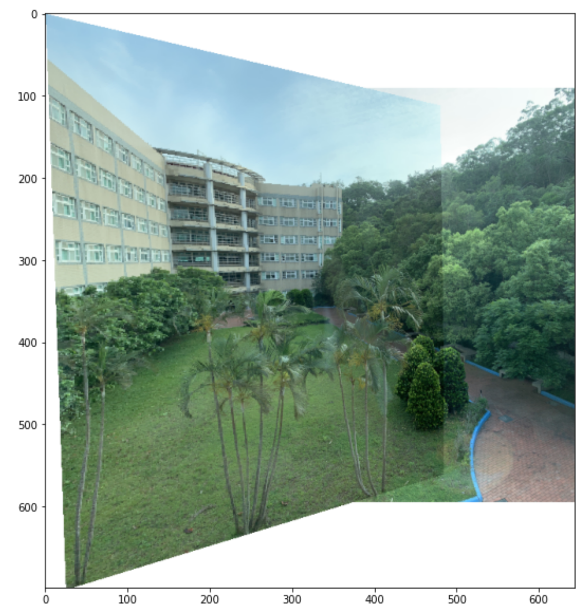
Following is the result generated by GP-GAN:

(github: https://github.com/wuhuikai/GP-GAN)

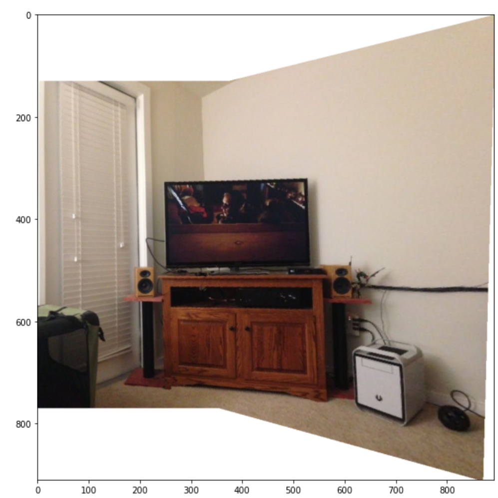
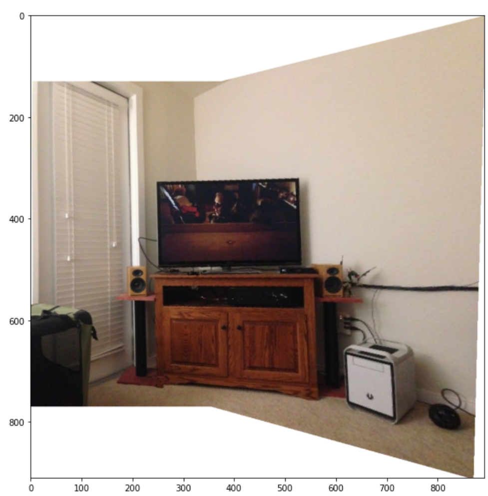
The source image is the pixel value from image 2 after calculating with the homography function from the procedure mentioned in section 3.

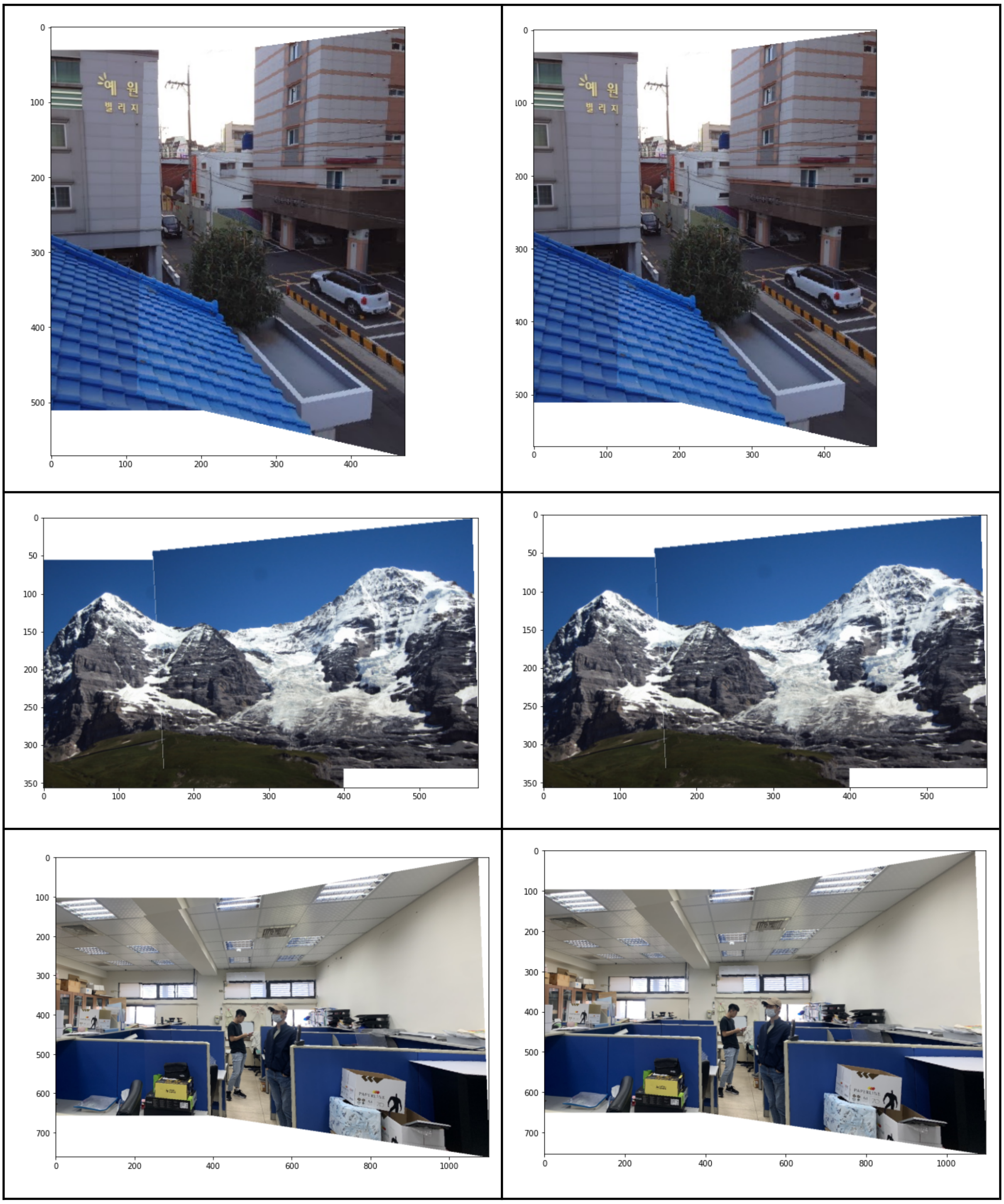
mask	src img	des img
		
		

without GP-GAN	with GP-GAN (worse)
	

4.2. We use other methods to accomplishInterest points detection & feature description. We don’t find too much difference when applying other methods.

surf	orb
	



5. Conclusion

In this homework, we have accomplished the image stitching work by detecting the interest points and finding the best matching to wrap 2 images together to get our result. In this process, we have researched different methods to find the interest points and descriptions, meanwhile, applying the DL-blending technique(GP-GAN) that we have recently studied. Although, The result doesn't come out well, it was still a good try and an interesting topic to discuss and improve the result. Overall, we enjoyed the lab and learned image stitching techniques.

6. **Work Assignment Plan with team members**
   we have finished the homework together !