Group 27

Computer Vision – HW3

A. Introduction

The goal of the assignment is to stitch panoramic image automatically. We learn how to use SIFT to get the match points, do RANSAC algorithm, and warp an image to create panoramic image.

B. Implementation Procedure

Step 1. Use SIFT to get the match points

1. Read two images and transform images to gray level.
2. Find the key points and descriptors of two images with SIFT in OpenCV.
3. Use Sum of Squared Differences (SSD) for any pair: (des1, des2)

$$SSD = \sum_{i=0}^{127} (des1[i] - des2[i])^2$$

4. Store all the good match point pairs after ratio test (ratio = 0.5).

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

5. Show all the match point pairs.

Step 2. Choose good point pairs by RANSAC

1. We randomly select 3 key points of img1 and img2 respectively.

2. Use the pair of key points to calculate the affine transform matrix.

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

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

$\mathrm{M}_{2\times3}$ on the equation above is the transform matrix we found.

3. Use the matrix to transform img2's points coordinates into img1's points coordinates.

4. Calculate the error between img1's points coordinates and transformed img2's points coordinates (Here we use mean mean square error)

5. If the error is smaller than the threshold we set, put the corresponding pair of img1 and img2 points in the array (Here we call the array the inner group)

6. Do 1.~5. each iteration, and find which iteration's number of the inner group is the most, and return those pairs of points.

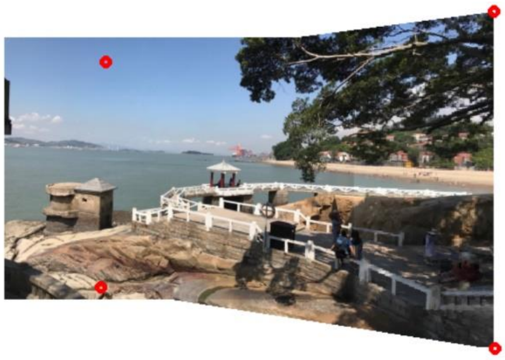
Step 3. Find Homography matrix H

1. In each pair of images ,we have image1 points $p_i = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}$ and image2 points $P_i = \begin{bmatrix} U_i \\ V_i \\ 1 \end{bmatrix}$
2. So according to the definition of H, we have $p_i = HP_i$ where H =$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}$
3. Rewrite the relation to an equation form, we will get $\begin{cases} u_i(h_3P_i) - h_1P_i = 0 \\ v_i(h_3P_i) - h_2P_i = 0 \end{cases}$ for 1 pair of points, and we have n pairs of points each plane. So we will have n equations and rewrite them to the matrix form.
4. $\begin{bmatrix} P_1^T & 0 & -u_1P_1^T \\ 0 & P_1^T & -v_1P_1^T \\ \vdots & \vdots & \vdots \\ P_n^T & 0 & -u_nP_n^T \\ 0 & P_n^T & -v_nP_n^T \end{bmatrix} \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} = \boldsymbol{X}h = 0 \xRightarrow[\text{may not be 0}]{\text{Because of noise,Xh}} \text{minimize} \|\boldsymbol{X}h\|^2$
5. subject to $\|h\|^2 = 1$
6. And we can solve this by SVD
7. $\boldsymbol{X}h_{2n*9} = \boldsymbol{U}_{2n*9}\boldsymbol{D}_{9*9}\boldsymbol{V}_{9*9}^T$, and we set h equal to the last column of V(the last row of V^T), and resize h(1*9) to H(3*3)

Step 4. Warp image to create panoramic image

1. To begin with, we would like to clarify some name in this step to make the following description clearer.
   - ✓ image1: the image be left unchanged.
   - ✓ image2: the one that we try to warp to match image1.
2. Use homography matrix we found in step 3 to find coordinate of four corners of image2 in image1's coordinate.

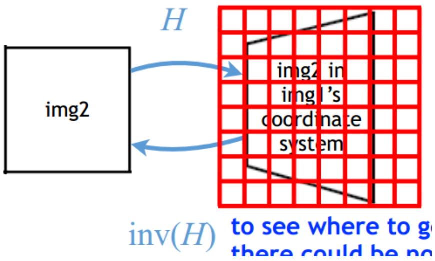
   We show four corners with red dot on the right picture.



3. All the pixel within the four corners in image1’s coordinate, we use inverse of homography matrix to find which pixel they match to in image2. If the value is not an integer, we simply find the nearest pixel. (the formula below is using homography coordinate)

$$\begin{bmatrix}x1\\y1\\z1\end{bmatrix} = H\begin{bmatrix}x2\\y2\\z2\end{bmatrix}$$

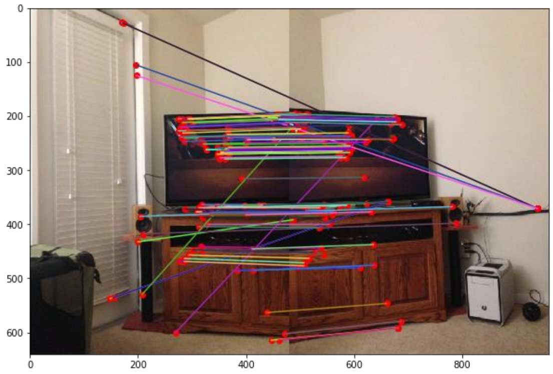
$$\begin{bmatrix}x2\\y2\\z2\end{bmatrix} = H^{-1}\begin{bmatrix}x1\\y1\\z1\end{bmatrix}$$

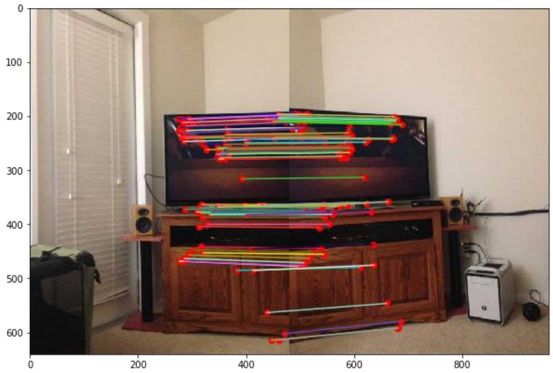
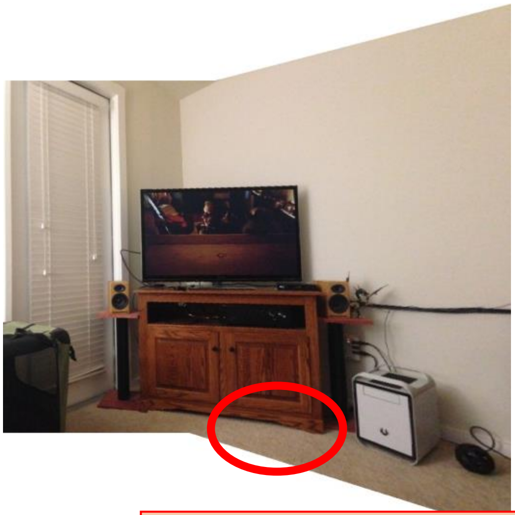
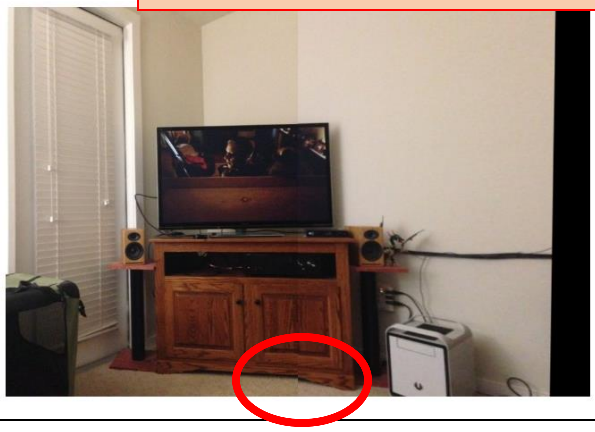


4. Then, we simply combine two images with image1’s coordinate.

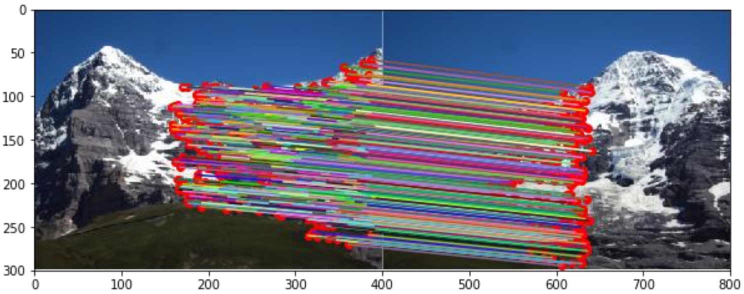
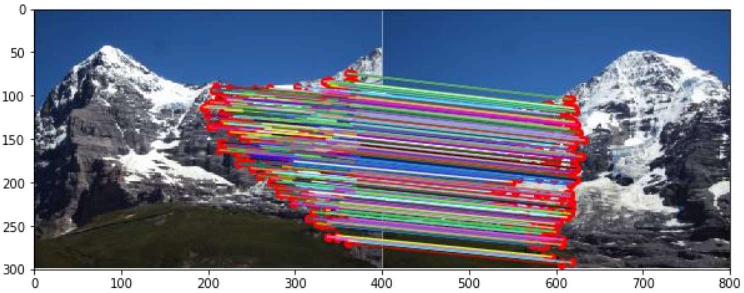
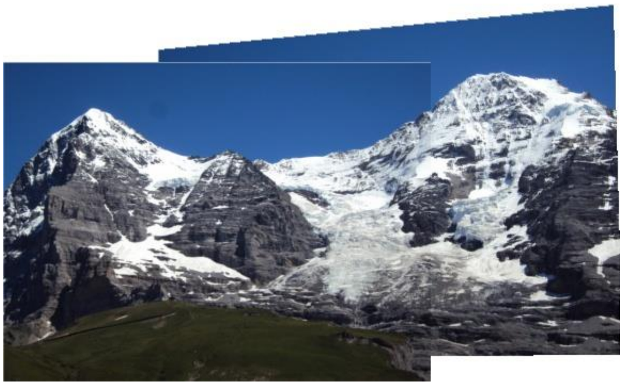
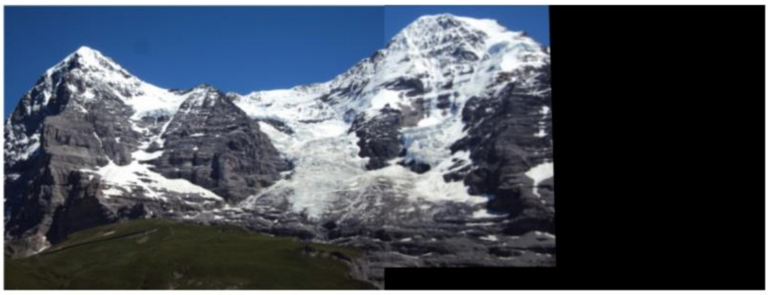
C. Experimental Result

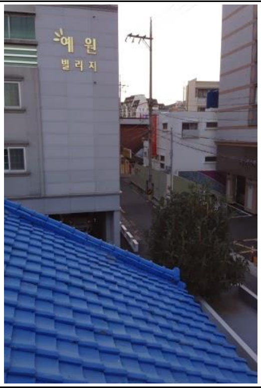
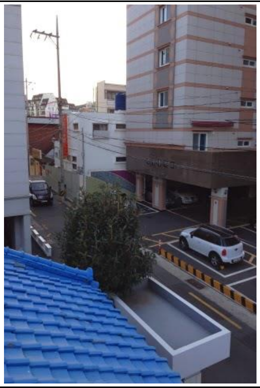
- The result of TA’s image

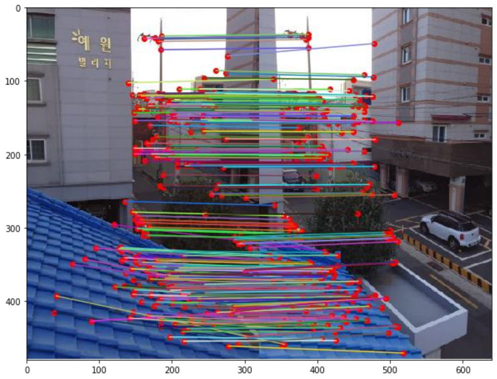
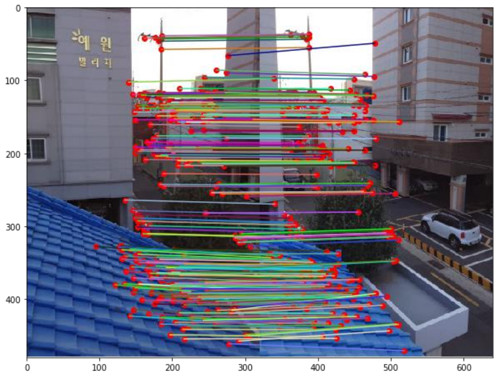
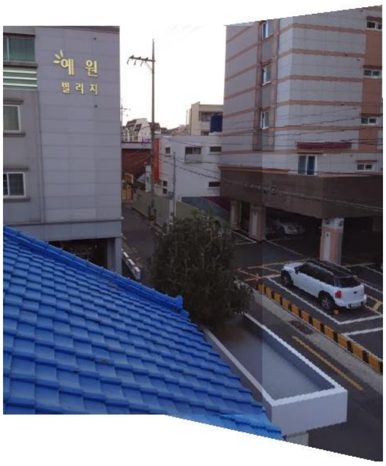
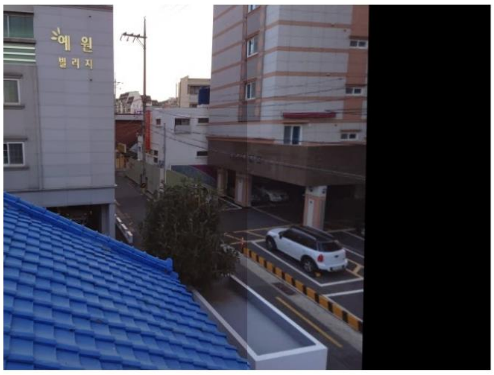
Original image		
Match points w/o RANSAC		

Match points w/ RANSAC threshold=20	
Our panoramic image	 We did a better job here
Result of build in function	

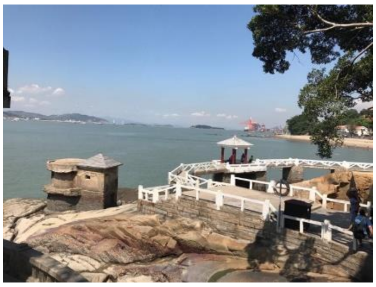
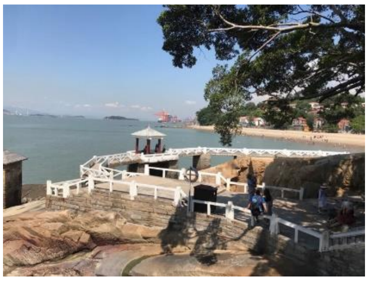
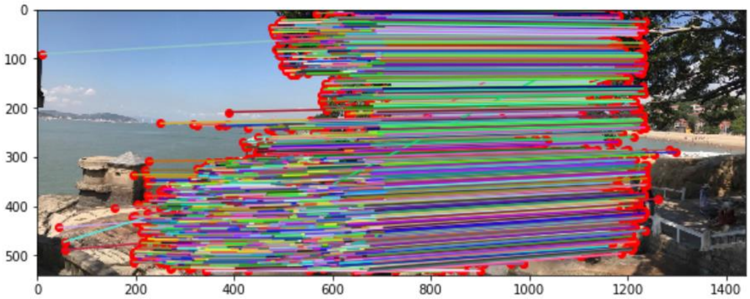
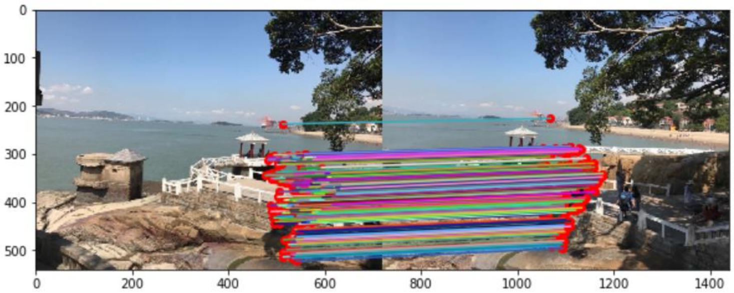
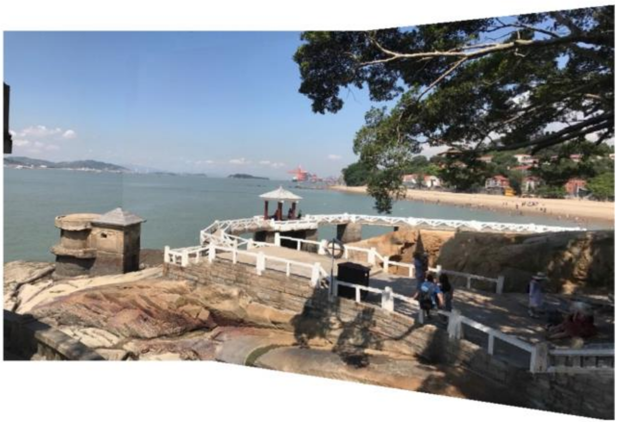
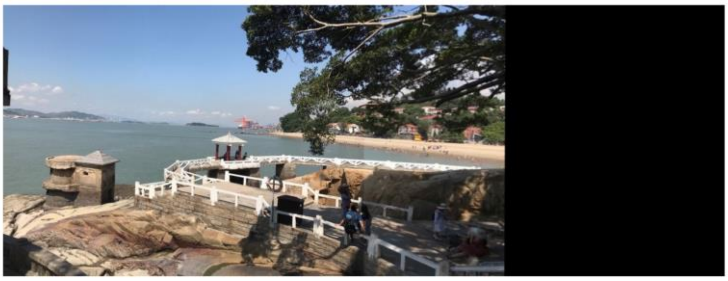
Original image		

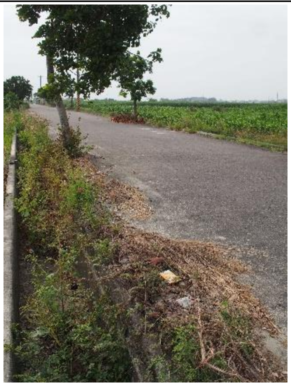
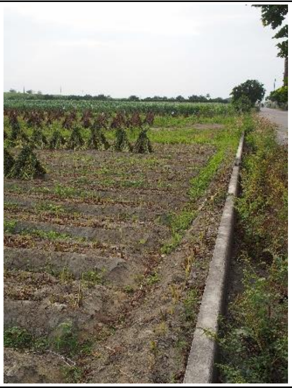
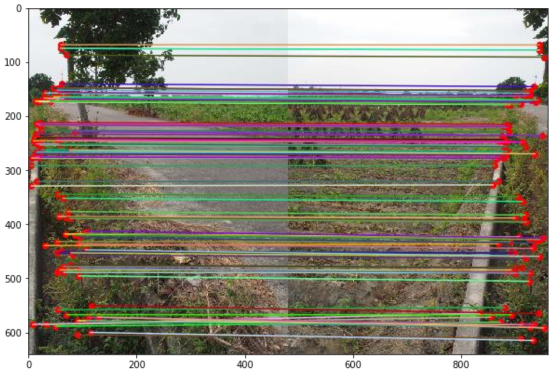
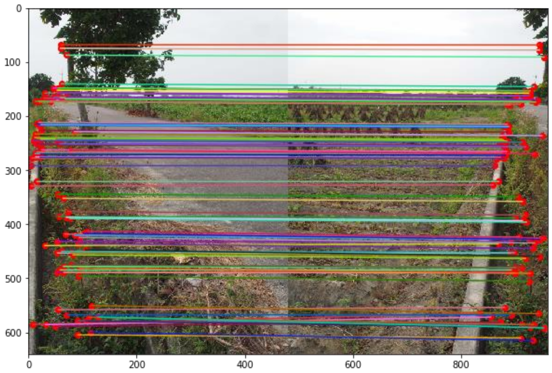
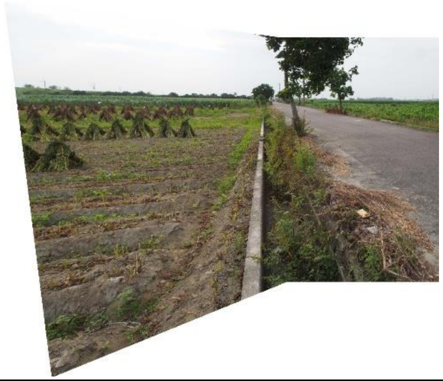
Match points w/o RANSAC	
Match points w/ RANSAC threshold=1	
Our panoramic image	
Result of build in function	

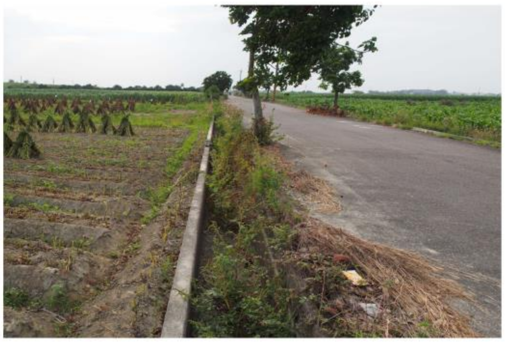
Original image		

Match points w/o RANSAC	
Match points w/ RANSAC threshold=20	
Our panoramic image	
Result of build in function	

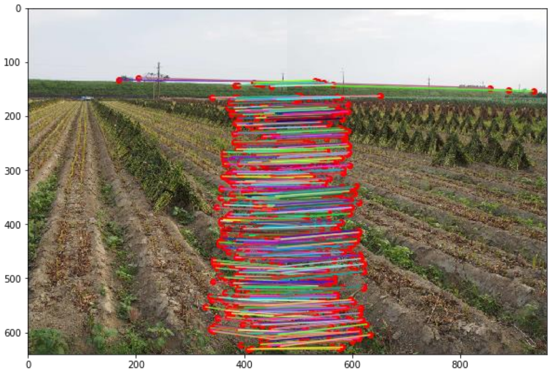
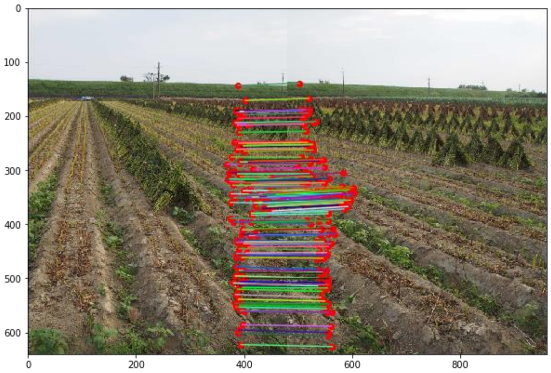
- The result of our image

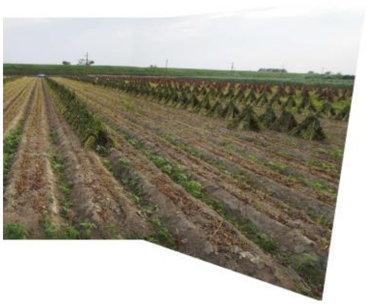
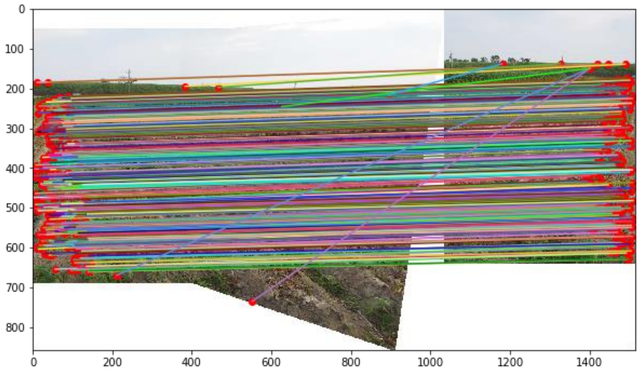
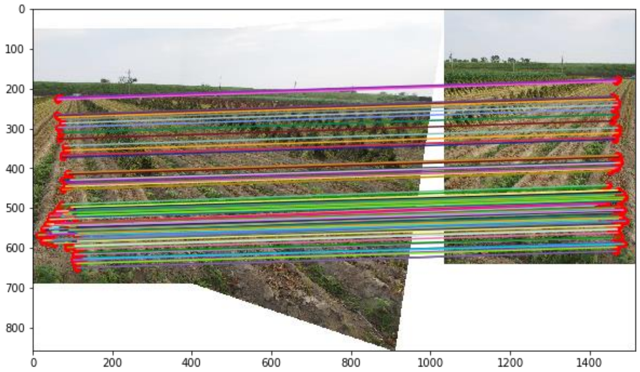
Original image		
Match points w/o RANSAC		
Match points w/ RANSAC threshold= 1		
Our panoramic image		
Result of build in function		

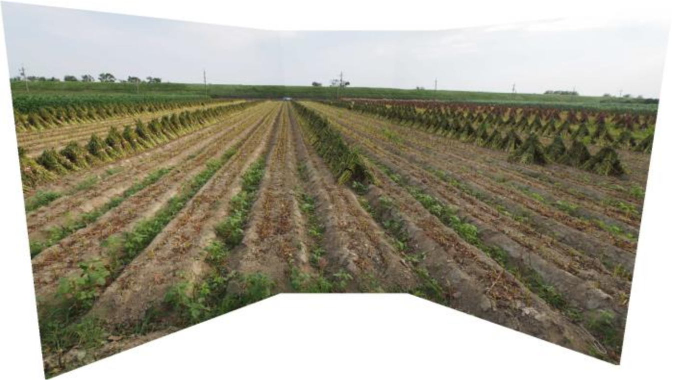
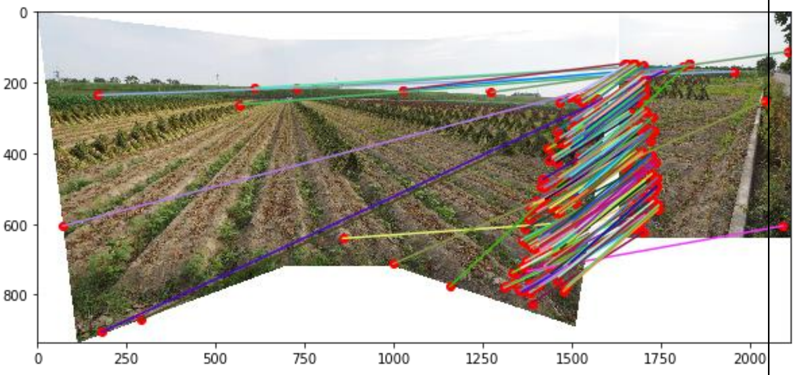
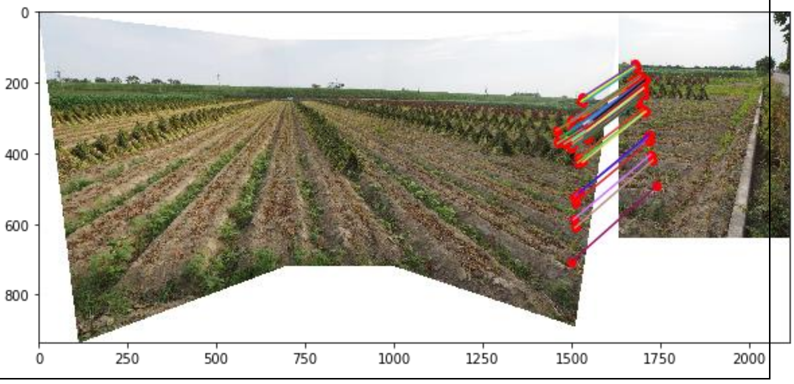
Original image		
Match points w/o RANSAC		
Match points w/ RANSAC threshold=1		
Our panoramic image		

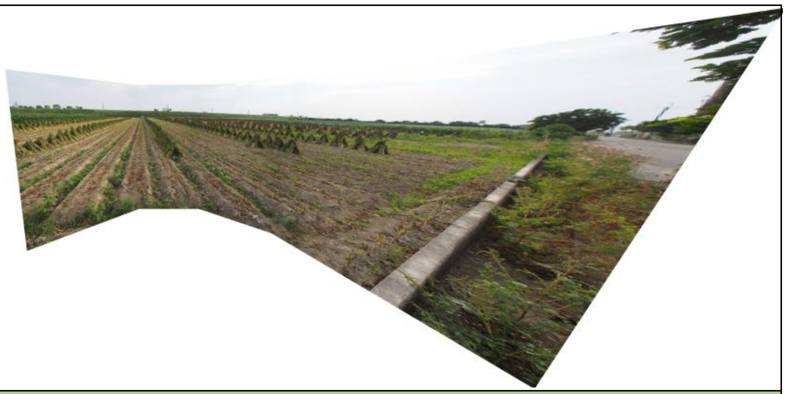
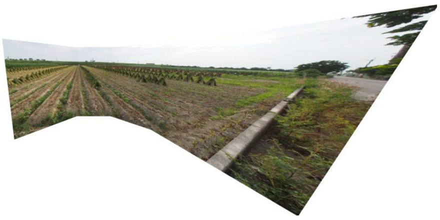
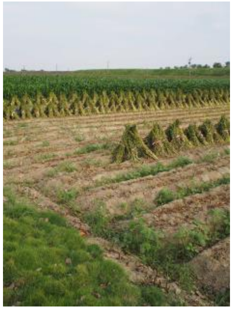
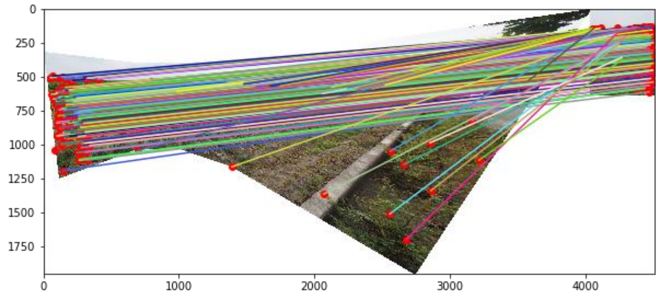
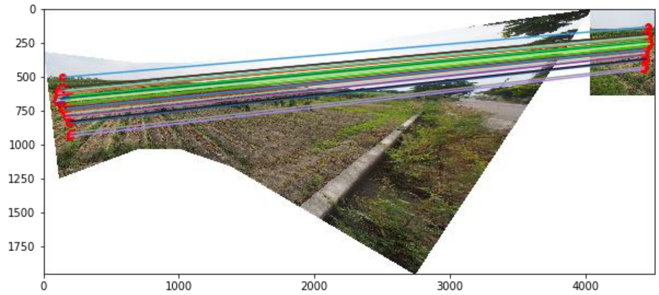
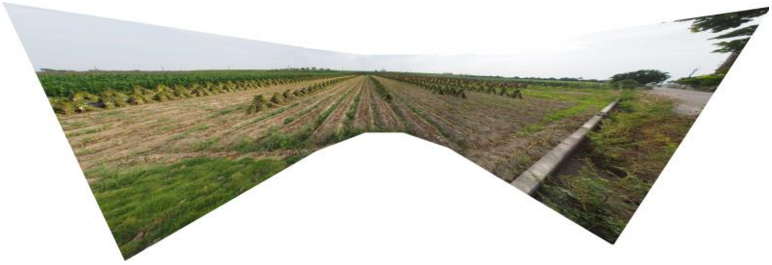
Result of build in function	

- Result of stitching multi-image

stitch 2 images		
Original image		
Match points w/o RANSAC	 0 100 200 300 400 500 600 0 200 400 600 800	
Match points w/ RANSAC threshold= 1	 0 100 200 300 400 500 600 0 200 400 600 800	

Our panoramic image		
stitch 3 images		
Original image		
Match points w/o RANSAC		
Match points w/ RANSAC threshold=1		

<table>
<tr><td>Our panoramic image</td><td colspan="2"></td></tr>
<tr><td colspan="3">stitch 4 images</td></tr>
<tr><td>Original image</td><td> </td><td> </td></tr>
<tr><td>Match points w/o RANSAC</td><td colspan="2">
</td></tr>
<tr><td>Match points w/ RANSAC threshold= 1</td><td colspan="2">
</td></tr>
</table>

Our panoramic image		
stitch 5 images		
Original image		
Match points w/o RANSAC		
Match points w/ RANSAC threshold= 1		
Our panoramic image		

D. Discussion

1. SIFT is no longer available in OpenCV > 3.4.2. In order to call cv2.xfeatures2d.SIFT_create(), we reinstall OpenCV 3.4.2 by following.
   - ➢ pip install opencv-python==3.4.2.16
   - ➢ pip install opencv-contrib-python==3.4.2.16
2. We don’t know what datatype KeyPoint in OpenCV is, so I google it and get the answer.

```
Class KeyPoint {
    Point2f pt;
    Float size;
    Float angle;
    Float response;
    Int octave;
    Int class_id;
};
```

3. We tried to stitch more than two picture. It took a lot of time to generate it. We even thought that our code had some problem to stitch more than two pictures. However, after a long while, the result looked great. We think this is due to the reason that if the image which taken at larger angle of rotation than previous image, it would get larger after warping. The larger the warp image is, the longer we need for our code to stitch it on target image. This is reasonable since some part of our code deal with pixels. The larger images have more pixels to process.
4. The first edition of our code had some bugs. The result did not look good. However, we did not aware it is our code’s problem. Thus, we tried to make some artificial features to help our code find better matches between images. After we found these bugs and fixed them, we found that there is no need to use these artificial features. Our code can get an excellent result without any additional features. Thus, all results shown on our report were generated with raw images.
5. When we do affine transform with random three points each image, it would raise an error because of singular matrix. We deal with that problem easily just using ‘try’ and ‘except’ to skip.

E. Conclusion

1. If ratio is too big or too small, we cannot get the good matching result. After testing many ratio value, we choose 0.5 to do feature matching.
2. Images with more texture can get more feature points. This helps a

lot to match images. Those picture with smooth objects has less good pairs to match. Thus, images with great texture have a better result than those without.

3. We can stitch more images, but we would take a lot of time to stitch with the image which taken at large rotation

F. Work Assignment