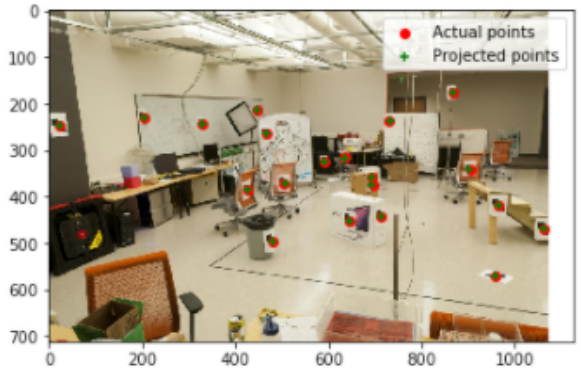
CS 6476 Project 4

Name: Shen-Yi Cheng
GT email: scheng98@gatech.edu
GTID: 903514405

Part 1: Projection Matrix

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

```
Time since optimization start 0.2518620491027832
The projection matrix is
 [[-2.04554508e+00  1.18126375e+00  4.05588148e-01  2.44822741e+02]
 [-4.55828586e-01 -3.04147973e-01  2.14988413e+00  1.66188164e+02]
 [-2.24222867e-03 -1.09957031e-03  5.71552114e-04  1.00000000e+00]]
The total residual is 14.711475
```



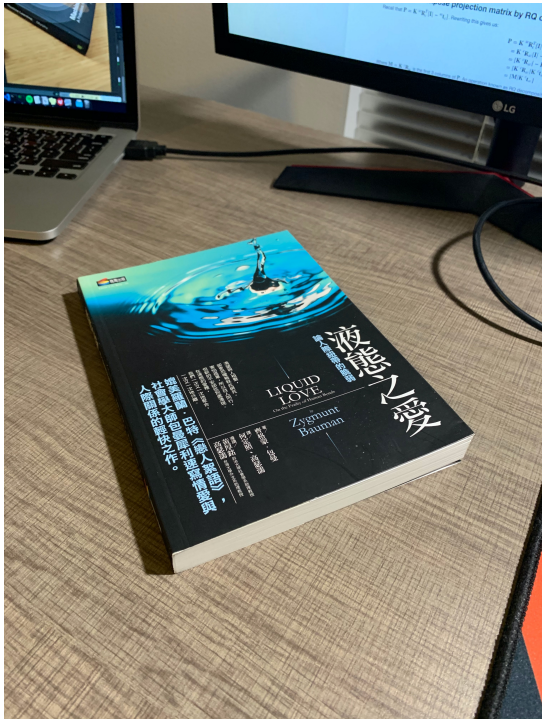
Part 1: Projection Matrix

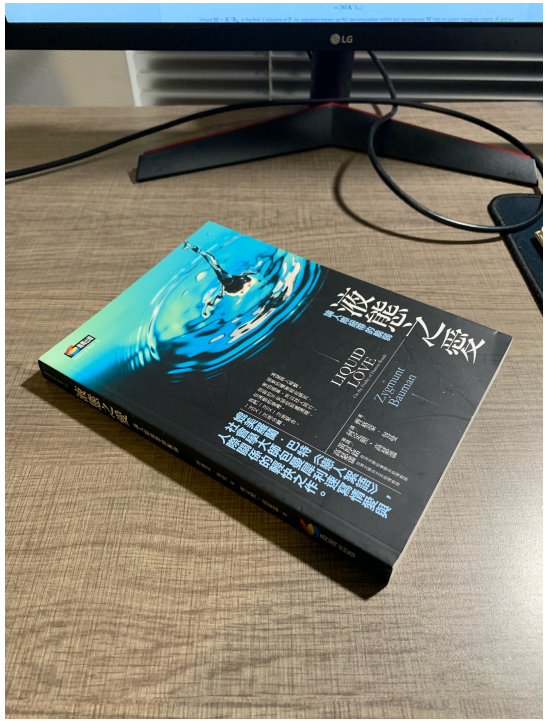
At least how many 3D-2D point correspondences do you need to estimate the projection matrix? Why?

At least 8 points for projection matrix. Because the least_square solution needs 8 points to do estimation.

Part 2: Estimation with Your Own Images

< insert the two images of your fiducial object here>





Part 2: Estimation with Your Own Images

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

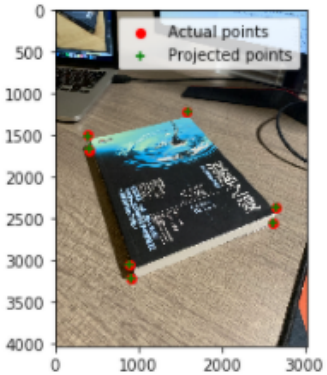
In [1553]:
```
initial_guess_K = np.array([[ 1000,   0, 2000],
                            [   0, 1000, 1500],
                            [   0,   0,  1]])

initial_guess_wRc_T = np.array([[ 0.5,   -1,  0],
                            [   0,    0, -1],
                            [   1,  0.5,  0]])

initial_guess_I_t = np.array([[   1,    0, 0, -30],
                              [   0,    1, 0, 20],
                              [   0,    0, 1, 30]])

initial_guess_P = np.matmul(initial_guess_K, np.matmul(initial_guess_wRc_T, initial_guess_I_t))
```

In [1558]:
```
initial_guess_K = np.array([[ 1000,   0, 2000],
                            [   0, 1000, 1500],
                            [   0,   0,  1]])

initial_guess_wRc_T = np.array([[ 0.5,   -1,  0],
                            [   0,    0, -1],
                            [   1,  0.5,  0]])

initial_guess_I_t = np.array([[   1,    0, 0, -30],
                              [   0,    1, 0, 30],
                              [   0,    0, 1, 10]])


initial_guess_P = np.matmul(initial_guess_K, np.matmul(initial_guess_wRc_T, initial_guess_I_t))
```

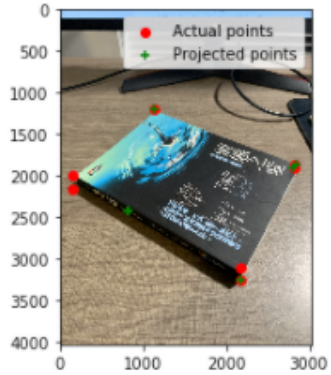
Part 2: Estimation with Your Own Images

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

```
Time since optimization start 0.03651714324951172
The projection matrix is
 [[-1.09389430e+01  9.99388252e+01 -2.92562509e+01  9.19998416e+02]
 [-2.45582815e+01 -1.33204464e+01 -1.57244064e+02  3.22589007e+03]
 [ 2.50066409e-02  7.36196063e-03 -2.17943005e-02  1.00000000e+00]]
The total residual is 52.039234
```

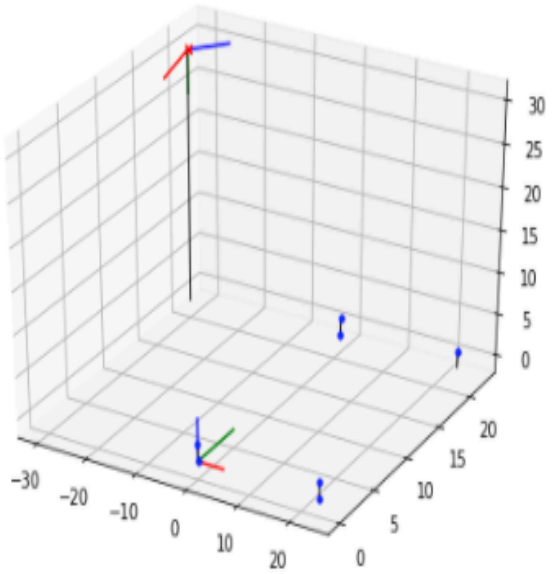


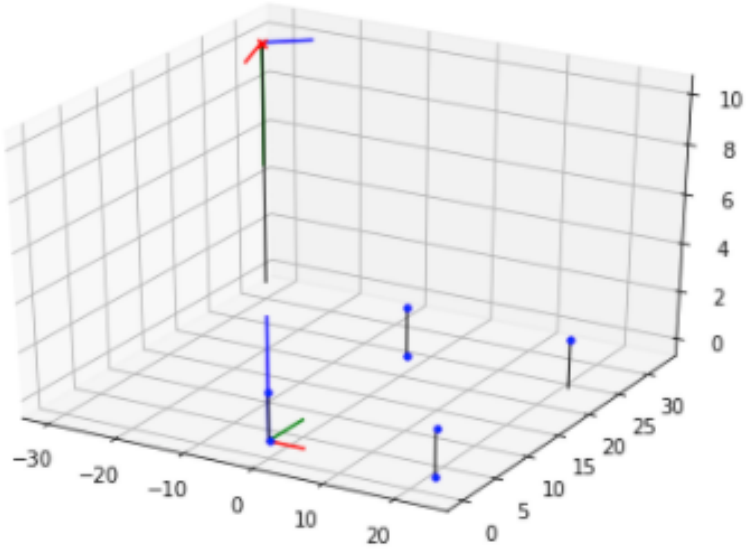

```
Time since optimization start 4.772753953933716
The projection matrix is
 [[ 1.81986569e+06  4.80937109e+03 -5.14365554e+04  2.17040565e+03]
 [ 5.32920374e+06  3.17499368e+03 -3.48122040e+04  3.25900984e+03]
 [ 2.19958140e+03  1.70423119e+00 -1.82654538e+01  1.00000000e+00]]
The total residual is 3098.154950
```



Part 2: Estimation with Your Own Images

<insert visualization of both camera poses of images you took here>





Part 2: Estimation with Your Own Images

- What would happen to the projected points if you increased/decreased the x coordinate, or the other coordinates of the camera center t? Write down a description of your expectations in the appropriate part of your writeup submission.

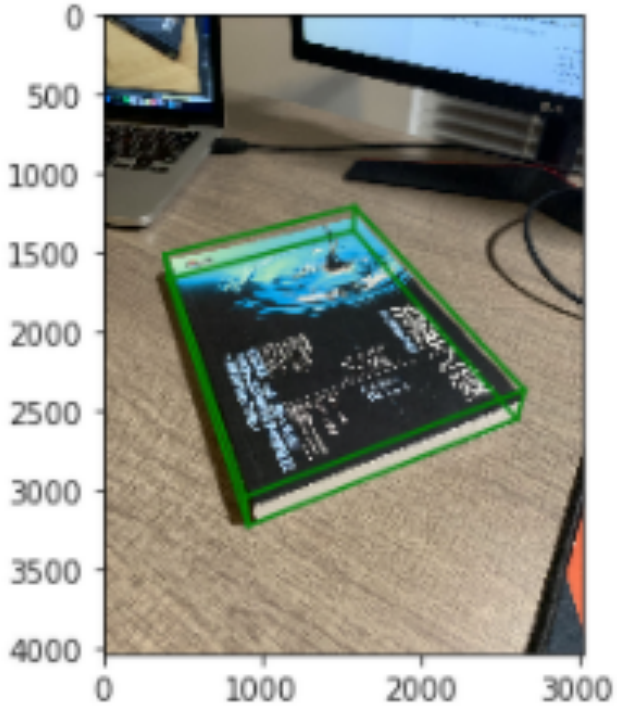
- The estimation will be shift if we increased/decreased the coordinate of the camera center. Changing x would shift up and down. Changing y would ship right and left.

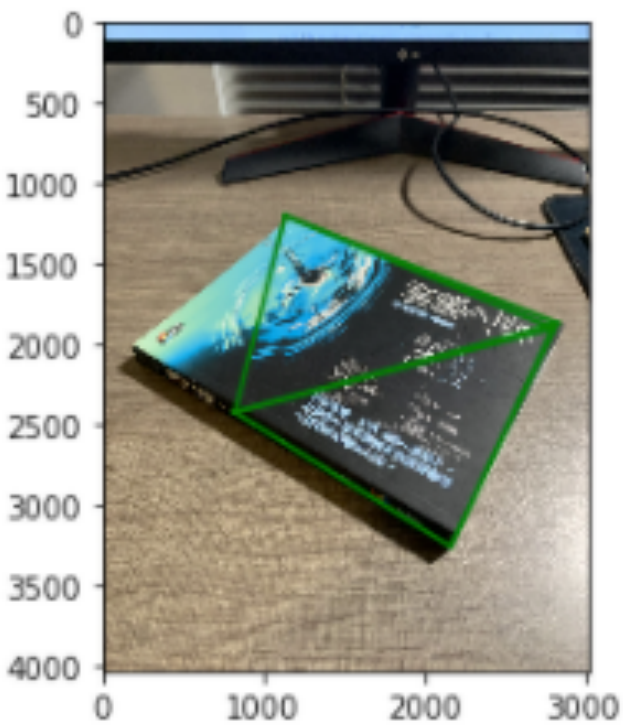
- Perform this shift for each of the camera coordinates and then recompose the projection matrix and visualize the result in the Jupyter notebook. Was the visualized result what you expected?

- Yes, the projection dots do shift followed by the changing of x or y.xw

Part 2: Estimation with Your Own Images

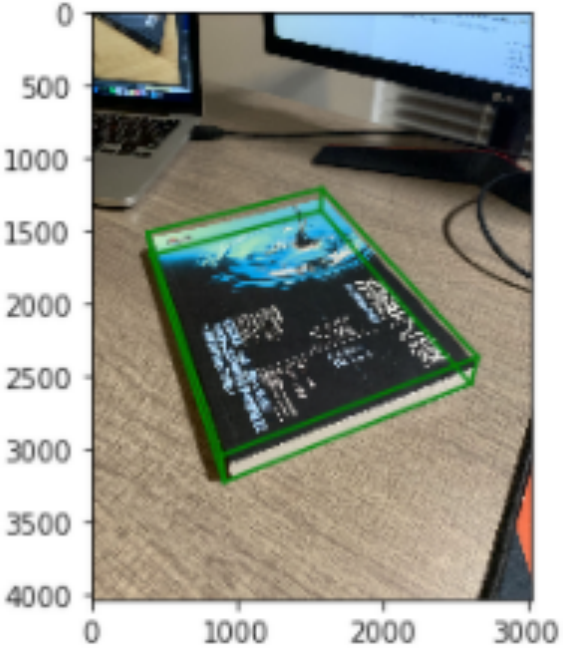
<insert both images you take with bounding boxes around your fiducial object>

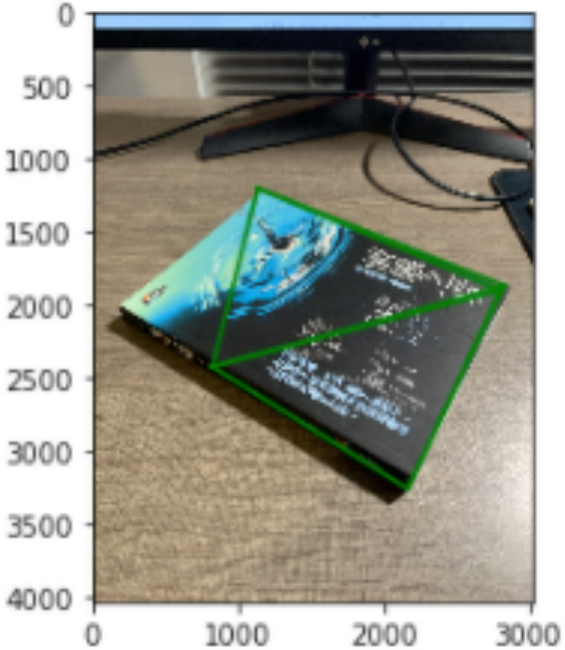




Part 3: Camera Coordinates

<insert both images you take with bounding boxes around your fiducial object>





Part 3: Camera Coordinates

The first three rows of the transformation matrix are just the extrinsic parameters of our camera, why is that?

Because the first three rows are directly from the wtc matrix.

Wtc matrix is the coordinate of the camera

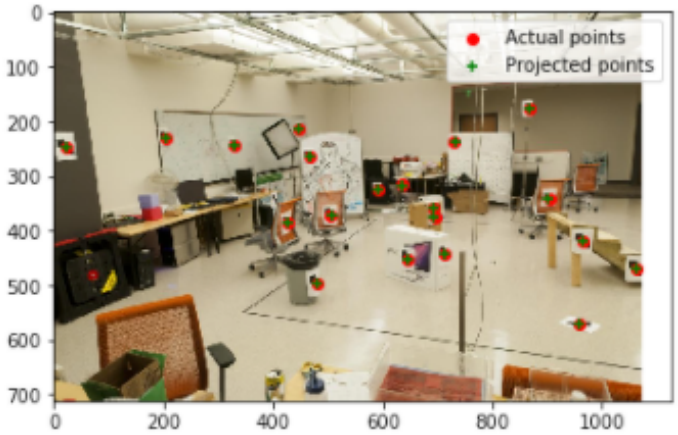
Say if we know the coordinates of 3D points in camera 1 coordinate system, and we want to transform them into camera 2 coordinate system, what should we do? You may assume that the poses of both cameras are known.

We can use the projection matrix of camera 1 to get the world coordinate of the 3D points and use the projection matrix of camera 2 to transform into camera 2 coordinate system

Part 4: DLT

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

```
The projection matrix is
 [[-2.04586455e+00  1.18558243e+00  3.91381081e-01  2.44002874e+02]
 [-4.56804042e-01 -3.02392053e-01  2.14706068e+00  1.66030240e+02]
 [-2.24595257e-03 -1.09488059e-03  5.61389950e-04  1.00000000e+00]]
The total residual is 15.544953
```



Part 4: DLT

Why do we need the cross product trick?

The homogeneous linear equation is:

0 = [x_k] x A * y_k

[x_k] is the matrix representation of the vector cross product

We pick up the eigenvector with the smallest eigenvalue. Will this eigenvalue be exactly zero? Why/why not?

The smallest eigenvalue would not be exactly zero in every circumstance. It based on the singular value decomposition result. However , each column of V represents a solution of Ah = 0. Moreover, SVD does not deal well with outliers.

EC: RANSAC Iterations Questions

How many RANSAC iterations would we need to find the fundamental matrix with 99.9% certainty from your Mount Rushmore and Notre Dame SIFTNet results assuming that they had a 90% point correspondence accuracy?

Iteration: 15

One might imagine that if we had more than 9 point correspondences, it would be better to use more of them to solve for the fundamental matrix. Investigate this by finding the number of RANSAC iterations you would need to run with 18 points.

Iteration: 43

If our dataset had a lower point correspondence accuracy, say 70%, what is the minimum number of iterations needed to find the fundamental matrix with 99.9% certainty?

Iteration: 168

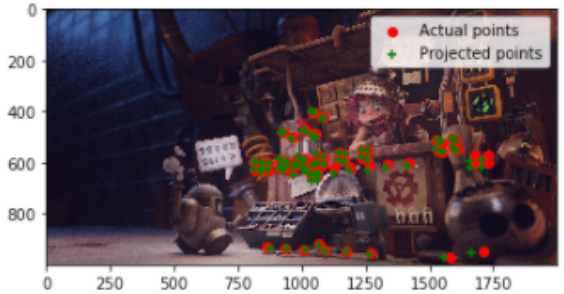
EC: RANSAC

<insert the number of inliers from your best model here>

```
In [1680]: P, _, _ = ransac_projection_matrix(corrupted_points_2d, corrupted_points_3d, 8.0, num_iterations)
Time since optimization start 0.8285520076751709
Time since optimization start 0.012006759643554688
Time since optimization start 0.010755062103271484
Time since optimization start 0.009585857391357422
Time since optimization start 0.008975982666015625
Time since optimization start 0.025533199310302734
Time since optimization start 0.024047136306762695
Time since optimization start 0.0876009464263916
Time since optimization start 0.103051900863647 46
Time since optimization start 0.01370096206665039
Time since optimization start 0.012523889541625977
Time since optimization start 0.07919001579284668
Time since optimization start 0.07579588890075684
Time since optimization start 0.009465932846069336
Time since optimization start 0.013101816177368164
Time since optimization start 0.0437321662902832
Time since optimization start 0.043591976165771484
Time since optimization start 0.007610797882080078
Time since optimization start 0.006042003631591797
Found projection matrix with support  55
```

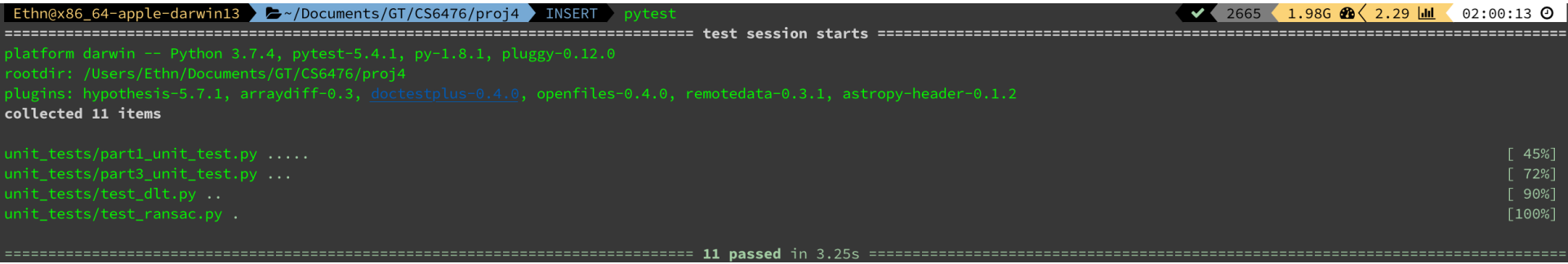
<insert the image and total residual here>

```
The projection matrix is
 [[ 6.76514918e-01  2.04469948e+00 -8.63529118e-03  1.27232943e+03]
 [-5.31230475e-01  8.62301708e-01 -1.73700523e+00  9.43435334e+02]
 [-6.37086858e-04  9.73965068e-04  1.27069061e-05  1.00000000e+00]]
The total residual is 680.084275
```



Tests

<Provide a screenshot of the results when you run `pytest` from the unit tests directory with your final code implementation (note: we will re-run these tests).>



Conclusions

<Describe what you have learned in this project. Feel free to include any challenges you ran into.>

It is difficult to find a perfect initial estimation

To many algebra in this project. I spend a lot of time to read and understand the concept.