Jacob Fund
903138567 ps3

1
a.

M matrix: M =

0.4583	-0.2947	-0.0140	0.0040
-0.0509	-0.0546	-0.5411	-0.0524
0.1090	0.1783	-0.0443	0.5968

last u, v points: 0.1406 -0.4527

residual =

0.0130

b.

mean of k = 8 :

0.0169

mean of k = 12:

0.0092
mean of k = 16 :

0.0091

The results shows that after given a certain number of points, the linear transformation of M * [X Y Z 1] converges to a minimum after a certain amount of K. Although not shown on this particular run, sometimes, since my implementation is different every time it runs, sometimes when k =12 it performs better than k = 16. So to conclude, only a certain amount of points are really needed for an accurate representation, that is assuming that there is no noise within the measurements of the 3d points.

bestM =

0.4573	-0.2958	-0.0074	0.0036
-0.0510	-0.0543	-0.5429	-0.0522
0.1069	0.1775	-0.0365	0.5967

c.

Camera location =

-1.5402
-2.3758
0.2858

2.

a. F =

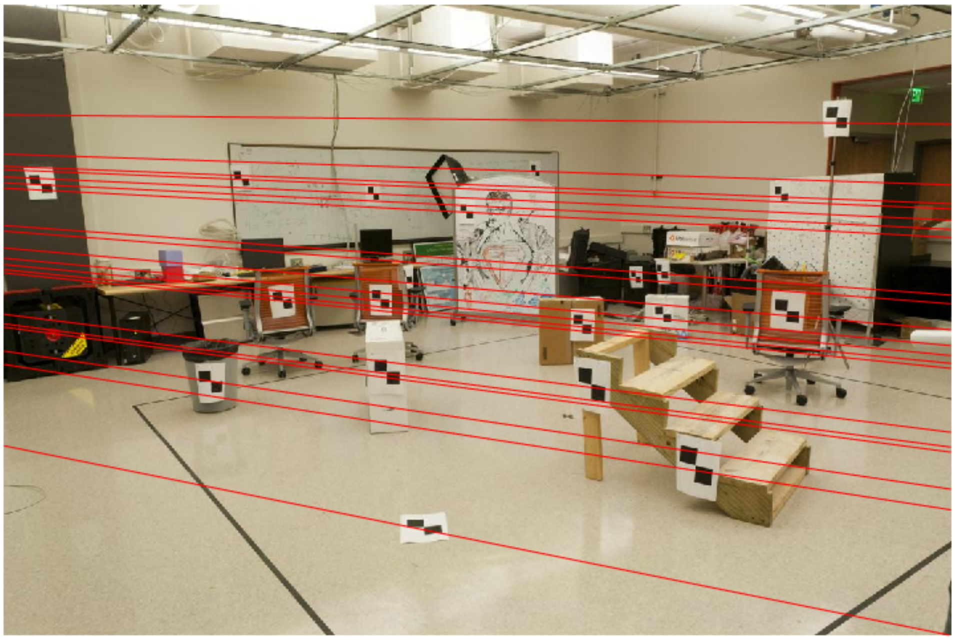
-6.60698417012622e-07	8.82396296136019e-06	-0.000907382302152603
7.91031620841439e-06	1.21382933021017e-06	-0.0264234649901806
-0.00188600197690852	0.0172332901072652	0.999500091906722

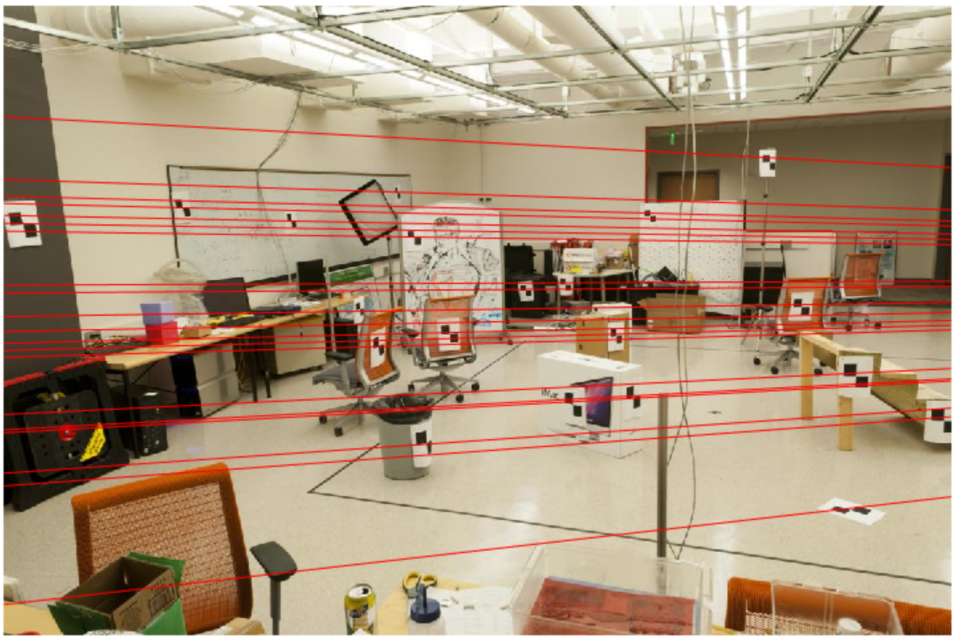
b. F_hat =

-5.36264198382353e-07	8.83539184115726e-06	-0.000907382264407744
7.90364770858056e-06	1.21321685010730e-06	-0.0264234649922034
-0.00188600204023565	0.0172332901014488	0.999500091906703

c.

Figure 2 ps3-2-c-1.png





Jacob Fund
903138567

ps3

2d.

ta =

0.0041	0	-2.5451
0	0.0041	-1.4316
0	0	1.0000

tb =

0.0045	0	-2.5088
0	0.0045	-1.4614
0	0	1.0000

F_hat =

-0.0052	0.0495	0.0532
0.0715	-0.0122	-0.6511
0.0112	0.7518	-0.0130

2e.

F =

-0.0000	0.0000	-0.0000
0.0000	-0.0000	-0.0037
-0.0003	0.0027	0.0853

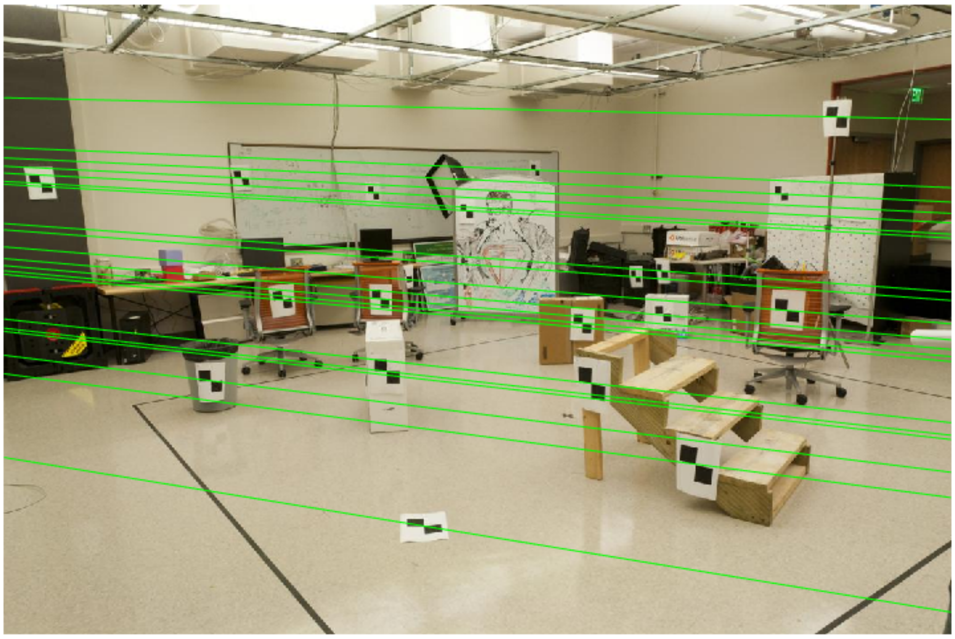


Figure 3 ps3-2-e-1.png

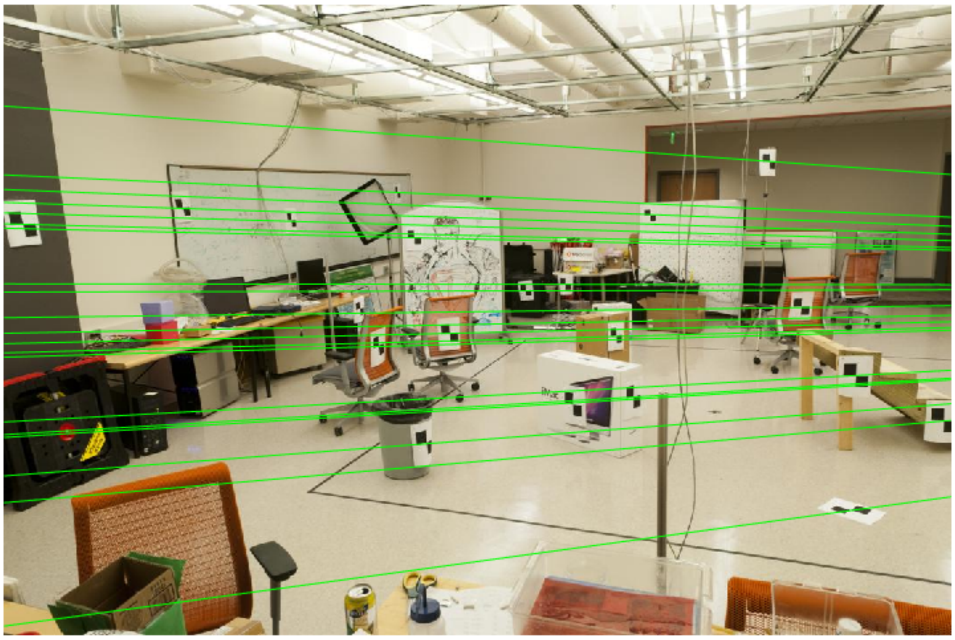


Figure 4 ps3-2-e-2.png