EECS 442: HW 4 Image Stitching

Part1: stitching pairs of images

1. Solution:
   1. First, input two images using imread(), and change them into grayscale images by rgb2gray.
   2. Develop the get\_feats. Here I used the harris.m function to calculate the features. I u Ised sigma and radius as 3 to get the best performance. The threshold I chose was 1000 which return around 250 features per picture.
   3. Develop the get\_matches solution. Here I used the pdist2 function in Matlab rather than the dist2 function which is provided by the homework. With the parameter of ‘euclidean’ and ‘smallest’ , it can return the smallest 2 values of the Euclidean distance between to two sets of features. Then I calculated the ratio distance of the smallest distance over the second smallest distance and select those less than 0.8. I also implemented the visualization method to match the corresponding features scatter on the picture which is discussed in discussion session: making a shift for the features of the image2 and use a color matrix to mark the relative features in two different images.
   4. After import the features we get to get\_transform, I designed the algorithm for RANSAC. I randomly pick four matched pairs and calculate the transform matrix of them. And I cast the transform matrix to all the features in image2. And calculate the deviation between two set of features. If the deviation is less than 0.01 after normalize, it will be considered as inliers. This is the session of one loop. The program randomly picks up 4 points each time and repeats the loop for 3000 times to get the max inliers and use the transform matrix as the output, as long as the number of inliers and avg\_residual.
   5. The final stage is to warp the image2 by the projective matrix we get. I used the [right,xdata\_range,ydata\_range]=imtransform(img2,H), thus I can use xdata\_range and ydata\_range to determine the size of the output image. Then I use the eye(3) matrix to do the transform for image1, and add them together. I chosed the average number of pixel in the area where image1 overlaps with image2.
2. Data Report: Inliers: 158 Avgnumber:0.4328
3. Feature Match:

You can check the matched pairs by the color and the location.

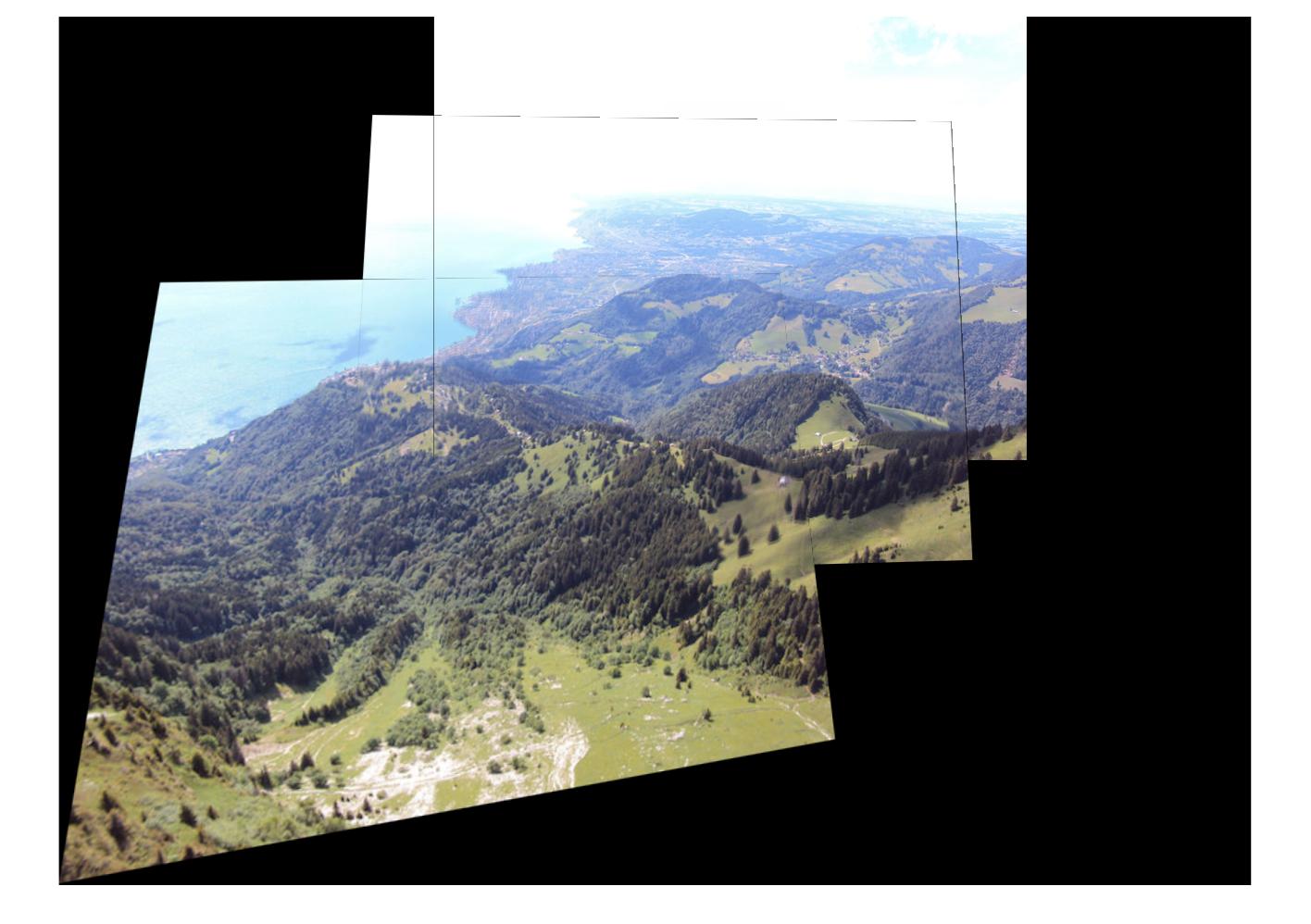
4.Result:

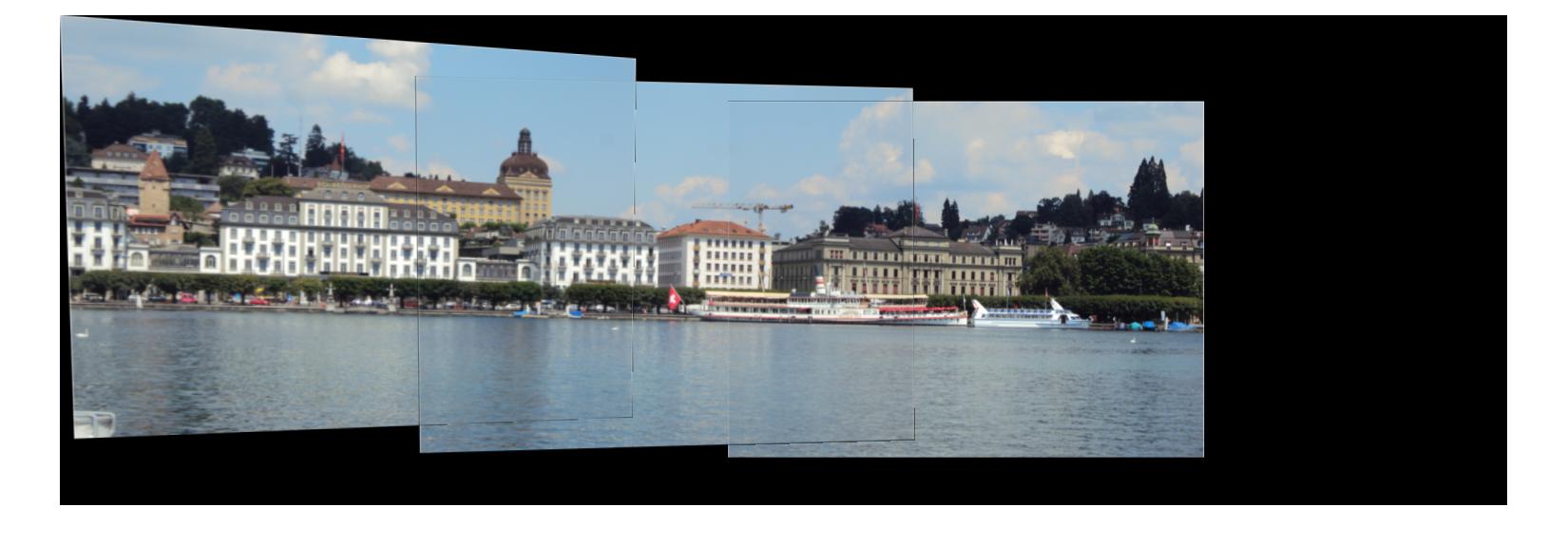
Part2: Multiple Image Stitching:

1. Solution:
   1. The basic algorithm is the same as in part1. The key point is choose the correct order of the images.
   2. I calculate the inliers between pictures, inliers12,inliers13,inliers23. The smallest value of the inlierXY means the XY should not be at the middle of the three pictures since they own less common features. Then I can use stitch\_images recursively to stitch three pictures.
2. Result
   1. Hill order:1-2-3



* 1. Ledge order: 1-2-3



* 1. Pier order: 1-2-3