Topic: Primitive Panorama Stitching

Name: Dazhi Li
NetID: dl939

Basic procedures:

For this homework, I took two pictures which have something in common. In other word, they have some overlaps. Only when we have common features in pictures, we can stich them together. Then I use opencv library to transfer both pictures to gray depth figures. Then I can use Sift feature extraction which is also supported by opencv to grab those features and descriptors out. After we extract the features hidden in the pictures out, I need match those features. Matching those features means I am trying to get the common points and marked them into pairs. There pairs are important evidence for us to do combine two figures together. Then I use RANSAC to estimate the homography based on match results. Finally, I will stitch the images together to get a result.

Requirement:

opencv-python-3.4.2.16
opencv-contrib-python-3.4.2.16
matplotlib-3.3.0

Code example:

See attachment

1. **Using a digital camera, take two images from the same position but with different angles.**

   In this part I took two slightly different pictures in Rutgers University Livingstone Campus. I have marked those two pictures with different tags. The picture which is on the left side of the panorama image is called query image. This image will not be rotated or pitched in the result. And there is another picture on the which is on the right side of the panorama image is called train image. This image will be slightly rotated or pitched based on our estimation of homography to be stitched with the query image.

   I took two beautiful autumn school scenes in the Livingstone campus and they are shown below in Fig 1.

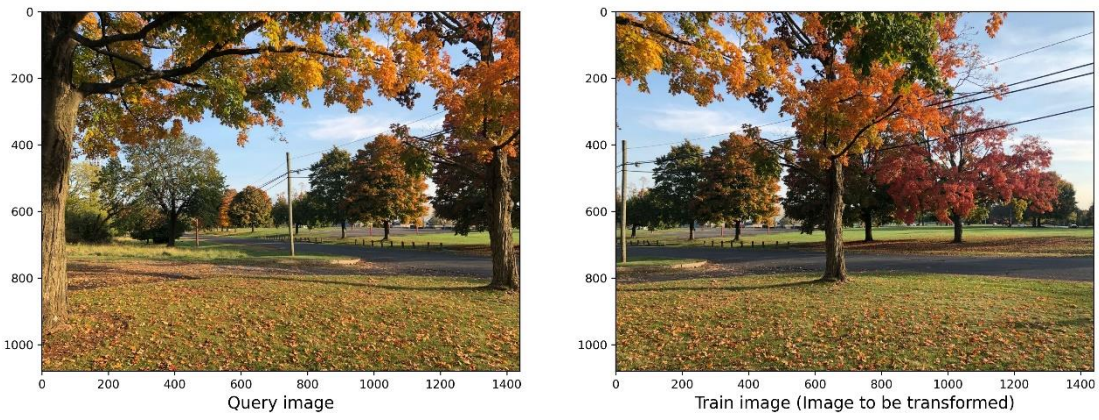
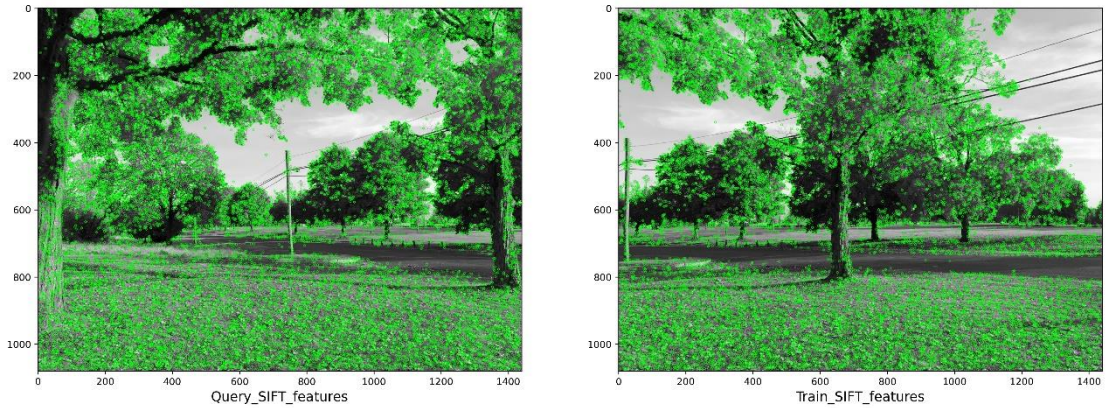


Fig 1 Two pictures taken in Livingston Campus

2. **Find the SIFT-key points and descriptors for both the images**

In this part I am going to find the SIFT-key points and descriptors from my pictures. In other words, I need to extract features from my pictures. To extract the SIFT key points, I need transform my pictures to gray depth pictures first. Then I use opencv library which already has built-in SIFT feature extraction function. Then I use opencv to draw those key points on the gray depth pictures which is shown below in Fig2.



As we can see those green circles are SIFT features extracted from the gray depth figure. There are a lot of common SIFT features in both picture because the overlapping. If we want to stich the two pictures together, we need to find the overlapping part, which will be talked next part.

3. **Match the correspondence points. Include pseudocode. Show visually these matched pairs of points.**

The main idea of finding the overlapping between two pictures is matching those SIFT key points. There are a lot of ways to achieve this. Here I used the brute force matching. As I know, people can even use KNN matching which is another method to achieve key points matching. Brute force matching is not a comprehensive method. Computer will try every key points on the query image and every key points on the train image to try if they are similar and how similar they are. Opencv also has a built-in BF(Brute force) matcher to accomplish this. One argument I would like to mention is the cross check. If we set the cross check to true, which means we need not only the query image features match the train image features but also the train image features match the query image features. Simply speaking, it is a stricter matching method if you said yes to the BF matcher.

After I find those matches between the train image and query image, I must count how many matches found. If there are only few matches found, which means those pictures do not have something in common. That will throw us into a huge problem that, we cannot stitch those two pictures. If there are lot of matches found, bingo! Your pictures could be stitched based on your match result. In my example, there are totally 13168 matches found. And here I will just show you the 100 best matches in my pictures.

Fig 3 BF matching: 100 best matches

4. **Run RANSAC to estimate homography and stitch the two images together and include the final image in your report.**

   After we find corresponding points in two images, we need to calculate how to stitch them. At first, we need to do some little data manipulation. We need to sort the key points in both train image and query image as the same order in BF matching result. Then we can sue RANSAC to estimate the homography could be. Opencv has a built-in function named findHomography. This function can use RANSAC method to achieve finding the homography based on SIFT key points and matches. This function will return a transform matrix. This transform matrix contains the information on how to rotate or pitch the train image. Before I stitch the two images together, I must calculate the size of new figure. Here I just use the horizontal stitching example, which is adding two pictures width together to form a new width. And taking the maximum height as the new height. Therefore, I use the transform matrix on the train image in my new figure and adding the query image on the left of my image. Finally, a stitched panorama image is created!

Fig 4 Horizontal Panorama Stitched image