Problem Set 4: Geometry

Questions

1. **Calibration**

a. **Output**:

- The matrix M you recovered from the normalized points (3x4) [text response]

```
[[ 0.76785834 -0.49384797 -0.02339781  0.00674445]
 [-0.0852134  -0.09146818 -0.90652332 -0.08775678]
 [ 0.18265016  0.29882917 -0.07419242  1.        ]]
```

- The < u, v > projection of the last point given your M matrix [text response]

[0.14190586 -0.45183985]

- The residual between that projected location and the actual one given [text response]

0.00156368840848

b.

- Average residual for each trial of each k (10x3) [text response]

8	2.641	1.369	6.422	2.399	2.422	1.968	1.180	2.044	0.721	4.966
12	4.476	0.963	1.548	1.485	1.019	0.759	4.397	1.009	1.884	1.616
16	4.003	5.272	0.898	0.979	1.643	1.241	1.355	1.492	2.156	1.448

- Explain any difference you see between the results for the different k's [text response]

A k of 8 definitely had the lowest residual. On average, if you increase the number of points used to calculate the projection, you should get a lower error rate. This makes sense as you're able to fit the data better with more data points in the least squares calculation. Since it's an approximation, there is some noise with randomly chosen points, but overall the 16 points seemed to perform the best.

- The best M matrix (3x4) [text response]

```
[[ -2.03985815e+00   1.16221907e+00   4.57774310e-01   2.47459066e+02]
 [ -4.48606137e-01  -3.07151802e-01   2.13500823e+00   1.65341090e+02]
 [ -2.23419469e-03  -1.11268879e-03   6.27272999e-04   1.00000000e+00]]
```

c. Finally we can solve for the camera center in the world. Let us define M as being made up of a 3x3 matrix that we'll call Q and a 4th column we'll call m_4:

Output:
- The location of the camera in real 3D world coordinates [text response]

[[303.15728018], [307.171646], [30.44740071]]

2. Fundamental Matrix Estimation

a.

Function: `compute_fundamental_matrix(pts2d_a, pts2d_b) -> F`

Output:
- The matrix $\tilde{F}$ generated from your Least Squares function [text response]

```
[[ -6.60675944e-07   7.90642197e-06  -1.88480992e-03]
 [  8.82674944e-06   1.21863596e-06   1.72276843e-02]
 [ -9.08539064e-04  -2.64201801e-02   1.00000000e+00]]
```

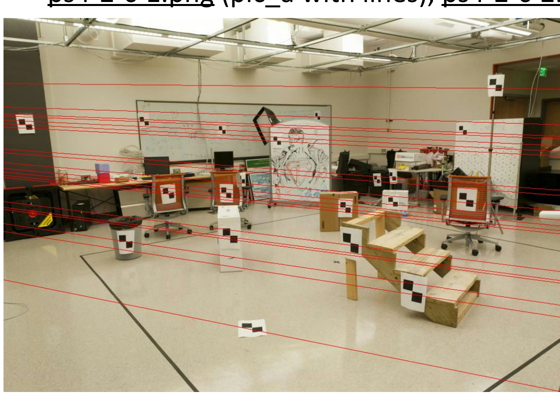
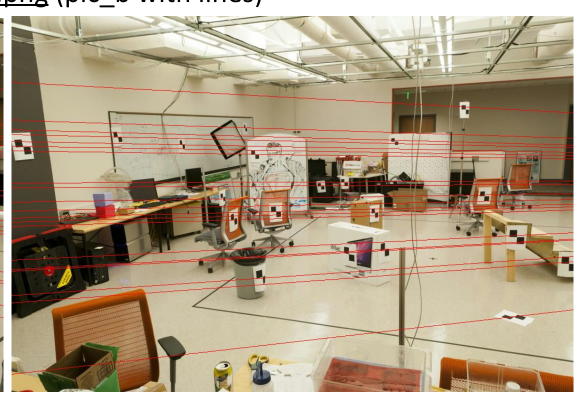
b.

Output:
- Fundamental matrix F [text response]

```
[[ -5.35883058e-07   7.89972529e-06  -1.88480998e-03]
 [  8.83820595e-06   1.21802118e-06   1.72276843e-02]
 [ -9.08539027e-04  -2.64201801e-02   1.00000000e+00]]
```

c. Now you can use your matrix F to estimate an epipolar line l_b in Image 'B' corresponding to point p_a in Image 'A': $l_b = F p_a$

Output:
- Images with the estimated epipolar lines drawn on them (to be included in report as well): ps4-2-c-1.png (pic_a with lines), ps4-2-c-2.png (pic_b with lines)

ps4-2-c-1.png ps4-2-c-2.png

EXTRA CREDIT PROBLEMS (2-d and 2-e)

d.

Output:

- The matrices T_a, T_b and $\widehat{F}$ [text response]

T_a = [[0.00422627 0. -2.36227228]
[0. 0.0085811 -2.79400691]
[0. 0. 1.]]

T_b = [[0.00385727 0. -2.37877704]
[0. 0.00958482 -3.32497426]
[0. 0. 1.]]

$\widehat{F}$ = [[0.35084 -2.92141829 -0.7692956]
[-1.49580681 0.22455053 -22.87282182]
[-4.05894836 24.04079161 1.]]

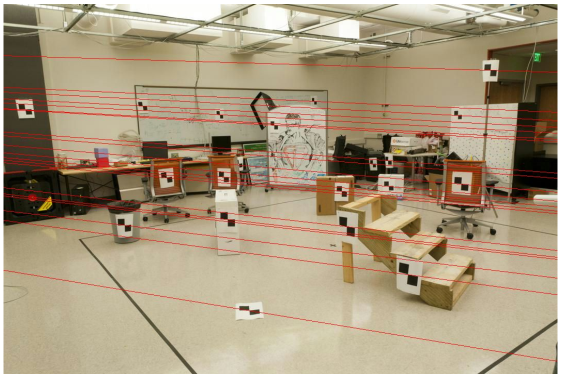
e.

Output:

- The new F [text response]

[[5.71933957e-06 -9.66978043e-05 2.53206011e-02]
[-6.05921672e-05 1.84689051e-05 -1.91377377e-01]
[3.38104151e-04 2.59523165e-01 -5.80819930e+00]]

- Images with the "better" epipolar lines drawn (to be included in report as well): ps4-2-e-1.png (pic_a), ps4-2-e-2.png (pic_b)



ps4-2-e-1.png

ps4-2-e-2.png