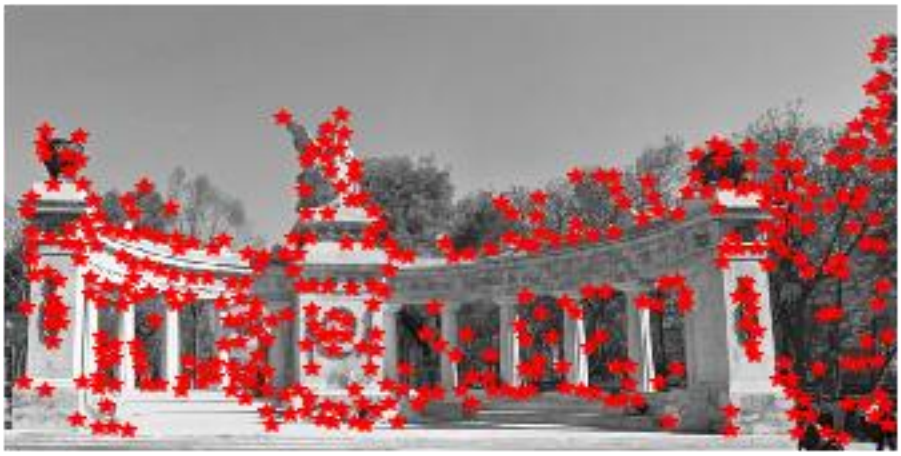
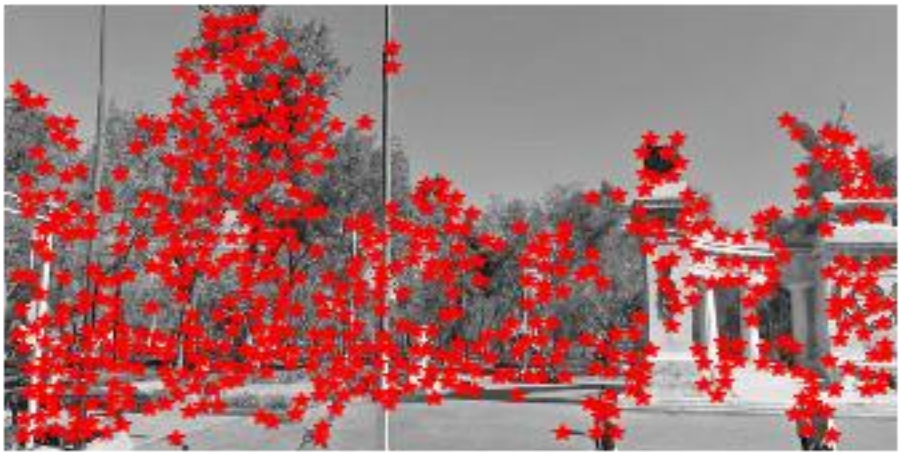
Part 1:

For this part, I firstly tried to use Harris detector to detect key points shown as below

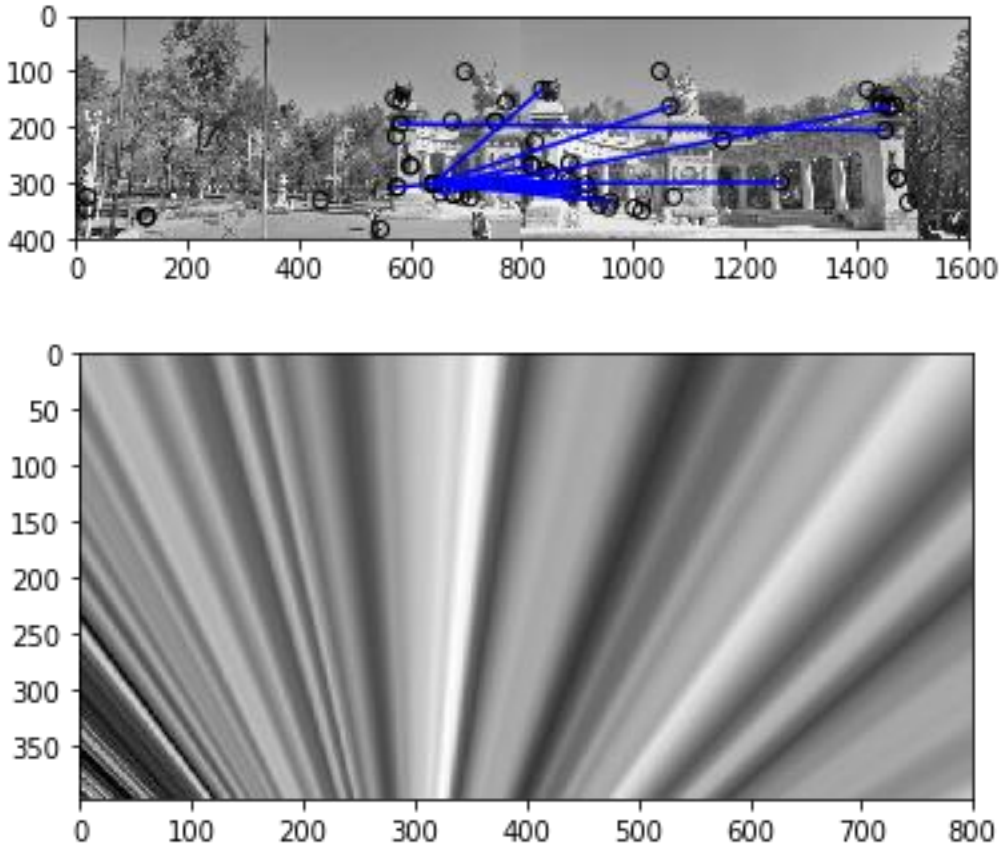




Then I extract the descriptors using a window size and compute the sqeuclidean distance between each pair of normalized descriptors to get the distance matrix. After performing a threshold to disregard matches whose distance is larger than it, I got the matches as below

I noticed that this approach is sensitive to the threshold and descriptors' window size. A larger windows size seems to be more stable whereas a smaller one will lead to more matches on the trees which are incorrect matches. Similarly, a larger threshold will increase the number of outliers significantly. But still, even with parameters tuned, the result matches still not good enough.

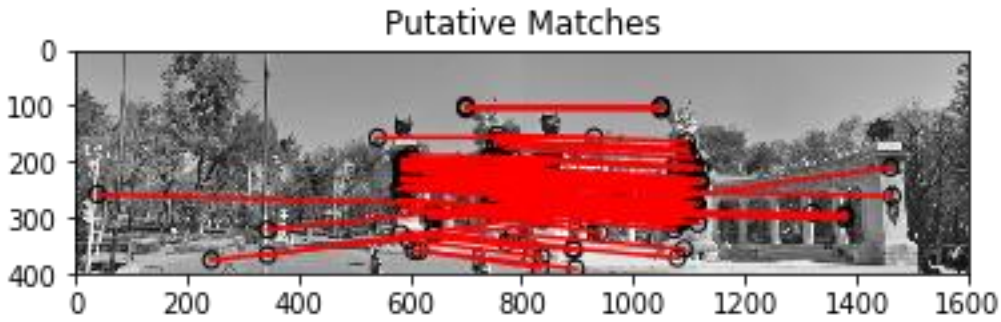
I performed RANSAC to fit the data. The inliers and warped images are shown below.



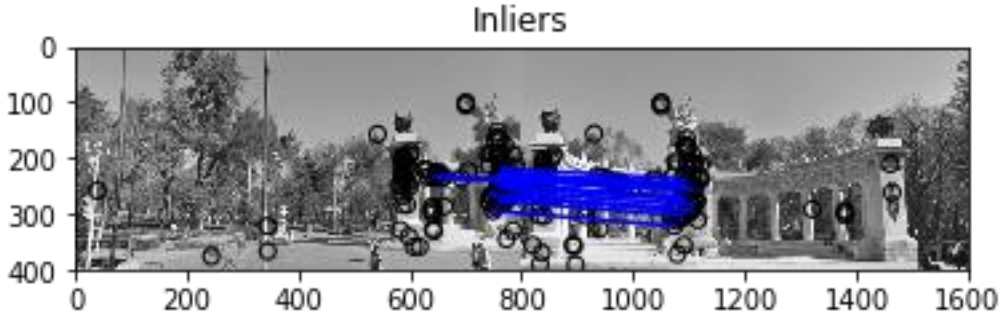
The result totally doesn't make sense. I also tried many different optimizations, but nothing works. After struggling for many hours, I gave up this approach and turned to use SIFT detector.

Similarly, I use SIFT detector to extract key points and descriptors. Then I get the pairwise distance matrix.

To get better putative matches, I tried a different approach than the one discussed in the lecture. I noticed that if we just apply a threshold to it, the output will still have many duplicates, where a feature in the first image matches with multiple features in another image. This is definitely not what we expect. What we expect is a one to one match. So, I apply a iterative improvement algorithm to get the best reasonable one to one match (see stable_match() in class match, also see stable marriage problem). By doing this, every feature in the image who has fewer features has a match to another image. Then I also apply a threshold to eliminate those matches with large distance. The putative matches are shown below.



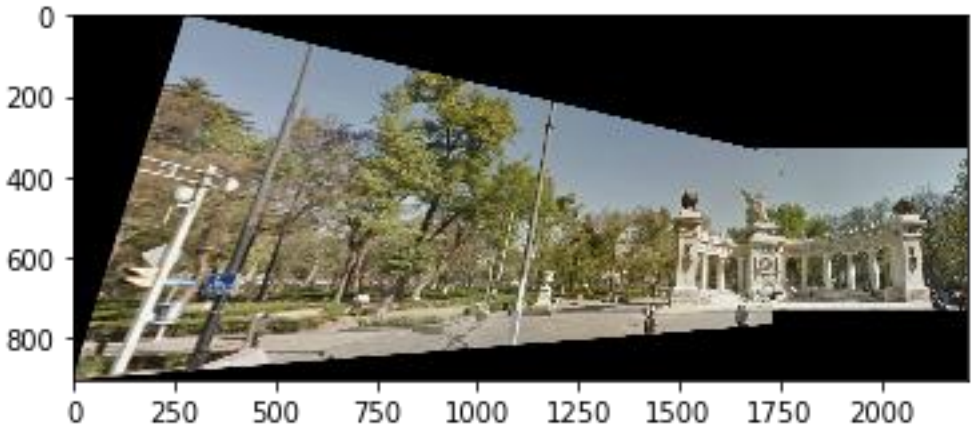
It looks very nice and much better than before. It does not have any outliers around the trees. Then I use RANSAC to compute the homography transformation matrix. For the RANSAC, I implemented the adaptive version as shown on the slides. The result is shown below.



Average inliers residual : 0.02167237010992018
Num of inliers : 16

When I try to get the warped image, I noticed the cut off phenomenon. The approach I used to solve it is to apply an affine transformation to the warped image to make it totally located in the canvas. The key point is to find the appropriate affine matrix H_affine. After finding it, we can turn H into H_affine @ H because x'' = H_affine @ (H @ x) = H_affine @ x'. The appropriate output image size can also be derived from the shape of the warped image after affine transformation (see stitch2() in class stitich for more details).

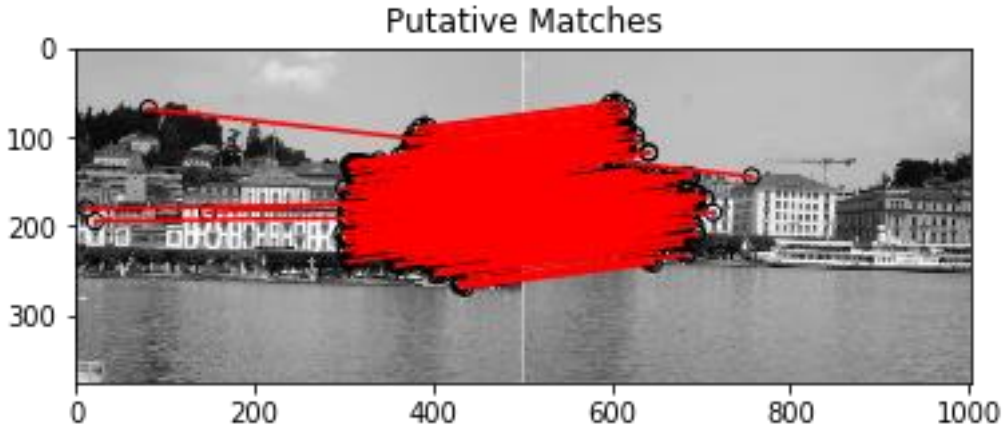
The result is shown below.

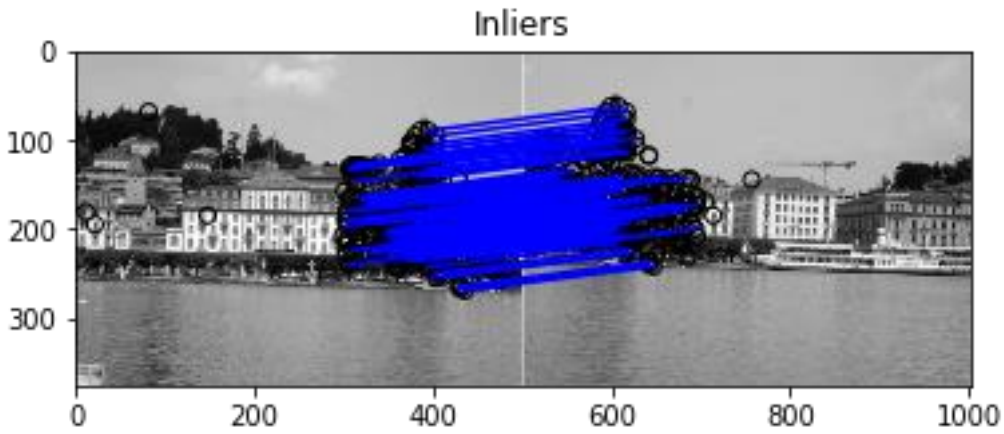


I also implemented a relatively more complete version of a system that can accept as many images as input and output the panorama image. The input images can be disordered.

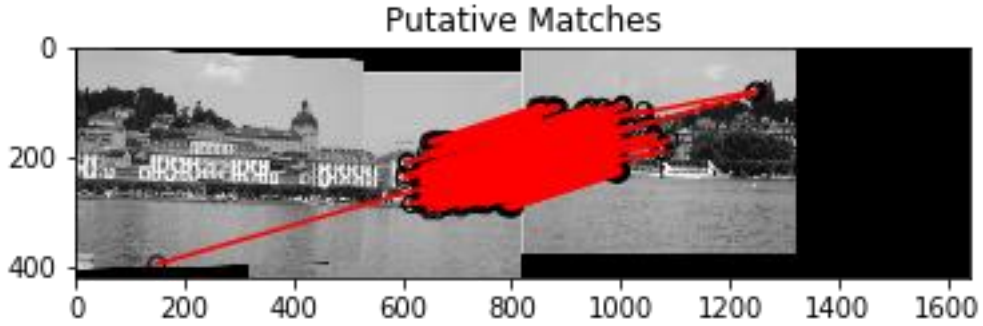
The results for all the other images are shown below.

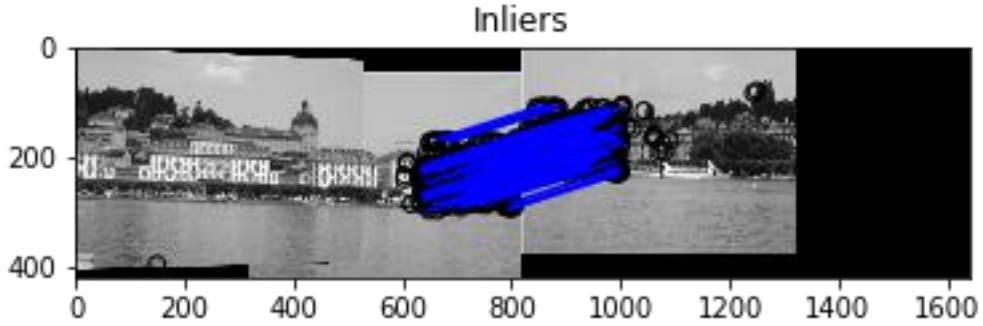
Pier:



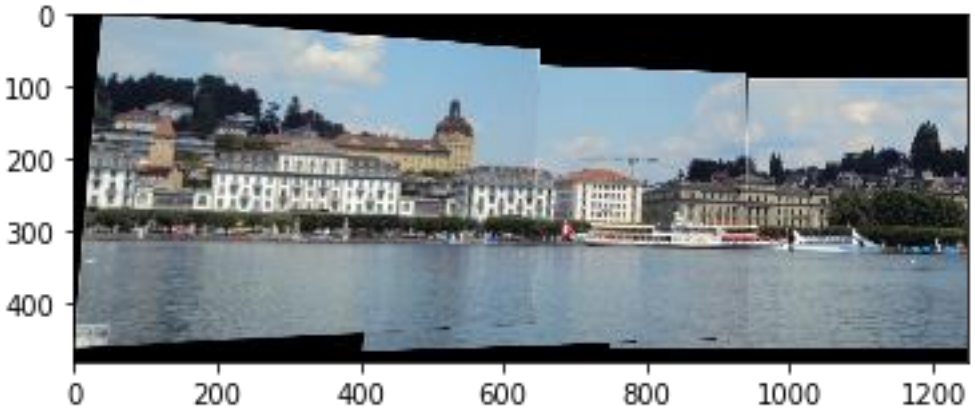


Average inliers residual : -0.016163405089134213
Num of inliers : 113

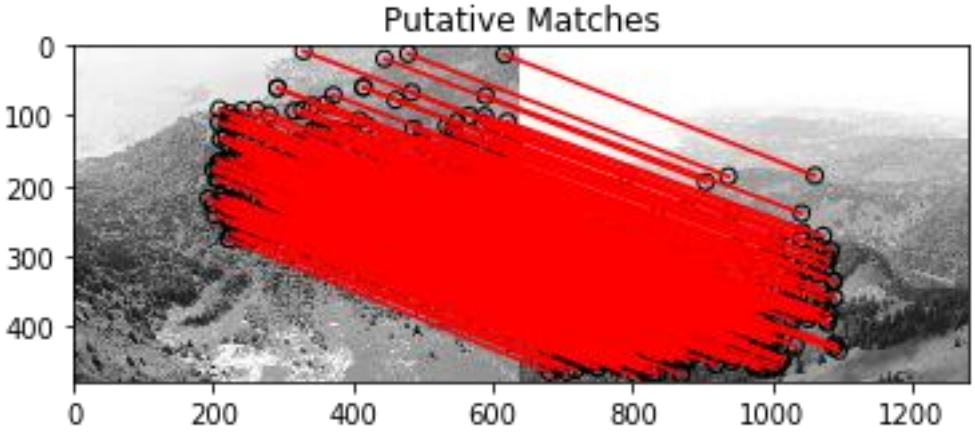


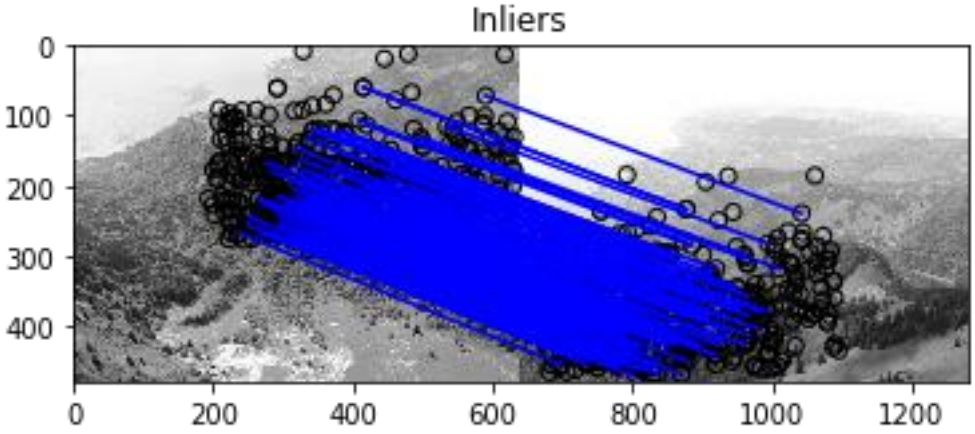


Average inliers residual : -5.387630590938538e-05
Num of inliers : 156

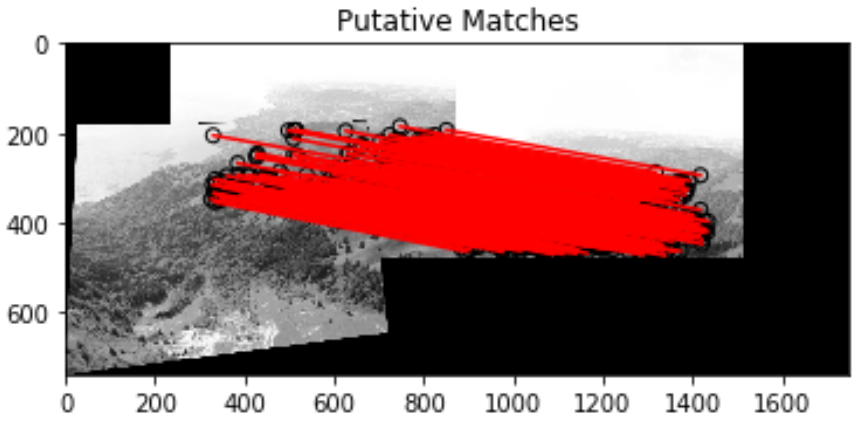


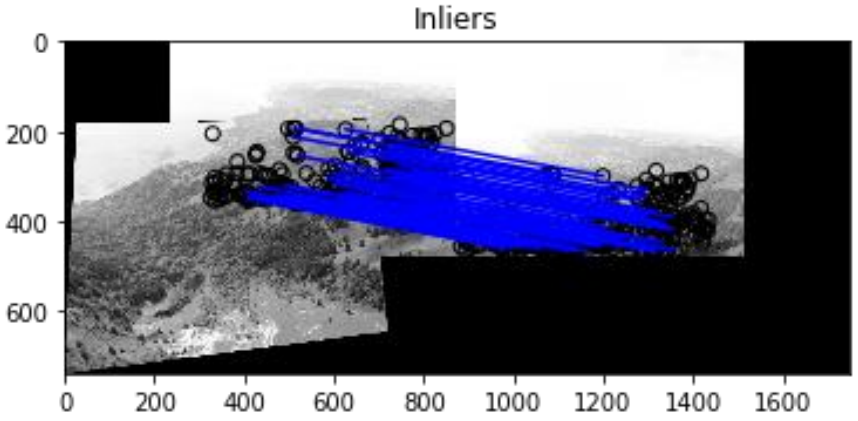
Ledge:



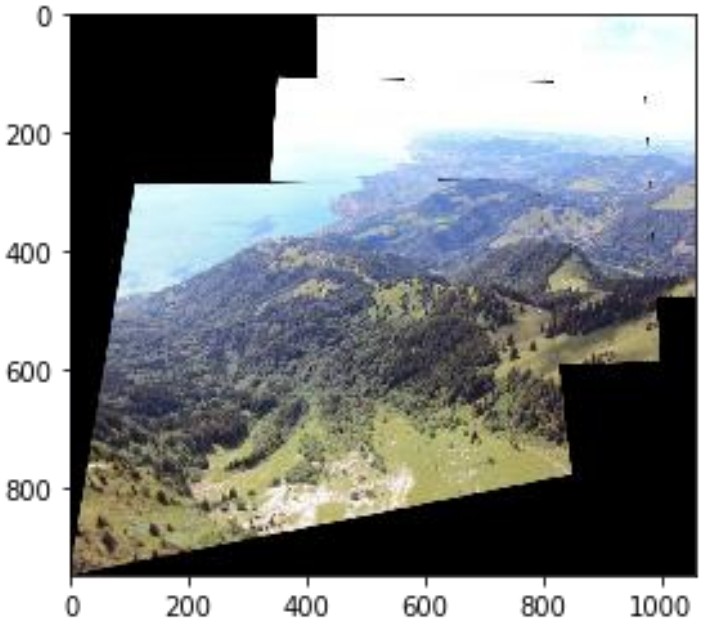


Average inliers residual : -0.03373963333556268
Num of inliers : 125

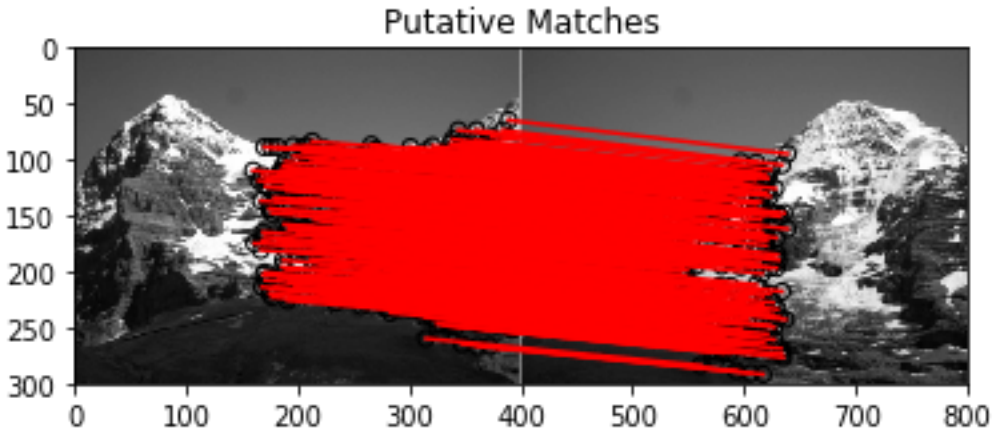


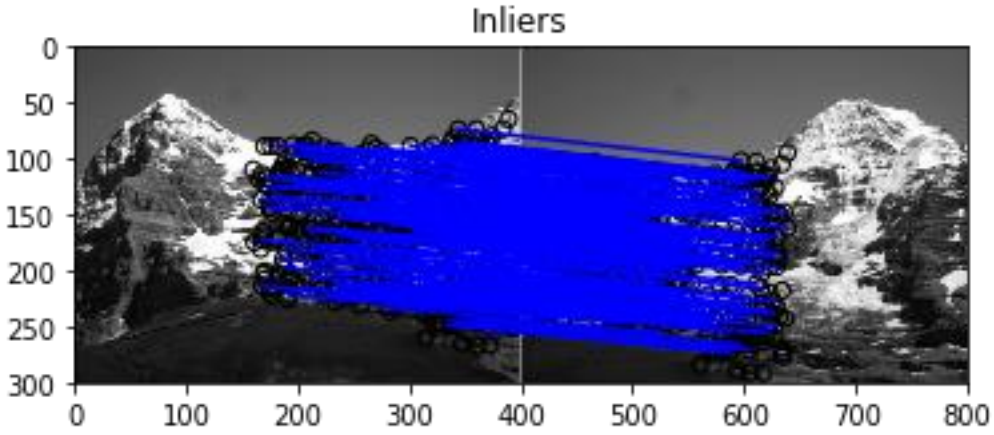


Average inliers residual : -0.0034294123514148583
Num of inliers : 70

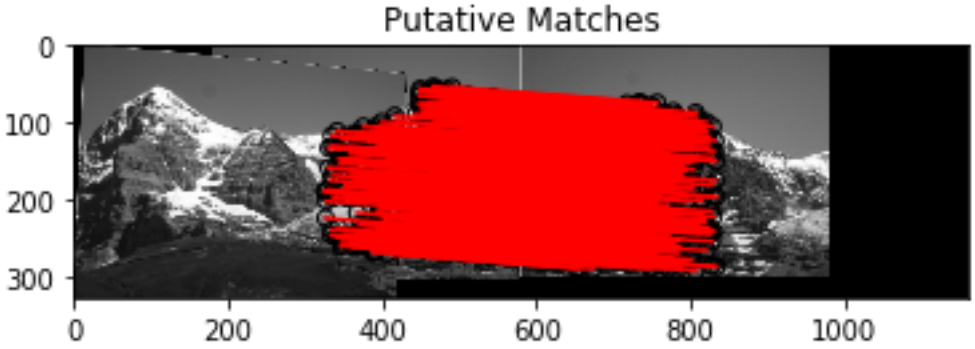


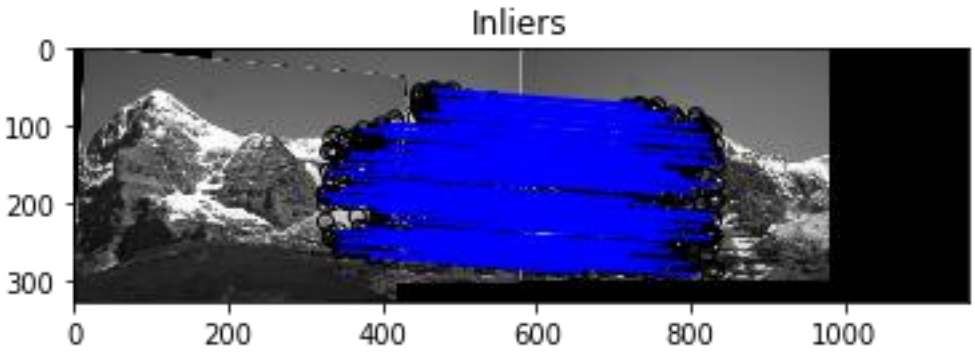
Hill:



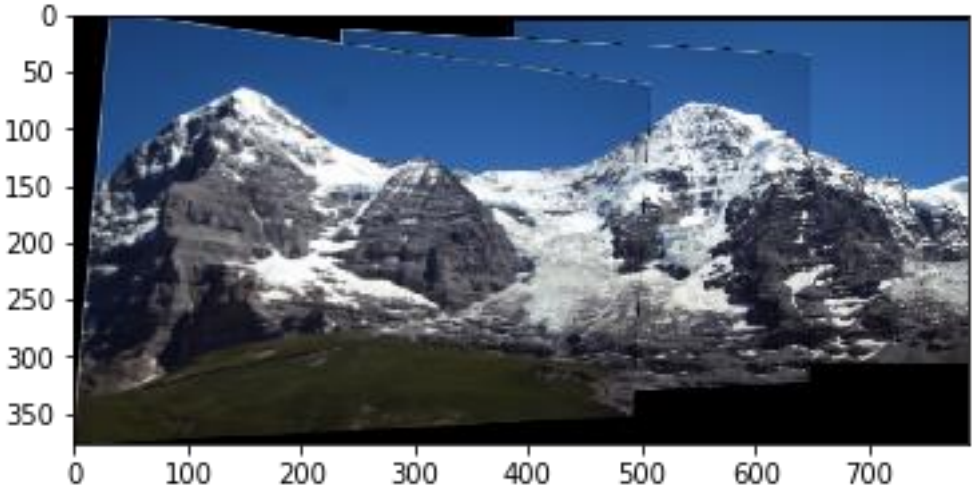


Average inliers residual : 0.015435940643672538
Num of inliers : 178





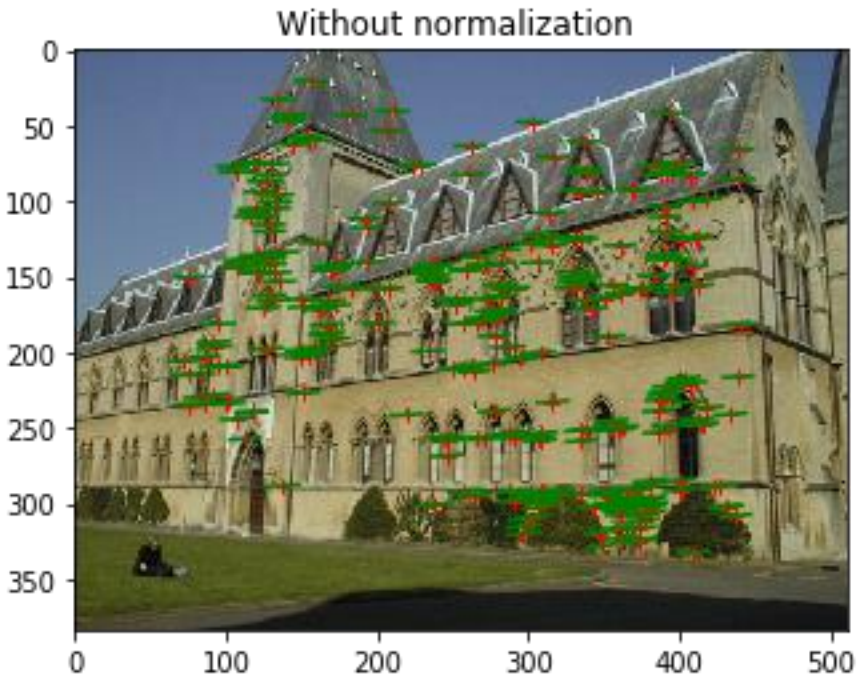
Average inliers residual : 0.01669235970948156
Num of inliers : 220



Part 2:

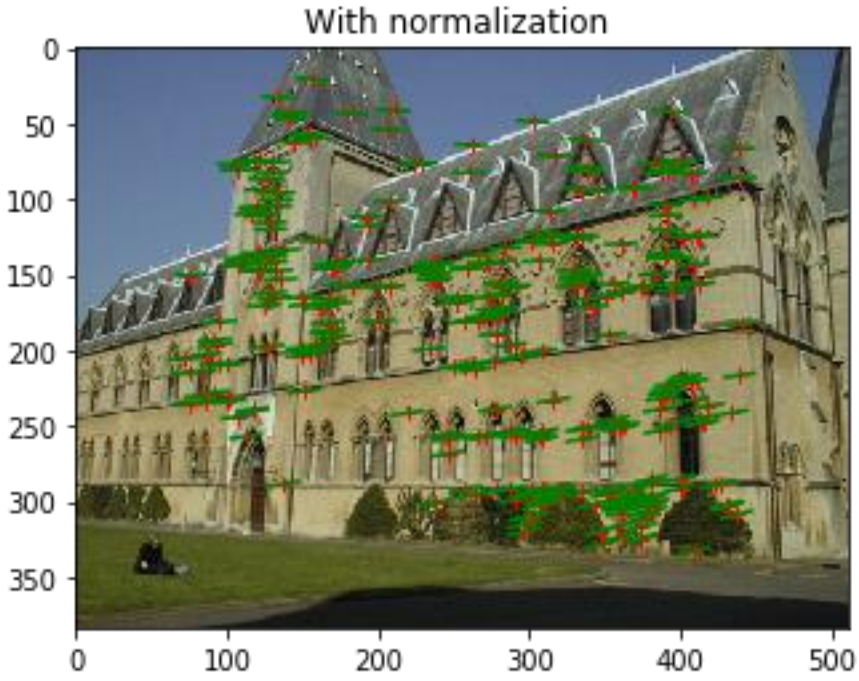
For this part, I write my RANSAC code but still use the ground truth matches.

The result points and epipolar lines of unnormalized fundamental matrix for library



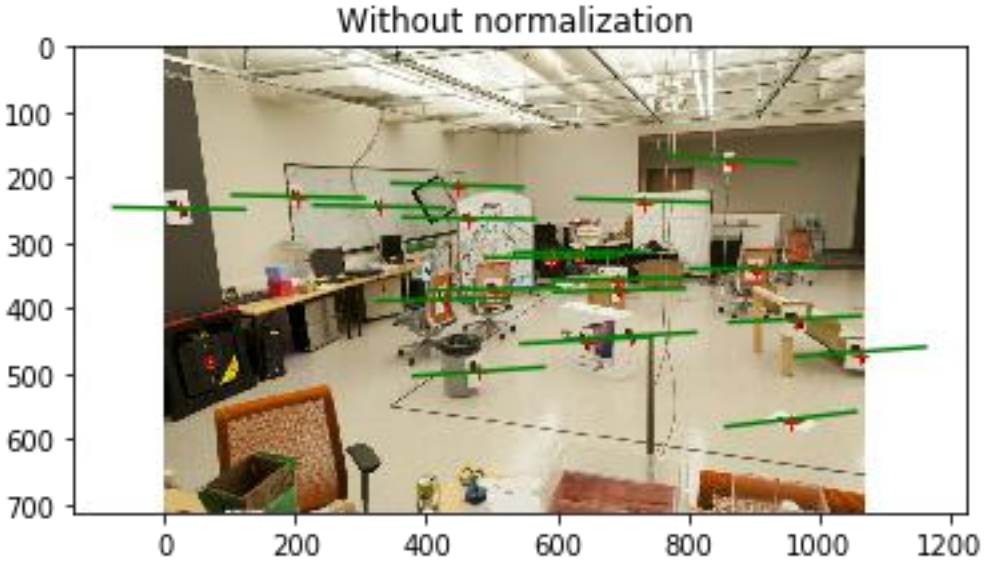
Average residual : 0.2570460821113142

The result points and epipolar lines of normalized fundamental matrix for library



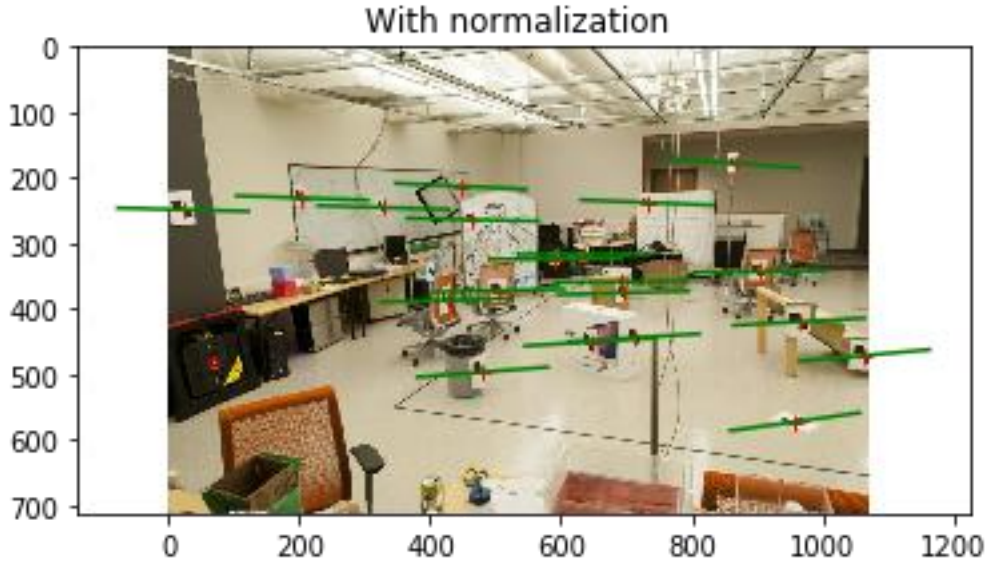
Average residual : 0.04975533578467253

The result points and epipolar lines of unnormalized fundamental matrix for lab



Average residual : 2.0925143029903195

The result points and epipolar lines of normalized fundamental matrix for lab



Average residual : -0.17715751749912415

The projection matrices of both cameras for lab and the residuals are shown belo w.

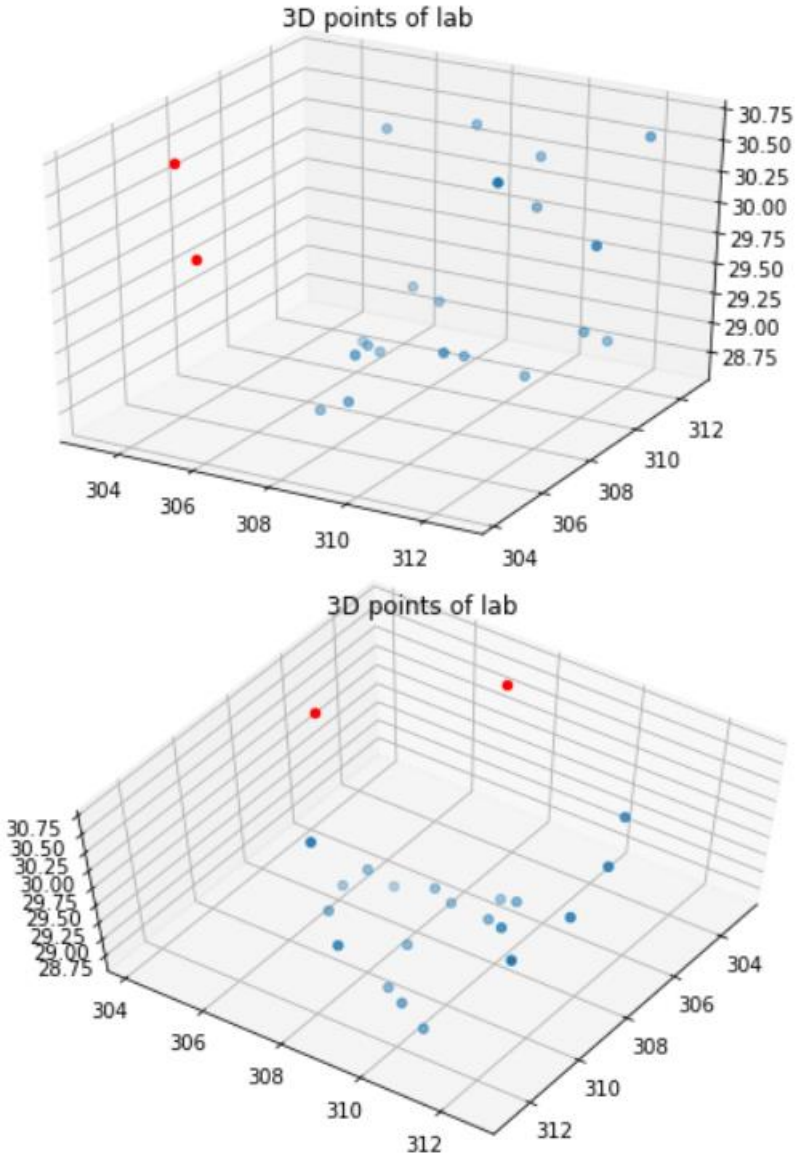
```
Projection matrix of camera 1 for lab :
[[-3.10362074e-03 -1.39774240e-04  4.11197867e-04  9.79225522e-01]
 [-3.01319438e-04 -6.38201720e-04  2.77147523e-03  2.02725458e-01]
 [-1.67080919e-06 -2.75197808e-06  6.36430680e-07  1.32883620e-03]]
```

Residual of projected 3d points and 2d points on camera 1 for lab : 0.7357448457 696192

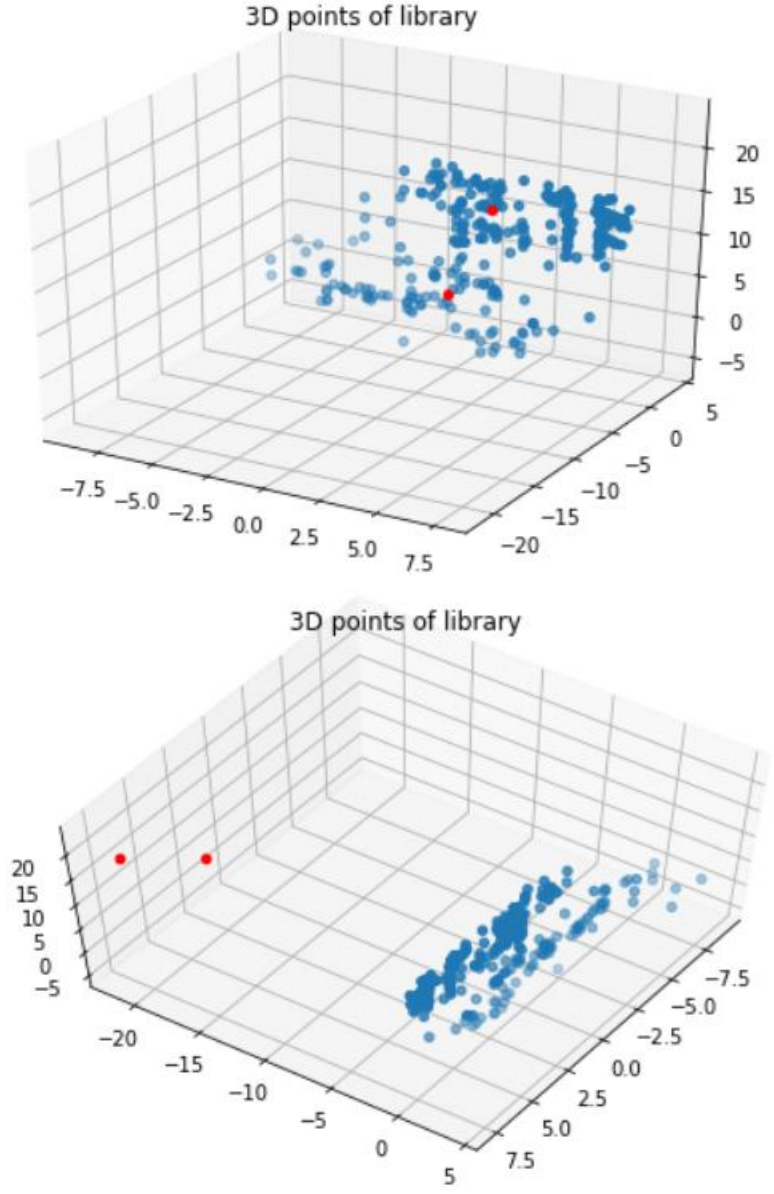
```
Projection matrix of camera 2 for lab :
[[-6.85684342e-03  3.93345809e-03  1.42248099e-03  8.26935919e-01]
 [-1.52984004e-03 -1.04059873e-03  7.27441373e-03  5.62178625e-01]
 [-7.52780003e-06 -3.72971553e-06  2.03830488e-06  3.36565373e-03]]
```

Residual of projected 3d points and 2d points on camera 2 for lab : 0.8181269975 334471

3D reconstruction points are shown below. Camera centers are shown as red poin ts.



Residual of 3d points of lab : 9.474273655934112e-05

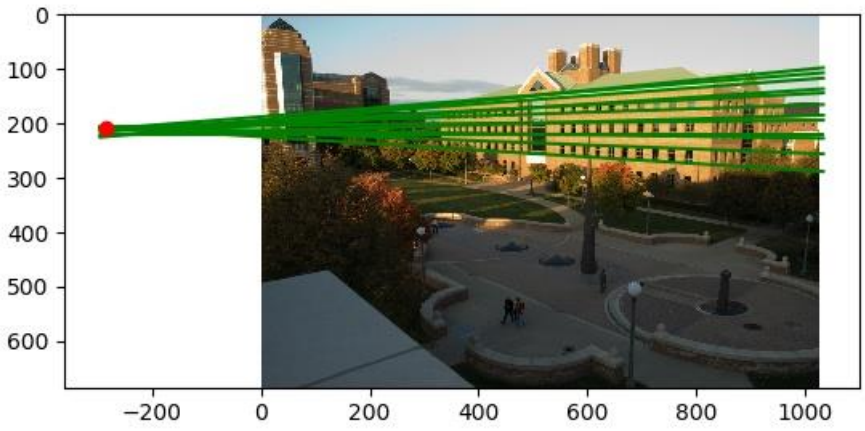


Residual of projected triangulation 3d points and 2d points on camera 1 for library : 0.079812528145207

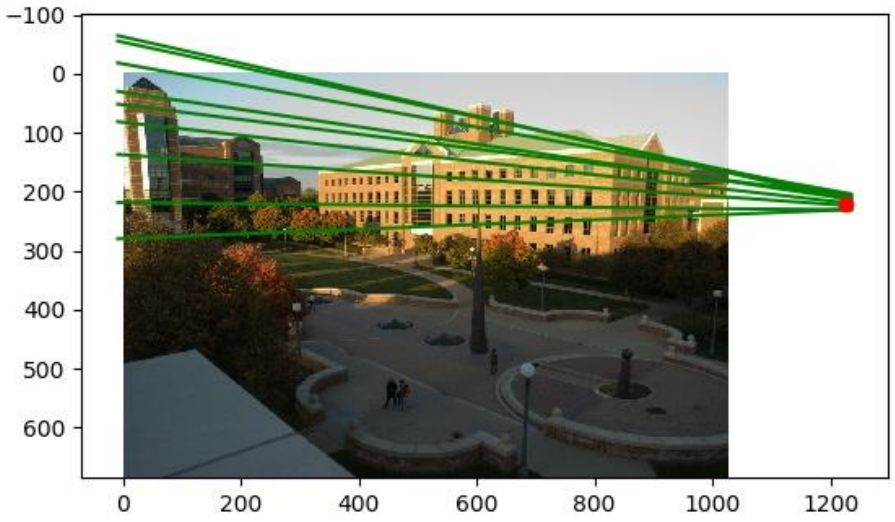
Residual of projected triangulation 3d points and 2d points on camera 2 for library : 0.092716950599549

Part3:

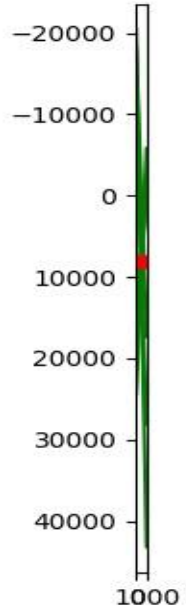
Vanishing points and the corresponding coordinates are shown below.



Coordinate: -285.32055908 209.46623593

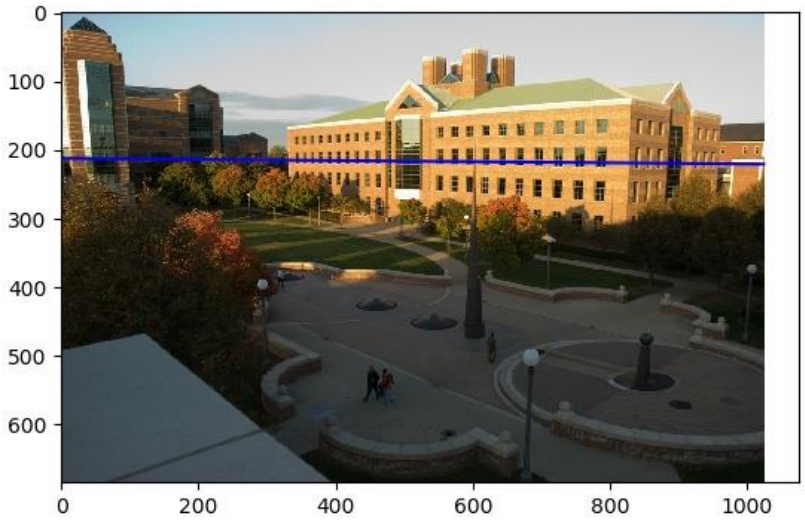


Coordinate: 1225.43591 221.943231



Coordinate: 494.936609 8126.49481

The horizon line and its parameters are shown below.



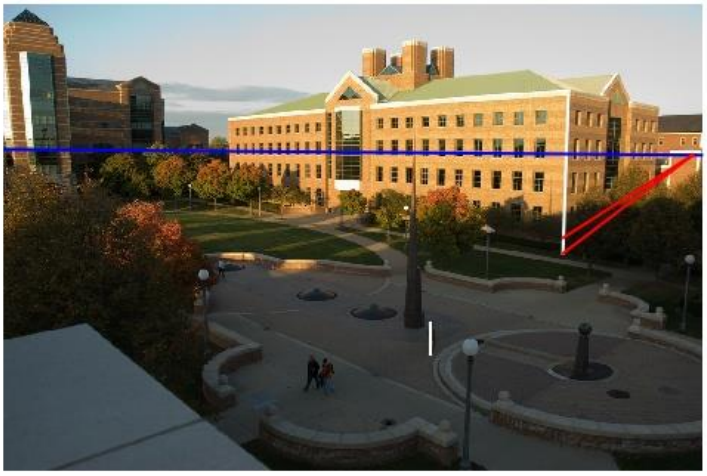
Parameters: 8.25849181e-03 -9.99965898e-01 2.11815410e+02

The focal length is 746.6196753948349
The optical center is 559.6766231525515 287.5566684306717
The rotation matrix R is

[[-0.66410722 0.00822129 -0.74759214]
[0.06545062 -0.99546118 -0.06908874]
[-0.74476696 -0.0948127 0.66055487]]

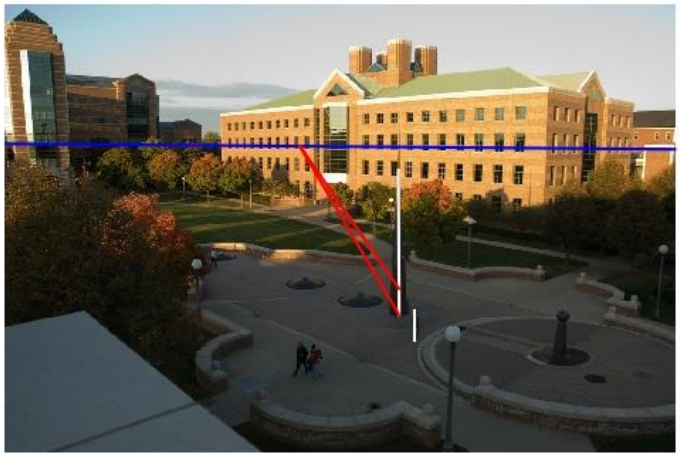
For the last part, target object and reference object are shown in white line, the li nes used to get point t are shown in red, horizon line is shown in blue.

The measurement process for CSL is



16.759m when reference is 5ft 6in
18.283m when reference is 6ft

The measurement process for spike statue is

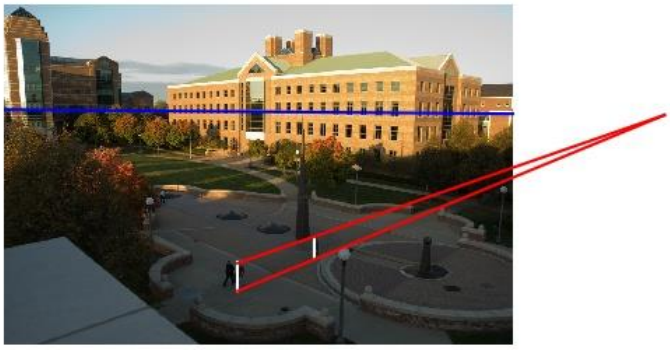


9.242m when reference is 5ft 6in
10.082m when reference is 6ft

The measurement process for lamp is

4.726m when reference is 5ft 6in
5.156m when reference is 6ft

The measurement process for the tallest man in the image is



1.715m when reference is 5ft 6in
1.871m when reference is 6ft

Extra points:

Part 1

I experiment with multiple images. I also build a relatively more complete system that can take as much input as you give and output the panorama result.

Part 2

I use my RANSAC code with ground-truth matches.

Part 3

I performed an measurement on the tallest man in the image.