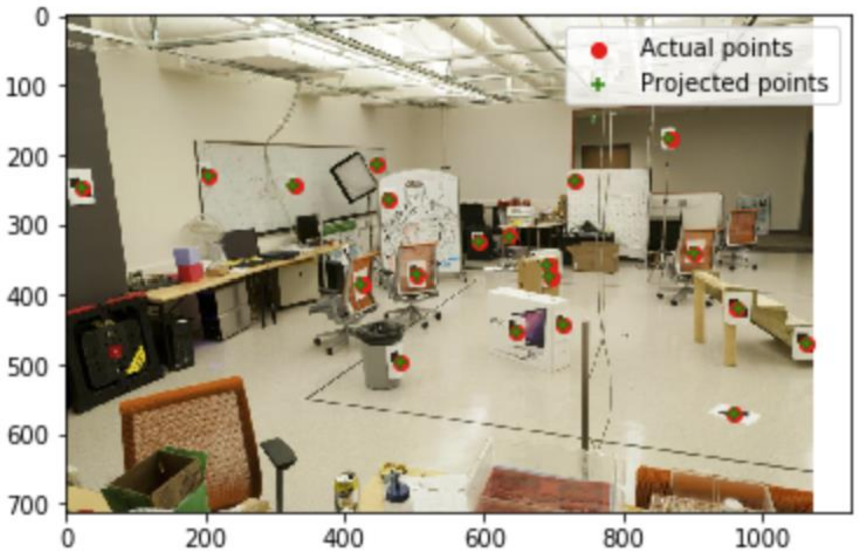
CS 4476/7646 Project 2

Wenyue Wang
wenyue_wang@gatech.edu
903204153

Part 1: Projection Matrix

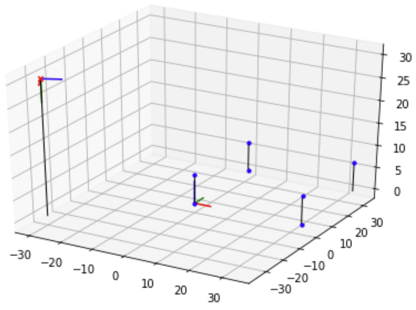
<insert visualization of projected 3D points and actual 2D points for image provided by us here>

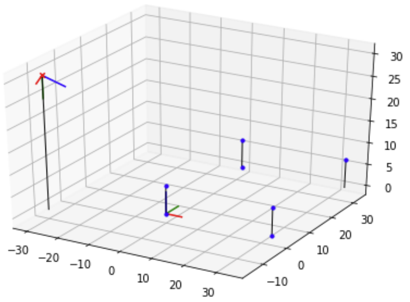


< insert the two images of your fiducial object here>

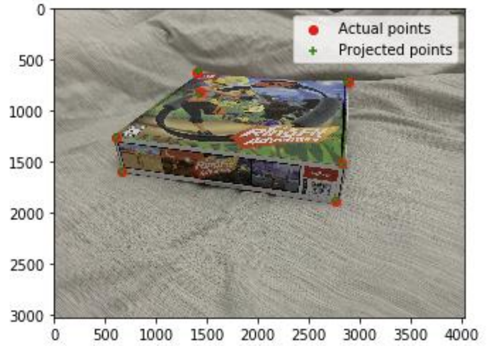
Part 1: Projection Matrix

<insert visualization the initial guesses for rotation matrix and camera center for the two images here>



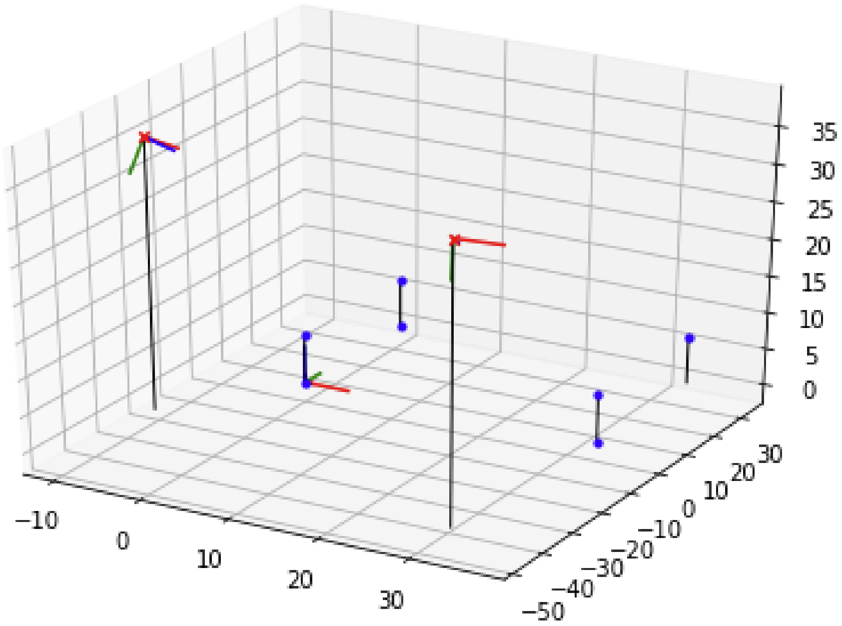


<insert visualization of projected 3D points and actual 2D points for both the images you took>



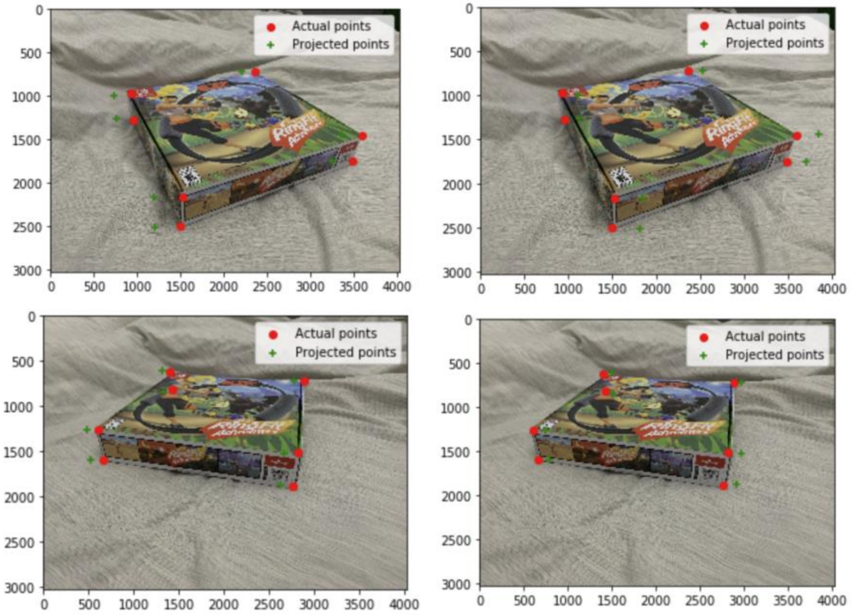
Part 1: Projection Matrix

<insert visualization of both camera poses here>

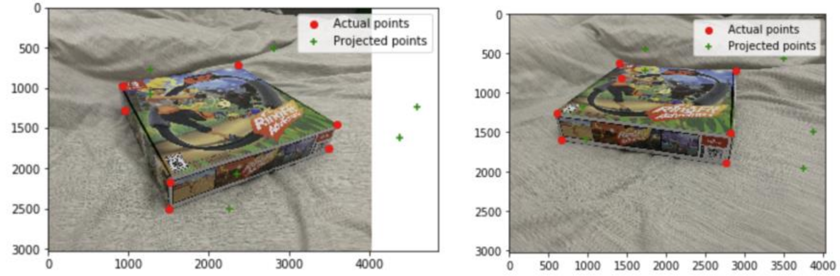


Part 1: Projection Matrix

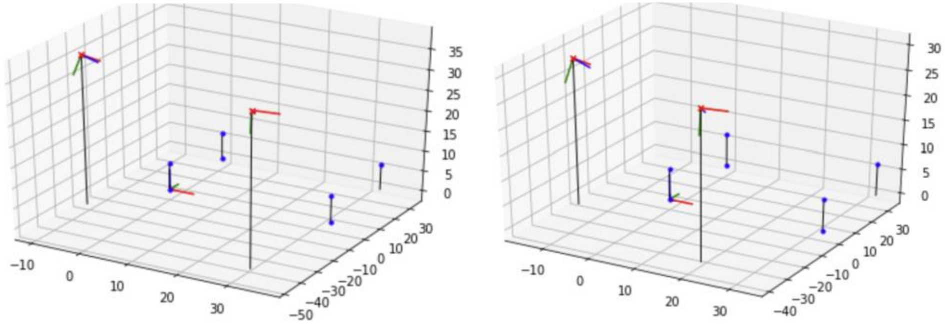
Before performing the projection, I changed t_x, t_y and t_z from P1 and P2 separately. By shrinking it by 0.8 I expected to see the projected points shift to the origin a little on each axis. And by enlarging it by 1.2 I expected to see them shift away from the origin. Below is what the projection like after I multiple x axis by 0.8 and 1.2, respectively. And it does show what I expected.



For question 2, I enlarged the x axis, and shrink the y and z axes. The finalized results for projection points, and camera centers before and after changing P is as follows. As we can see, the position of camera center is following the same pattern as we change the axes.

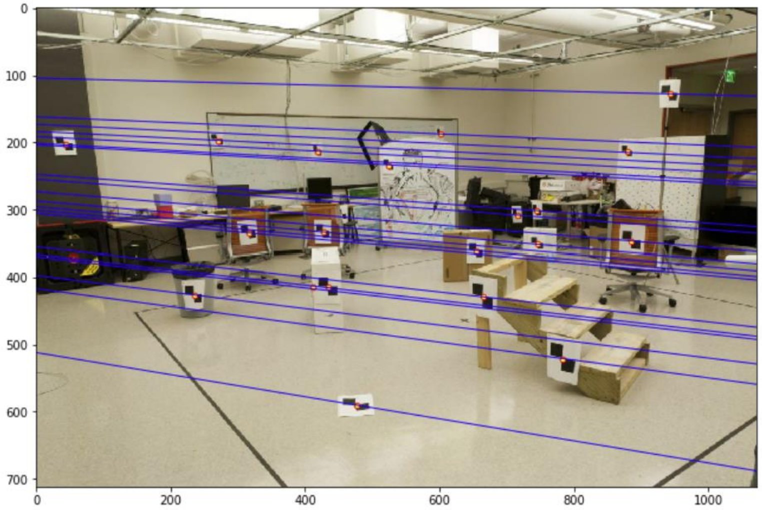



```
[ -9.11646544 -23.51398307  36.5307669 ]     [-13.98462423 -15.82208366  29.30502063]
[ 31.94304646 -45.07522369  37.41670315]     [ 21.48037636 -36.90156948  29.89356297]
```

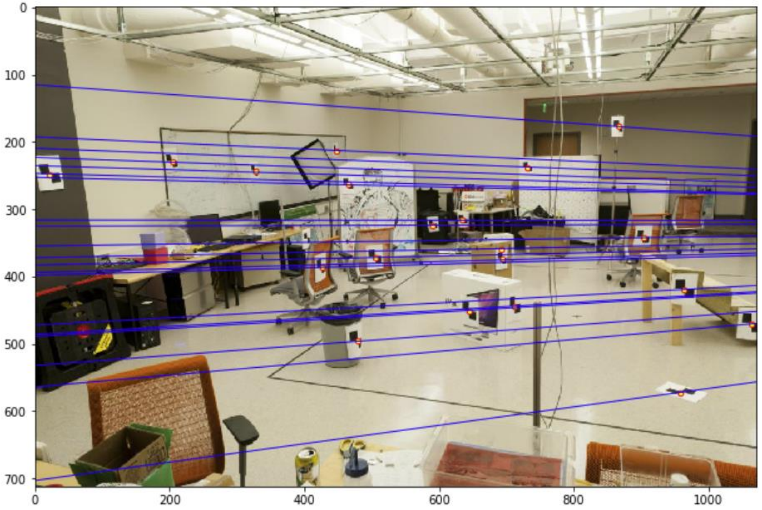


Part 2: Fundamental Matrix Estimation

Room: Left Image with Epipolar Lines



Room: Right Image with Epipolar Lines



Part 2: Fundamental Matrix Estimation

Fundamental Matrix Estimation Result:

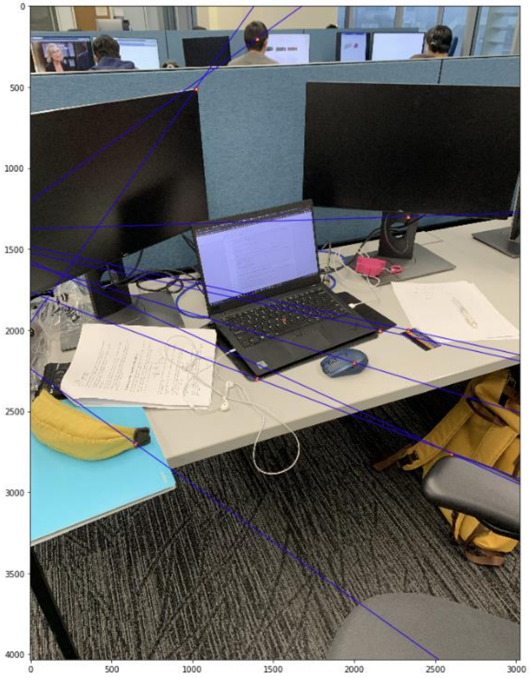
[[-0.00005608 0.00059327 -0.014601]

[0.00087223 -0.00015445 -2.39641427]

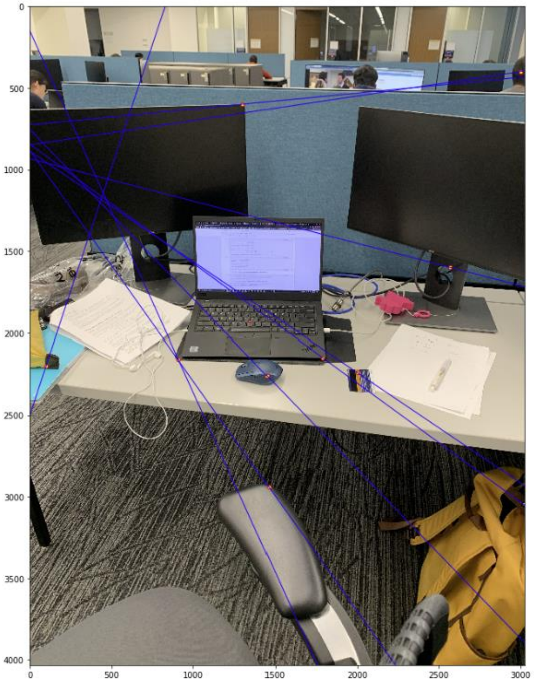
[-0.22260075 1.74869963 57.12943309]]

Part 2: Fundamental Matrix Estimation: Your Images

Your Image: Left Image with Epipolar Lines



Your Image: Right Image with Epipolar Lines



Part 2: Fundamental Matrix Estimation: Your Image

Fundamental Matrix Estimation Result:

[[-0.00000181 0.00001281 -0.01698484]

[-0.00001079 -0.00001262 0.01335842]

[0.00985009 0.01902202 -8.12341478]]

Part 2: Reflection Questions

1. Because the focal length of two images will be different

2. Because epipolar line is the line between epipolar and the projection of the point from the image

3. This will be similar as the forward motion case, where epipolar is the locus of the principal point of the camera, and epipolar lines go out from the principal point.

4. It means when taking these two photos, camera does not rotate but pure translation, so that two images are on the same plane

5. Because F is not unique, it varies based on camera and image positions

6. Because F is a 3*3 matrix, and 7 degrees of freedom. So it is singular with a rank of 2.

Part 2: Extra Credit: Fundamental Matrix Song

Reflect on the Fundamental Matrix Song

Link here: https://www.youtube.com/watch?v=DgGV3l82NTk

Fundamental matrix song uses a humor and interesting way to clearly explain fundamental matrix and its relevant concepts, like epipolar constraints, homogeneous design, SVD decomposition and rank deprivation. It helps me to connect all subsections of epipolar lines, and the hand-drawing style helps visualized how images formed.

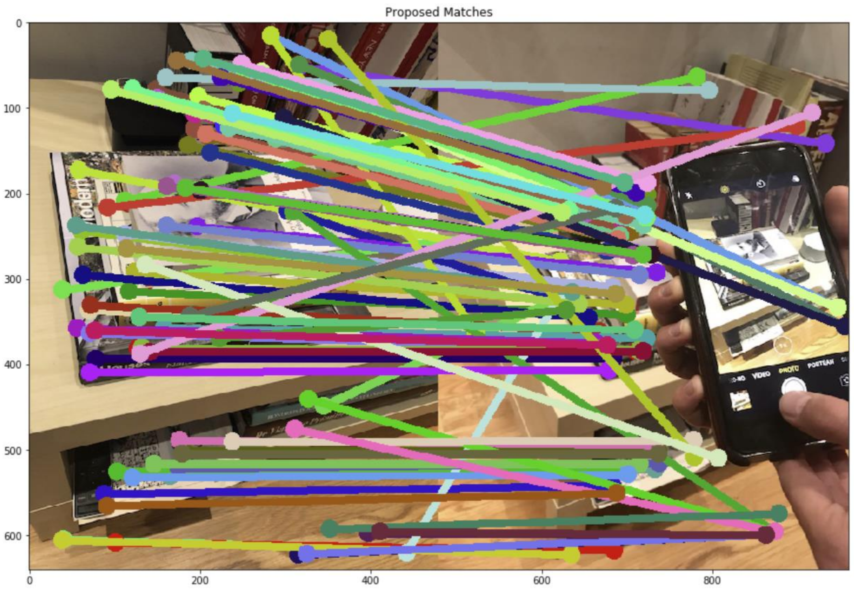
Part 3: RANSAC Iterations Questions

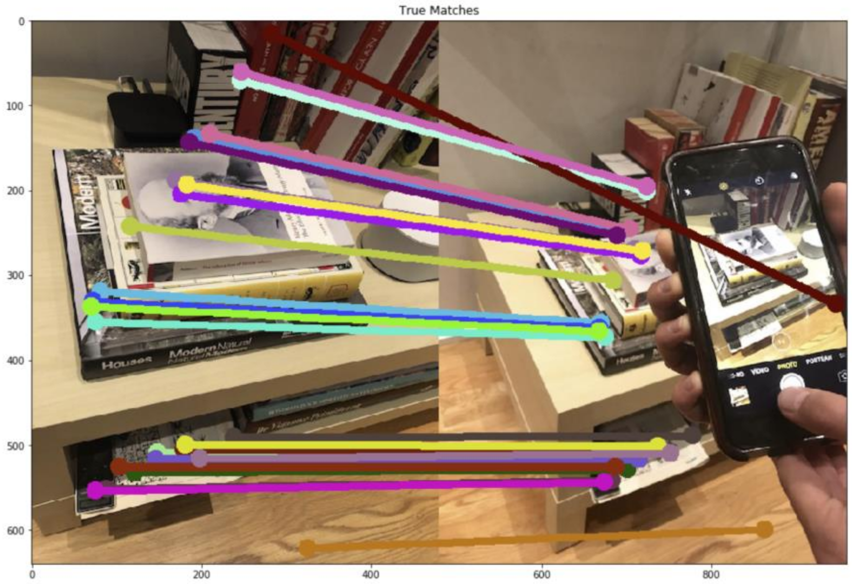
Delete the questions and type your answers to the three RANSAC Iterations questions from the jupyter notebook below:

1. 14 iterations
2. 42 iterations.
3. 167 iterations

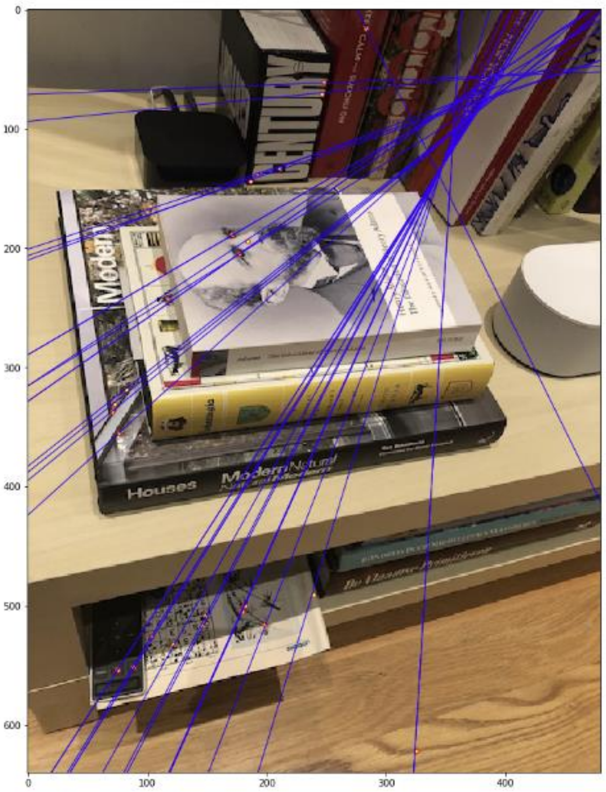
Part 3: RANSAC Inlier Matches

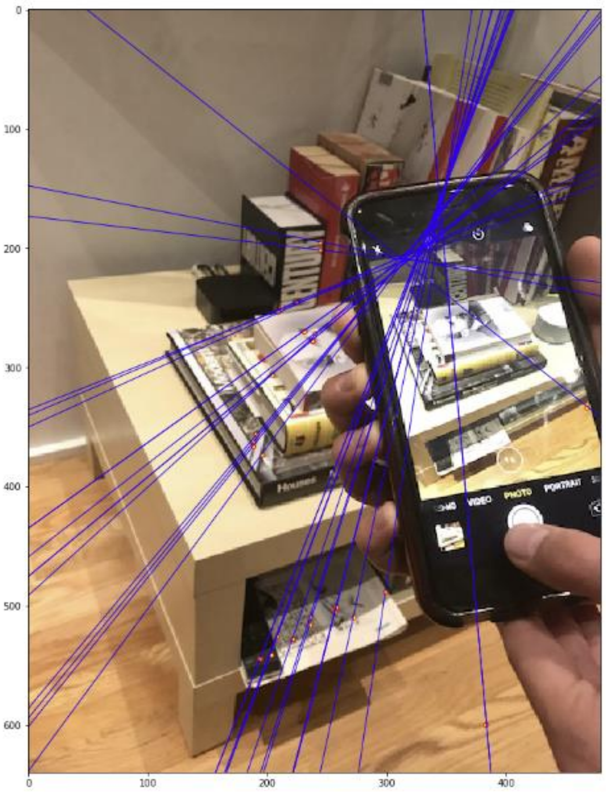
Paste two sets of images: displaying matches loaded from disk; and inlier matches after RANSAC





Part 3: RANSAC Implementation





Tests

```
part1_unit_test.py .....                                                                                       [ 25%]
test_essential_matrix_decomposition.py .F                                                                      [ 35%]
test_fundamental_matrix.py .........                                                                           [ 80%]
test_ransac.py ....                                                                                            [100%]

====================================================== FAILURES ======================================================
________________________ TestEssentialMatrixDecomposition.test_recover_rot_translation_from_E ________________________

self = <proj2_unit_tests.test_essential_matrix_decomposition.TestEssentialMatrixDecomposition testMethod=test_recover_ro
t_translation_from_E>

    def test_recover_rot_translation_from_E(self):
        R1, R2, t = recover_rot_translation.recover_rot_translation_from_E(self.I)
        assert np.allclose([0., 0., 1.57079633], R1) or np.allclose([0., 0., 1.57079633], R2)
        assert np.allclose([-0., -0., -1.57079633], R1) or np.allclose([-0., -0., -1.57079633], R2)
        assert np.array_equal([0., 0., 1.], t)

        R1, R2, t = recover_rot_translation.recover_rot_translation_from_E(self.E)
        assert np.allclose([ 1.42572617, -1.69270725, -2.19161805], R1) or np.allclose([ 1.42572617, -1.69270725, -2.191
61805], R2)
        assert np.allclose([-2.13337135,  0.89636235, -2.07923215], R1) or np.allclose([-2.13337135,  0.89636235, -2.079
23215], R2)
>       assert np.allclose([ 0.54960898, 0.80051726, -0.23896042], t)
E       AssertionError: assert False
E        +  where False = <function allclose at 0x0000017E7C841EE8>([0.54960898, 0.80051726, -0.23896042], array([-0.549
60898, -0.80051726,  0.23896042]))
E        +    where <function allclose at 0x0000017E7C841EE8> = np.allclose

test_essential_matrix_decomposition.py:37: AssertionError
------------------------------------------------ Captured stdout call ------------------------------------------------
[1. 1. 1.]
[4.60956962e+07 2.24592624e+05 2.31822386e-01]
================================================== warnings summary ==================================================
proj2_unit_tests/test_ransac.py::test_ransac_find_inliers
  C:\Users\Wenyue Wang\Documents\Spring 2020\CS 6476\HW\HW2\proj2_unit_tests\test_ransac.py:46: DeprecationWarning: elem
entwise comparison failed; this will raise an error in the future.
    assert outliers not in inliers

-- Docs: https://docs.pytest.org/en/latest/warnings.html
====================================== 1 failed, 19 passed, 1 warnings in 5.50s ======================================

C:\Users\Wenyue Wang\Documents\Spring 2020\CS 6476\HW\HW2\proj2_unit_tests>
```

Conclusions

From this project, I understand how epipolar geometry works and how RANSAC be applied. The visualization on the given example, and the rying out based on our own images extremely helps me to understand these concepts. The overall difficulty is reasonable, since Piazza do provides lots of hints and explanations.