EECS 442 Homework 4

Part 1. Pairwise images stitching:

a. Brief description of algorithms:

Feature extraction: harris.m is used for feature extraction. both sigma and radius are set to be 3 and threshold are set to 0.01. And the find_sift function given where the enlarge_factor was 1.5 as it recommended.

Feature distance computation: when the features are extracted with harris.m, find_feats.m is used to compute the SIFT feature descriptor of every feature. Then, The Euclidean distance of SIFT features is taken as the feature distance. The Eucildean distance is calculated by dist2() function.

Putative matching: Since the dist2() function returns an matrix of feature distance, we just need to use the result and select the 200 smallest entries. The selection code is like:

```
threshold = 200;
[Y,I] = sort(Distance(:));
I = I(1:threshold,1);
[i, j] = ind2sub(size(Distance),I);
```

Which provides the putative matches, and the indexes of these entries provides the putative matches of two images.

Actually, I created another function get_transMatrix(A,B) to get the transformation matrix of A to B, which used the method shown in the lecture slides.

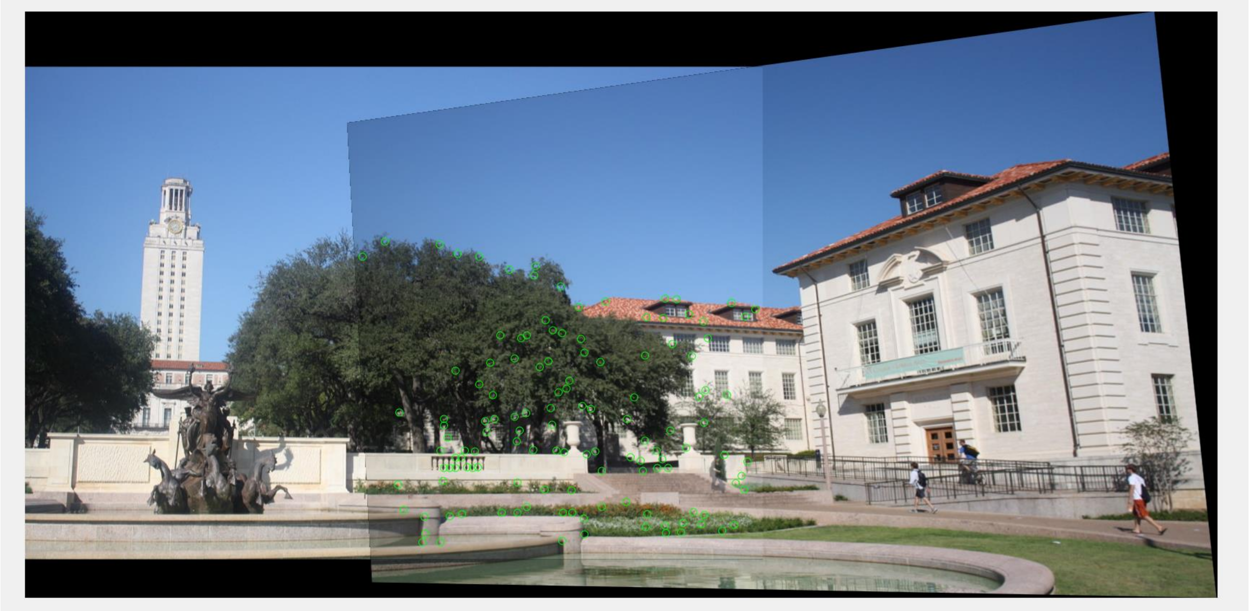
RANSAC: the algorithm of RANSAC is quite silimar to what the lecture slides said. The only difference is that the transform matrix is derived from all the inliers rather than the 4 groups of initial seed putative matches. In RANSAC, I called get_transMatrix() function in every loop to calcluate the transition of each seed putative matches set.

The loop run 10000 times. For each loop, it will choose 4 pairs of putative matches to calculate the initial transform matrix.

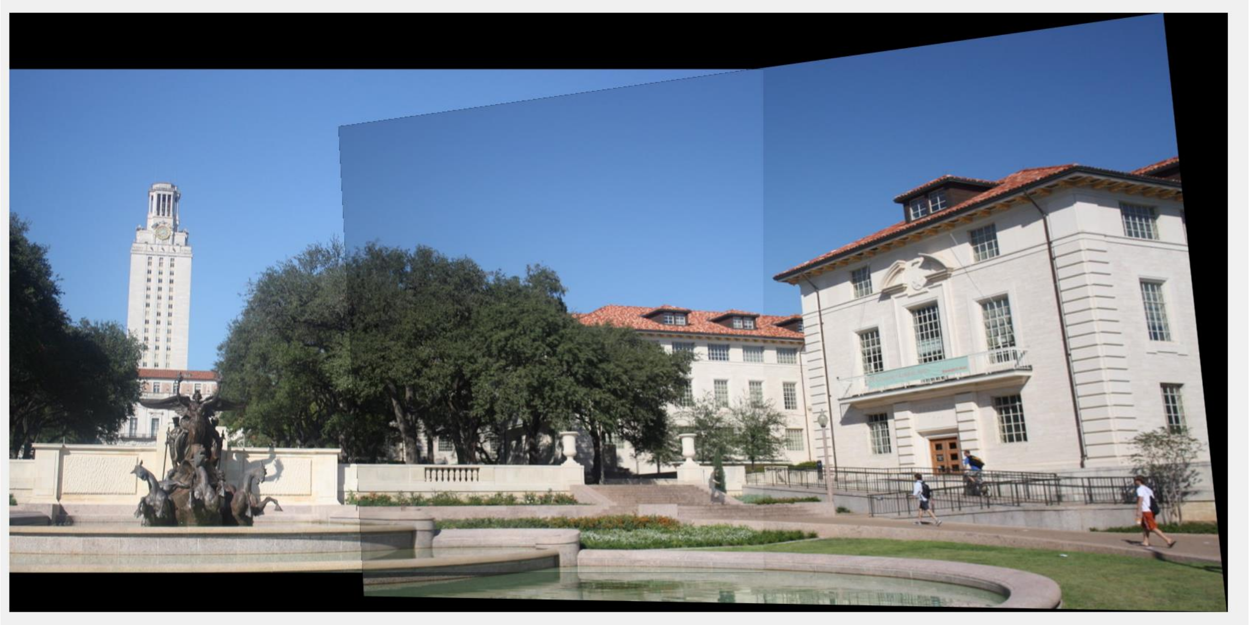
b. Number of homography inliers: 132; avg_residual: 2.6397

c. Images of pairwise images stitching:

With inlier matches shown:



Without inlier matches shown:



Issues appeared:

1. The boundary of each image in the stitched image is clearly seen, that is because, the index of the boundary is not an integer, which will cause the dark boundary appeared in the stitched image.

2. Sometimes the stitching/overlapping part of the image is not clear enough (sometimes blur appeared and the stitching results are not always the same, especially in the multiple images stitching). The reason is that the find_sift() function just provide us with non-rotation-invariant SIFT descriptors, thus, the dominant gradient orientation is not aligned, which will sometimes prevent us from getting the best stitching result. And the randomness of RANSAC is also a more significant reason.

Part 2. Multiple images stitching:

a. Brief description of algorithms:

Since there are three pictures in each set to be stitched together (denote as img1, img2,img3), the number of inliers of img1&img2, img2&img3, img1&img3 are calculated, the combination of the three above, which has the fewest inliers, denotes that the two images of the combination will not be stitched together. For example, if img1&img3 has the fewest number of inliers, then img1 cannot be directedly stitched with img3. Thus, we will firstly stitch img1 and img2 and denote the stitched image as img12, then we will stitch img12 and img3.

Three different situations are considered and the corresponding merging order:

situation 1: 2,3 cannot be stitched together:

first stitch img3 and img1 as img31, then stitch img31 and img2 as the final stitched image.

situation 2: 1,3 are not be stitched together:

first stitch img1 and img2 as img12, then stitch img12 and img3 as the final stitched image.

situation 3: 1,2 are not be stitched together:

first stitch img1 and img3 as img13, then stitch img13 and img2 as the final stitched image.

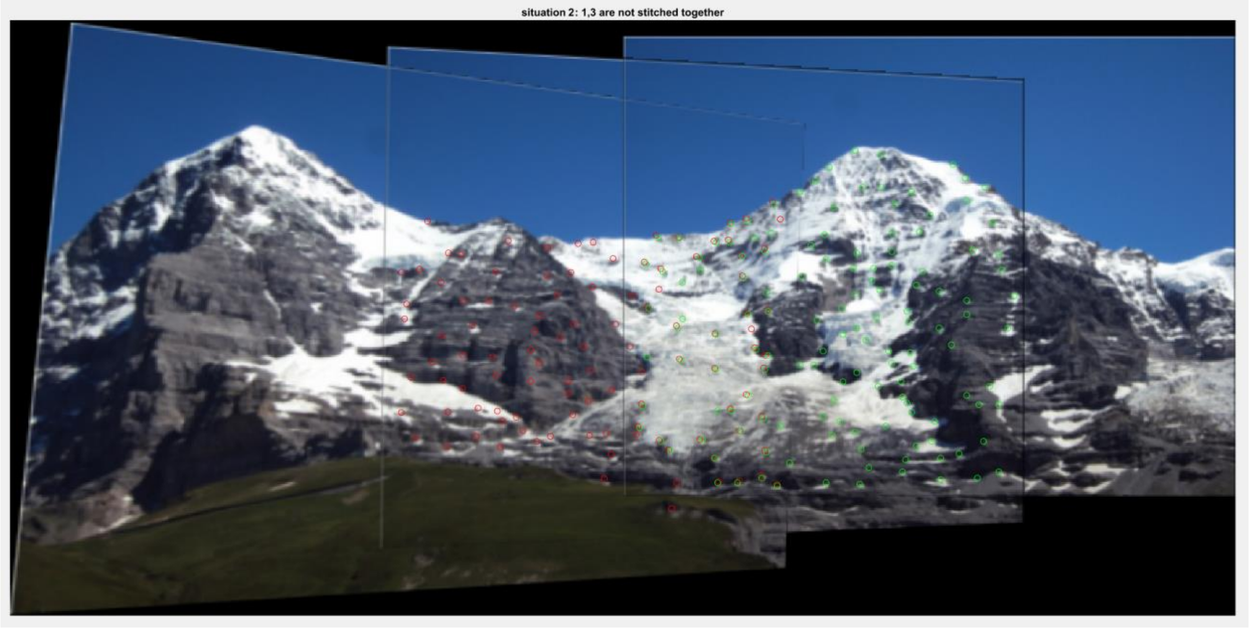
b. number of inliers(red denotes be chosen for merge):

	img1&img2	img1&img3	img2&img3
hill	110	47	139
ledge	191	110	188
pier	65	27	63

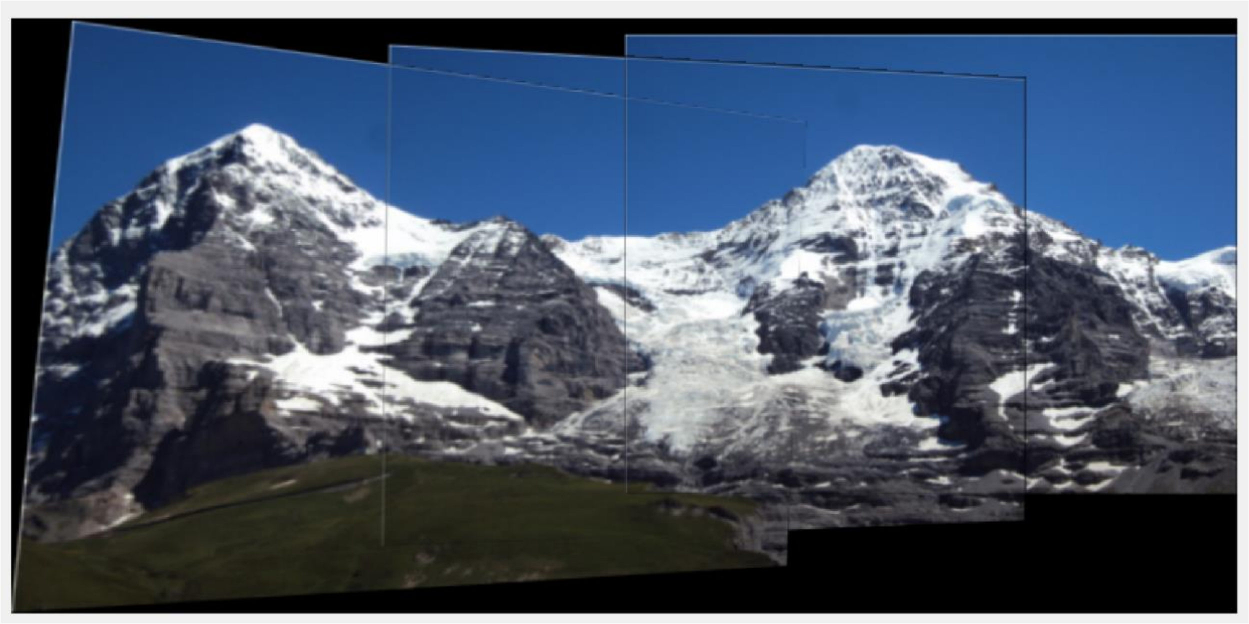
c. Images of triple images stitching:

Hill:

With inlier matches shown:

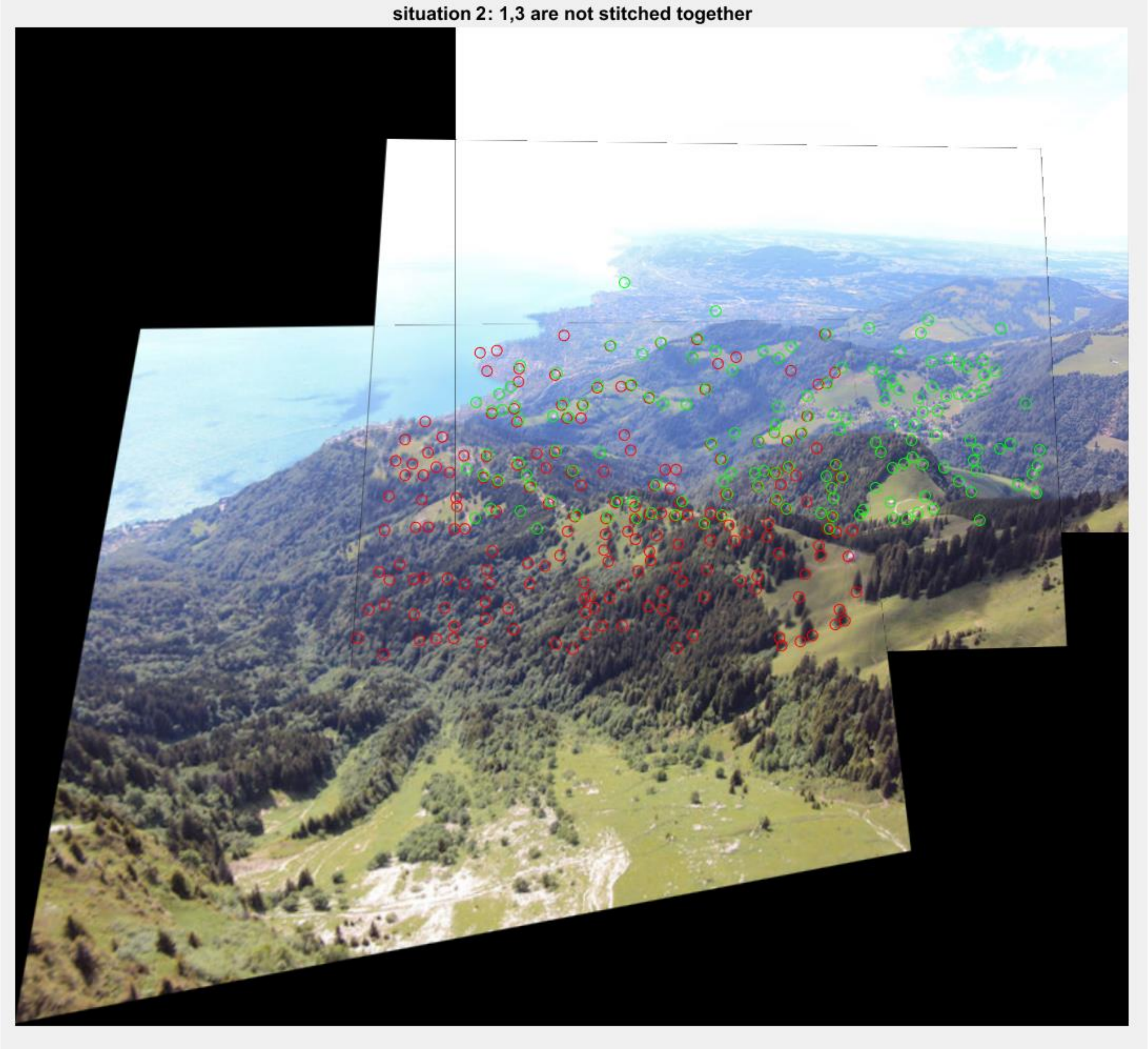


Without inlier matches shown:

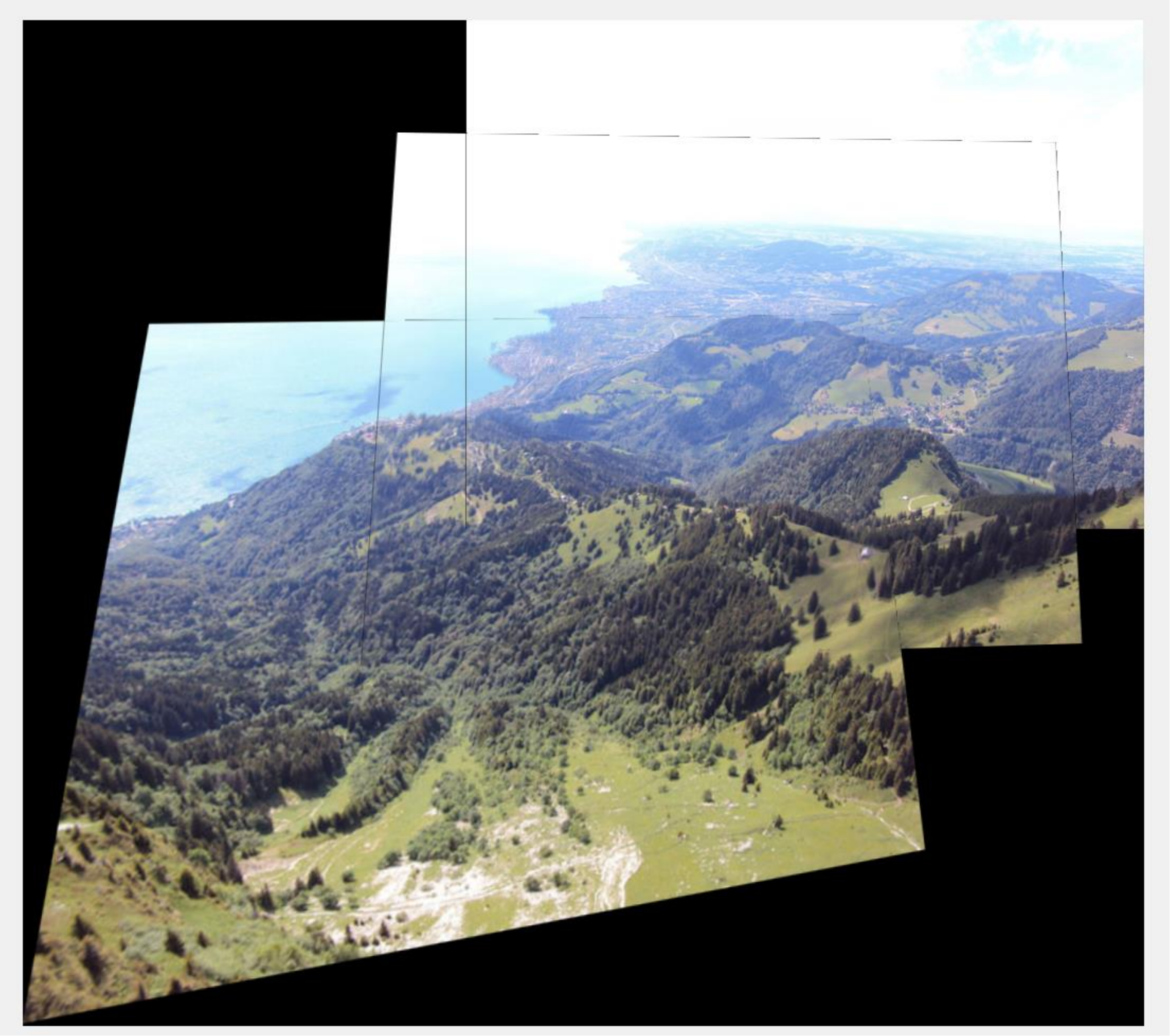


Ledge:

With inlier matches shown:

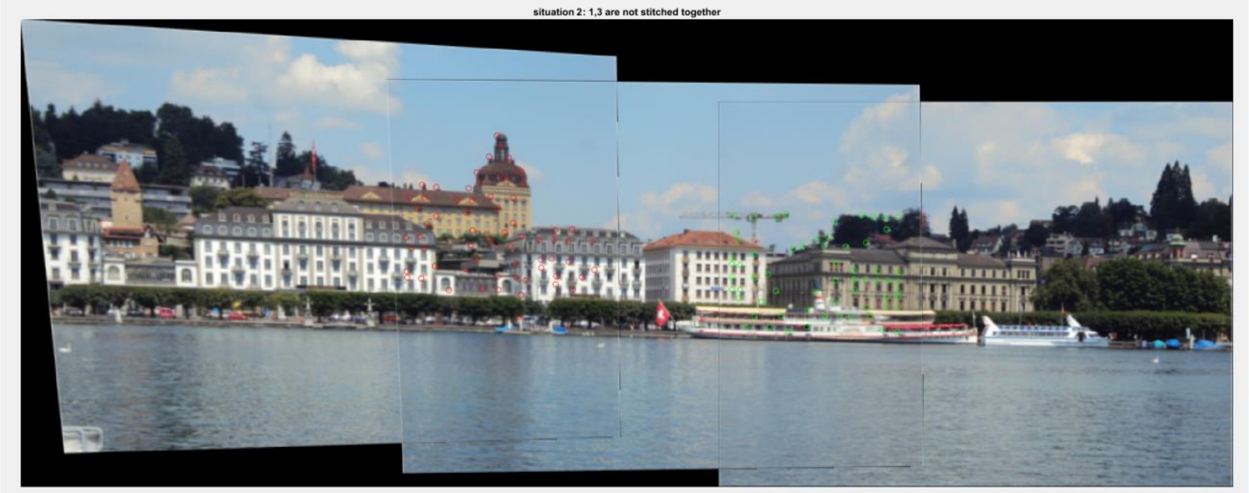


Without inlier matches shown:

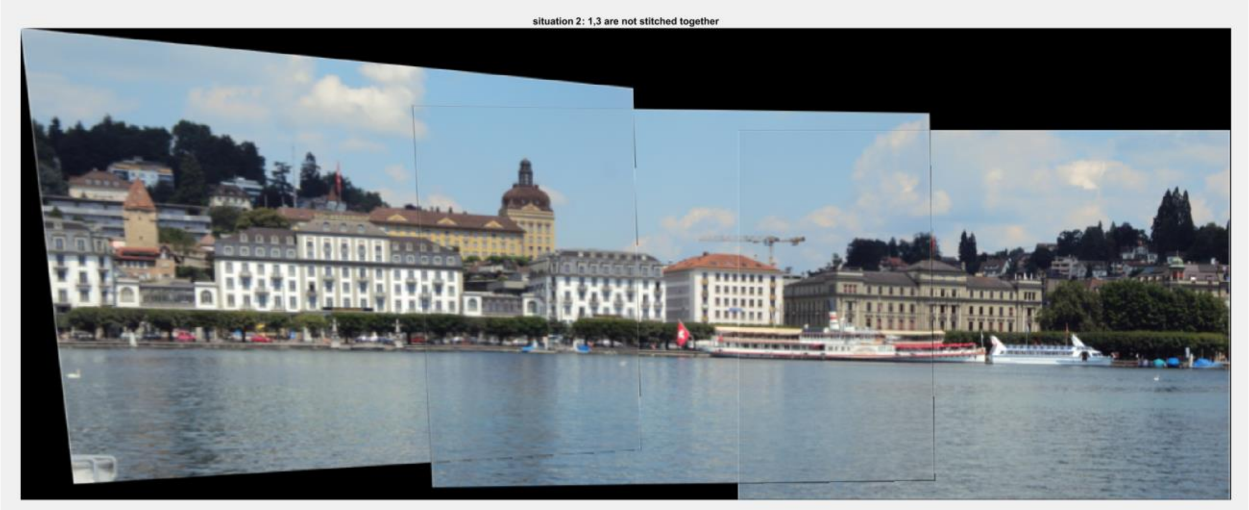


Pier:

With inlier matches shown:



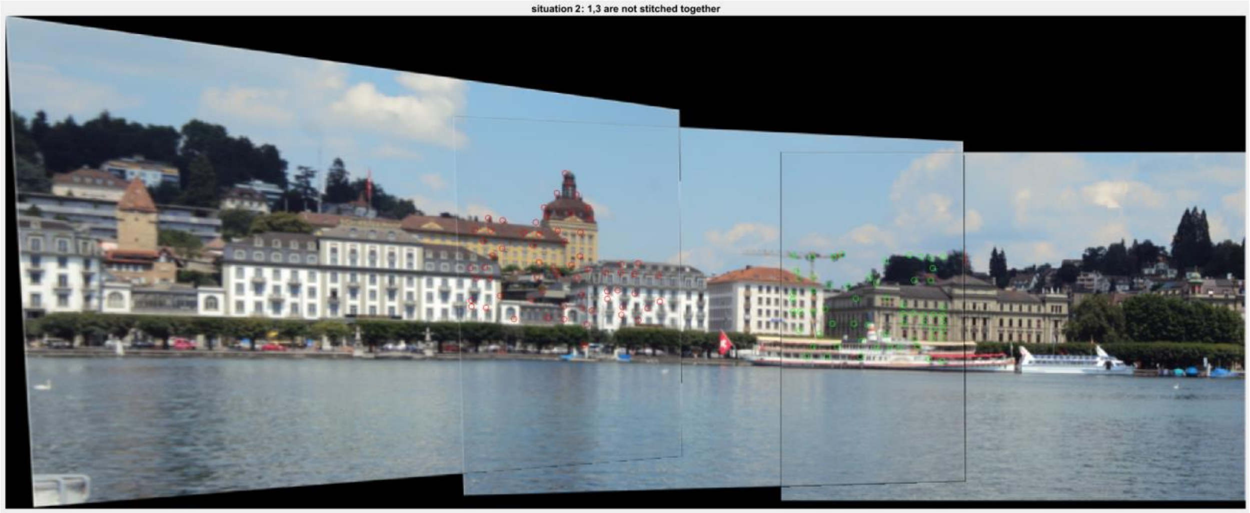
Without inlier matches shown:

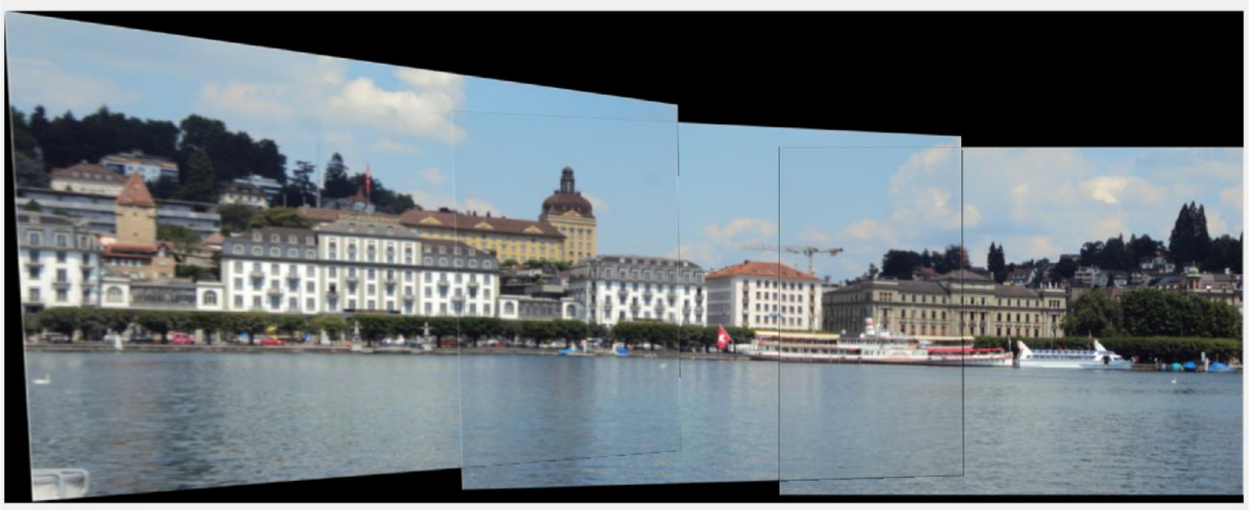


Issues appeared:

1. The boundary of each image in the stitched image is clearly seen, that is because, the index of the boundary is not an integer, which will cause the dark boundary appeared in the stitched image.

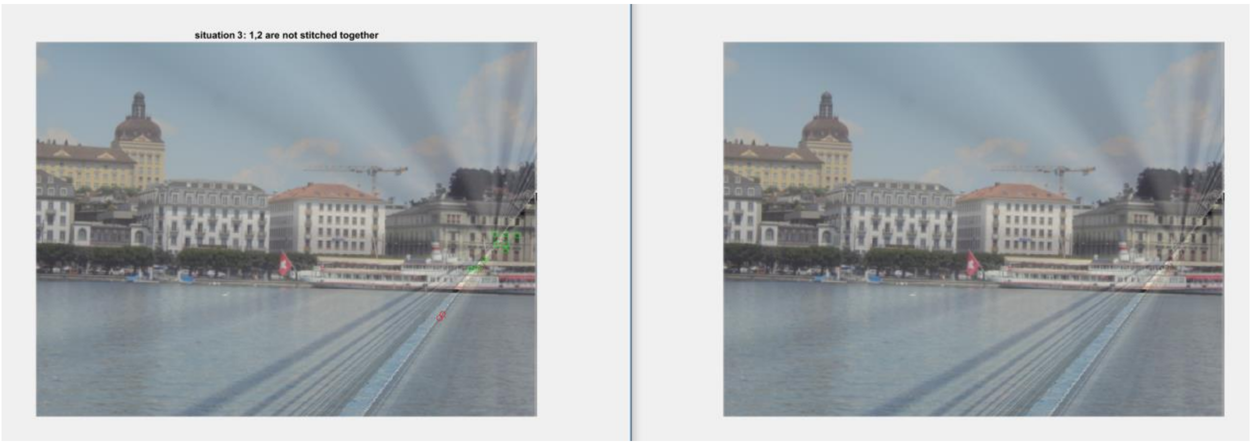
2. Sometimes the stitching/overlapping part of the image is not clear enough (sometimes blur appeared like shown in the below pictures of pier, especially the windows of the building in the stitching part). The reason is that the find_sift() function just provide us with non-rotation-invariant SIFT descriptors, thus, the dominant gradient orientation is not aligned, which will sometimes prevent us from getting the best stitching result. And the randomness of RANSAC is also a more significant reason.


situation 2: 1,3 are not stitched together



3. In pier set, if I set the threshold equals to 100 in my code, the number of inliers of img1 and img3 are a bit high, but img1 and img3, obviously should be directedly stitched together. That is because the arrangement and the feature of the window of the building are similar, which cause many ‘inliers’. Sometimes, in addition to the randomness of RANSAC and the SIFT unalignment, the three images are not stitched in the right order and cause result like the figure below. Setting smaller tolerance (25) in get_transform.m can solve the problem.

(Note: I just calculated the square of distance in my code, thus, threshold = 100 in my code is the same as threshold = 10 in the homework pdf of the instructor).



situation 3: 1,2 are not stitched together