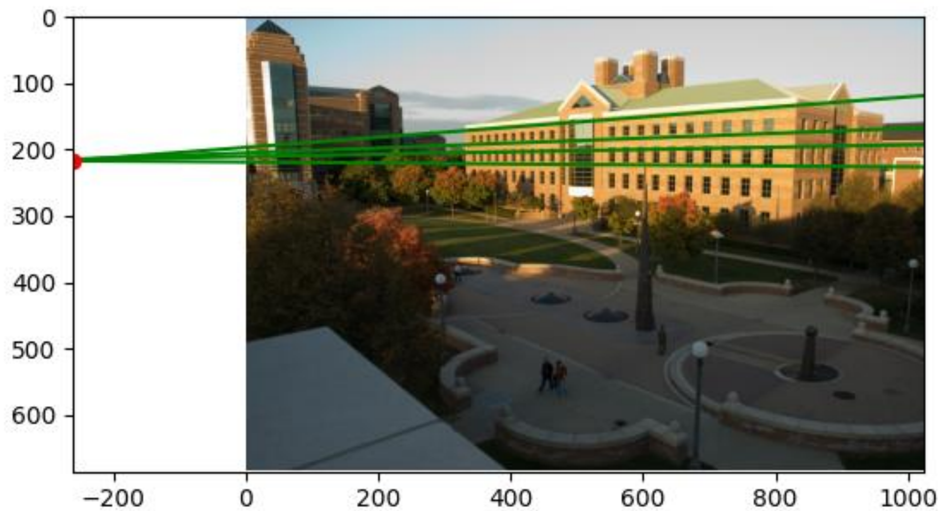
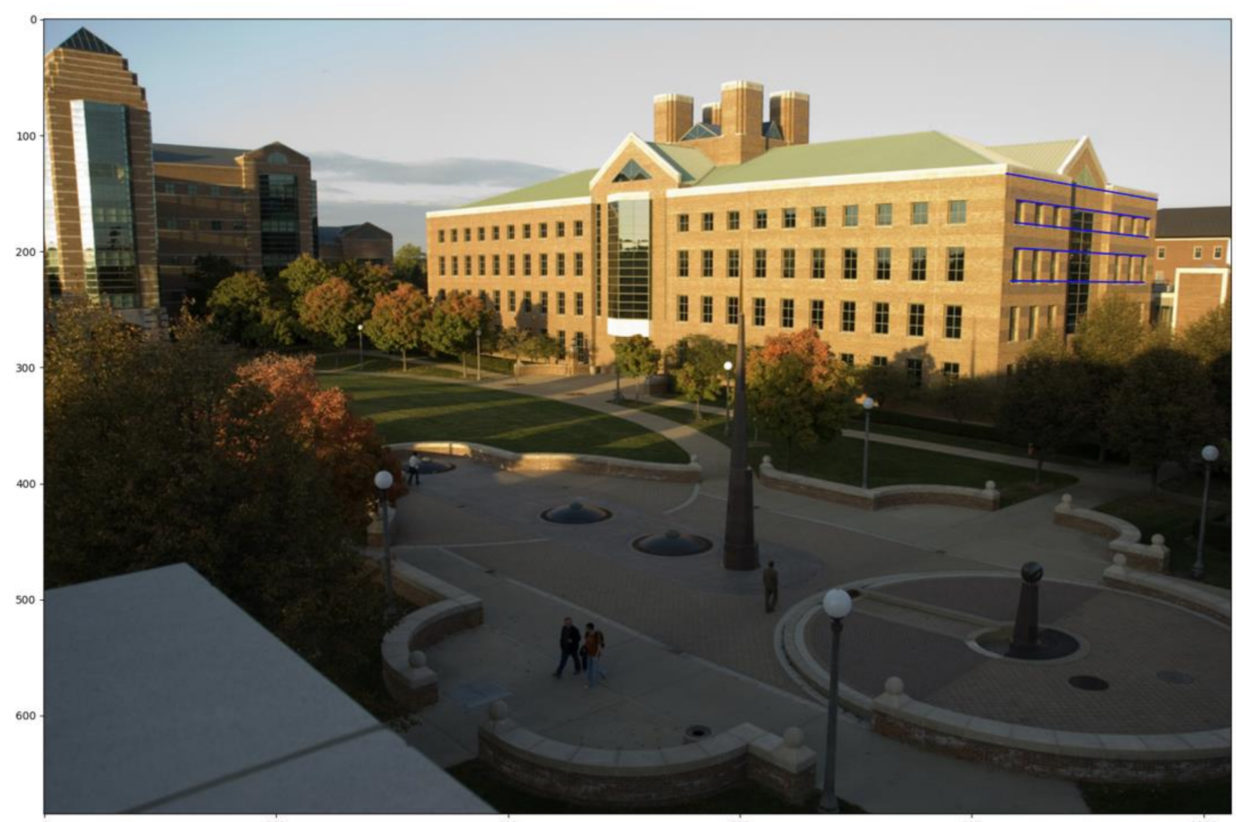
CS543 Assignment 4

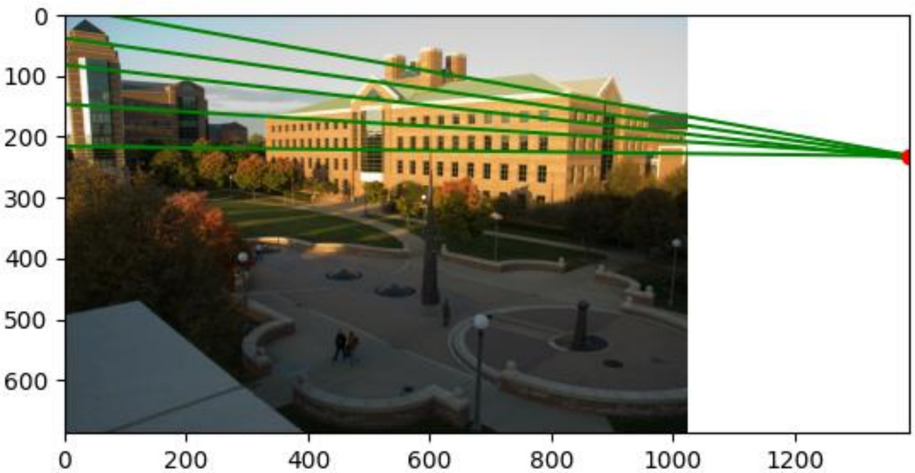
Your Name: Derek Yang
Your NetId: mcy3

Part 1 Single-View Geometry:

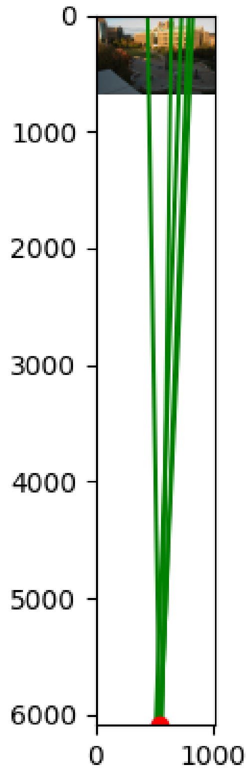
Plot the VPs and the lines used to estimate them on the image plane using the provided code.


0
100
200
300
400
500
600
−200
0
200
400
600
800
1000


0
100
200
300
400
500
600
0
200
400
600
800
1000


0
100
200
300
400
500
600
0
200
400
600
800
1000
1200


0
100
200
300
400
500
600
0
200
400
600
800
1000



Specify the VP pixel coordinates.

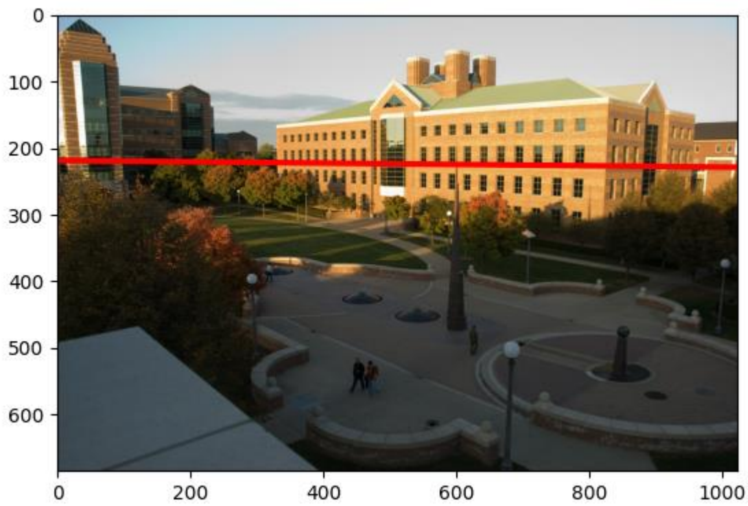
Every column is a vanish point.

```
print(vpts)
```

0.1s

```
[[-2.61161016e+02  1.38801135e+03  5.48998141e+02]
 [ 2.15832893e+02  2.32198534e+02  6.08936971e+03]
 [ 1.00000000e+00  1.00000000e+00  1.00000000e+00]]
```

Plot the ground horizon line and specify its parameters in the form a * x + b * y + c = 0.



Normalize the parameters so that: a^2 + b^2 = 1.

```
Click the two endpoints, use the right key to undo,
[-9.92305900e-03  9.99950765e-01 -2.18413783e+02]
```

Using the fact that the vanishing directions are orthogonal, solve for the focal length and optical center (principal point) of the camera. Show all your work.

Using the equation" v_i^TK^(-T)K^(-1)v_j = 0 " with the three orthogonal vanishing point we can solve for the focal length and optical center.

```python
# v_i^TK^(-T)K^(-1)v_j = 0

vp1 = vpts[:, 0][:, np.newaxis]
vp2 = vpts[:, 1][:, np.newaxis]
vp3 = vpts[:, 2][:, np.newaxis]

f, px, py= symbols('f, px, py')

K_inv = Matrix([[1/f, 0, -px/f], [0, 1/f, -py/f], [0, 0, 1]])

a = vp1.T * K_inv.T * K_inv * vp2
b = vp1.T * K_inv.T * K_inv * vp3
c = vp2.T * K_inv.T * K_inv * vp3

f, px, py = solve([a[0], b[0], c[0]], (f, px, py))[0]

return abs(f), px, py
```

```python
f, u, v = get_camera_parameters(vpts)
print("focal length:",f)
print("optical center x:",u)
print("optical center y:",v)
```

3.3s

```
focal length: 815.307725390433
optical center x: 606.051717012304
optical center y: 340.057184711083
```

Compute the rotation matrix for the camera.

```python
    Z = vpts[:, 0][:, np.newaxis]  # left
    X = vpts[:, 1][:, np.newaxis]  # right
    Y = vpts[:, 2][:, np.newaxis]  # down

    K = np.array([[f, 0, u], [0, f, v], [0, 0, 1]]).astype(float)
    K_inv = np.linalg.inv(K)

    r1 = K_inv.dot(X)
    r2 = K_inv.dot(Y)
    r3 = K_inv.dot(Z)


    r1 = r1 / np.linalg.norm(r1)
    r2 = r2 / np.linalg.norm(r2)
    r3 = r3 / np.linalg.norm(r3)

    R = np.concatenate((r1, r2, r3), axis=1)

    return R
```

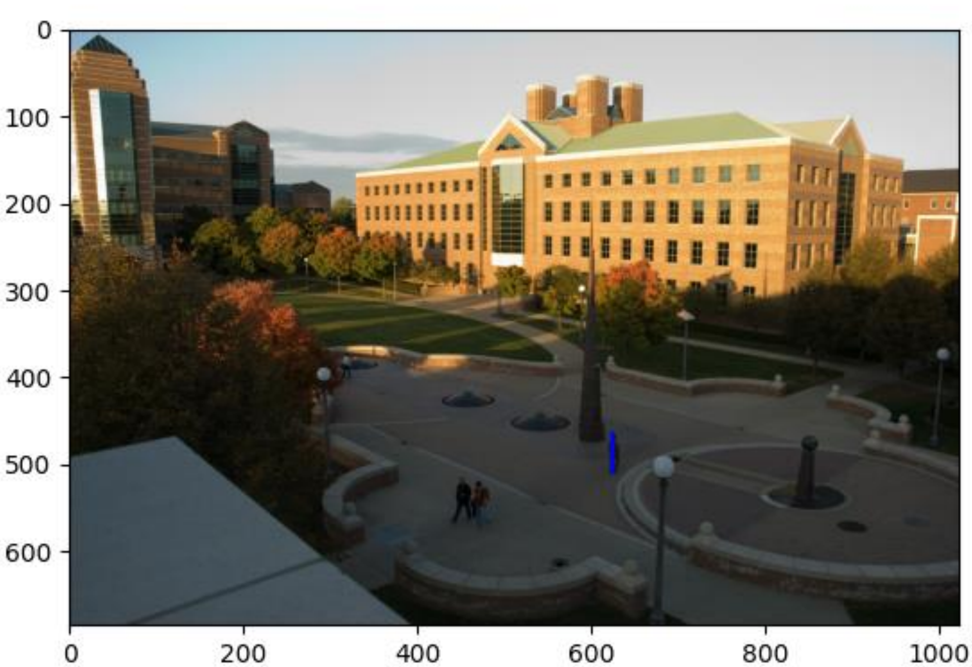
```python
R = get_rotation_matrix(vpts, f, u, v)
print("rotation matrix:")
print(R)
```

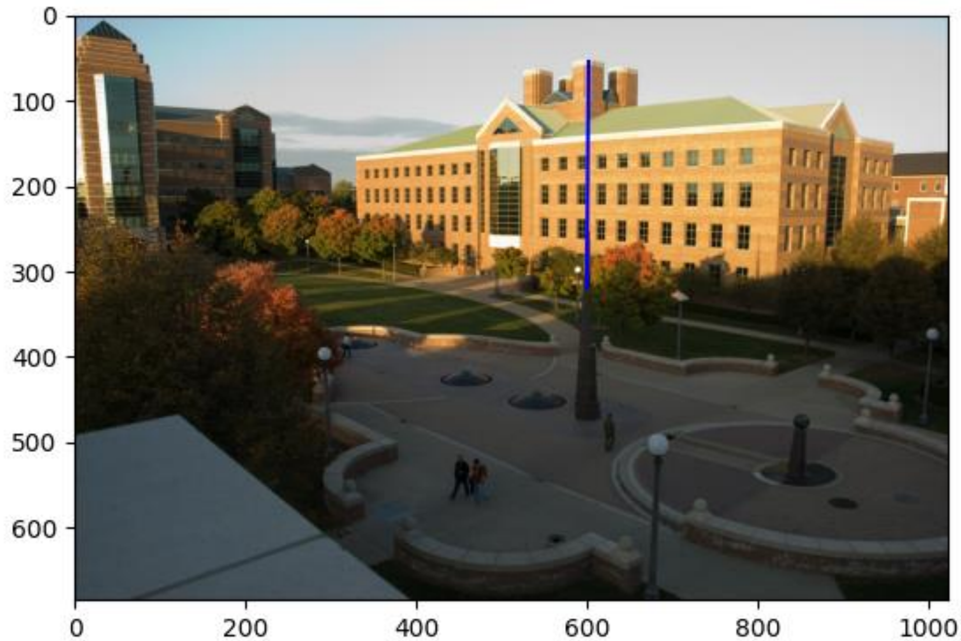
✓ 0.7s

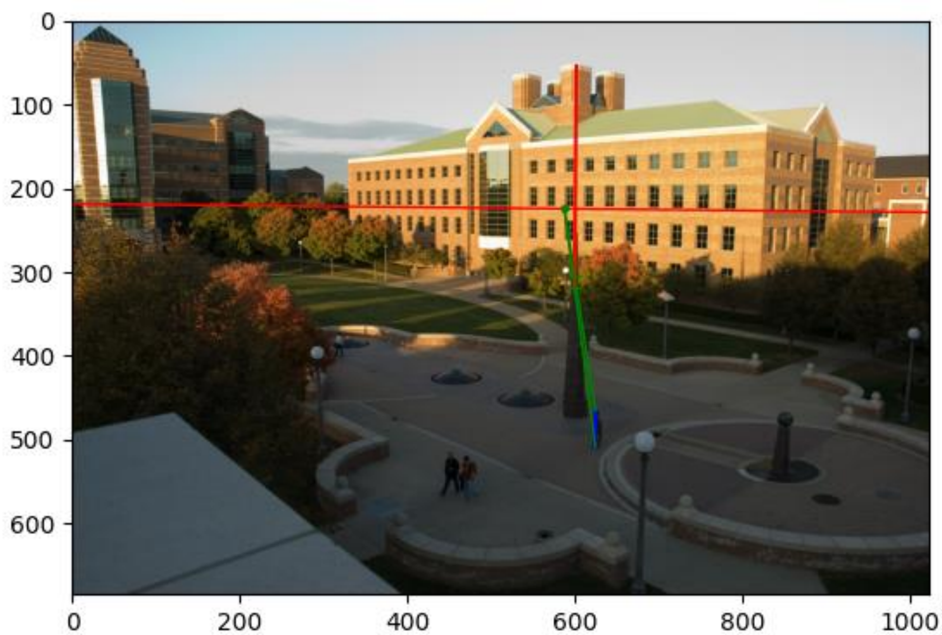
```
rotation matrix:
[[ 0.68905926 -0.00982477 -0.7246384 ]
 [-0.09504455  0.99004639 -0.10380116]
 [ 0.71844545  0.14039809  0.68126684]]
```

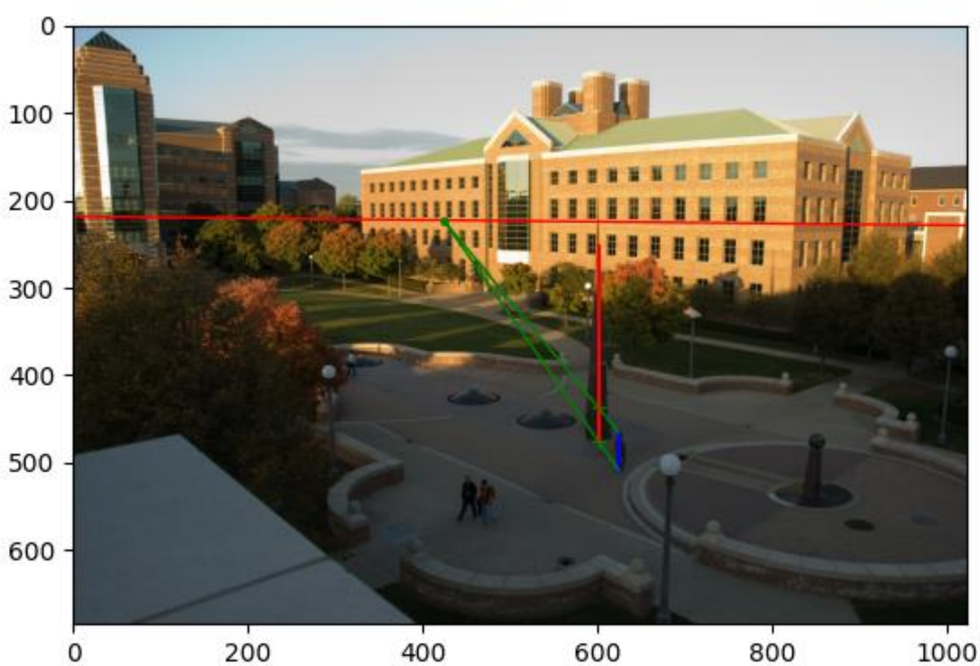
Estimate the heights of (a) the CSL building, (b) the spike statue, and (c) the lamp posts assuming that the person nearest to the spike is 5ft 6in tall. In the report, show all the lines and measurements used to perform the calculation.

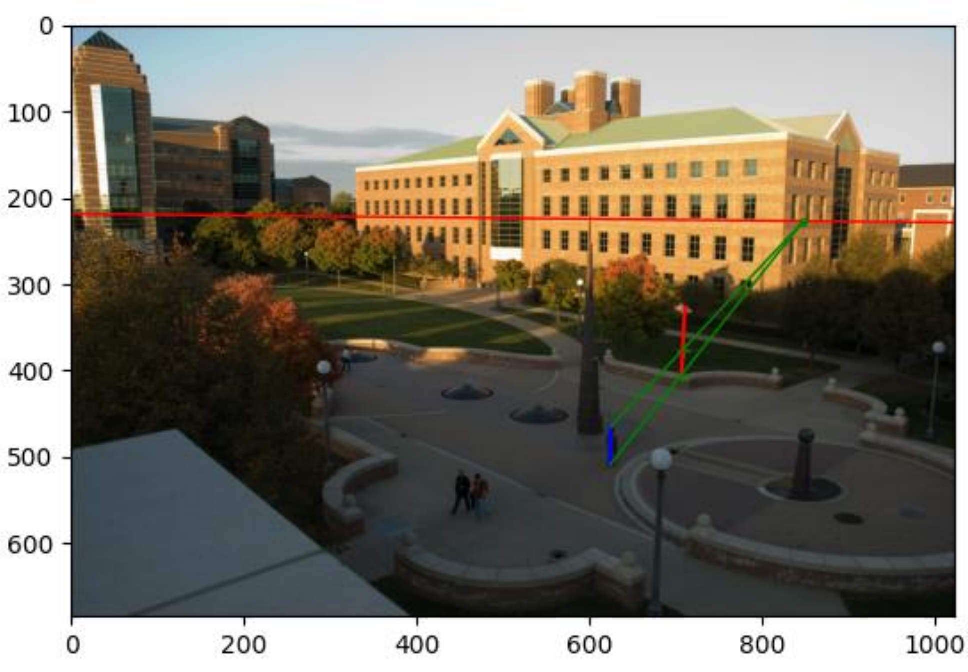
```
Click on the bottom coordinate of the lamp posts
Height of CSL building is: 31.03936544788752 meters
Height of the spike statue is: 9.395838393395714 meters
Height of the lamp posts is: 4.4943715569645235 meters
```




0
100
200
300
400
500
600
0
200
400
600
800
1000


0
100
200
300
400
500
600
0
200
400
600
800
1000


0
100
200
300
400
500
600
0
200
400
600
800
1000


0
100
200
300
400
500
600
0
200
400
600
800
1000

How do the answers change if you assume the person is 6ft tall?

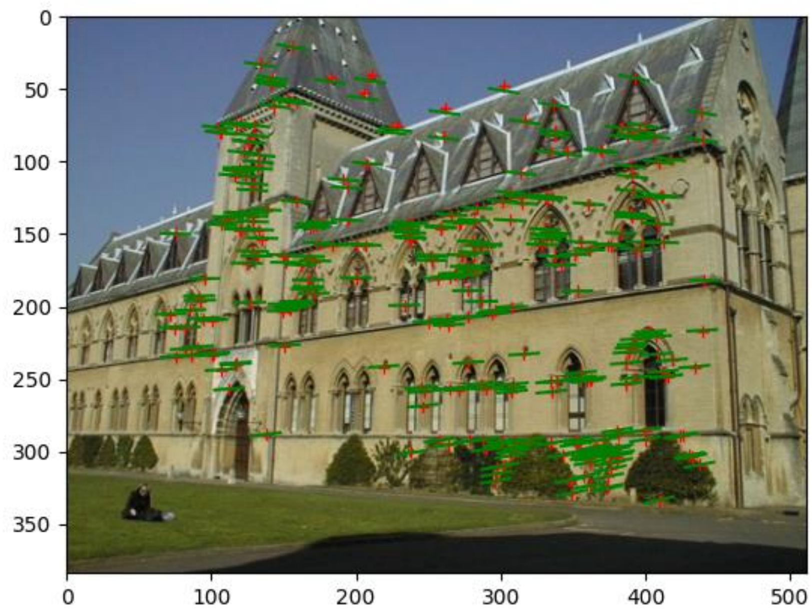
```
#H = 1.6764  # 5ft 6in to meters
H = 1.8288  # 6 ft to meters
```

```
click on the bottom coordinate of the lamp posts
Height of CSL building is: 26.803930019117047 meters
Height of the spike statue is: 9.793735346193037 meters
Height of the lamp posts is: 5.323090155855427 meters
```

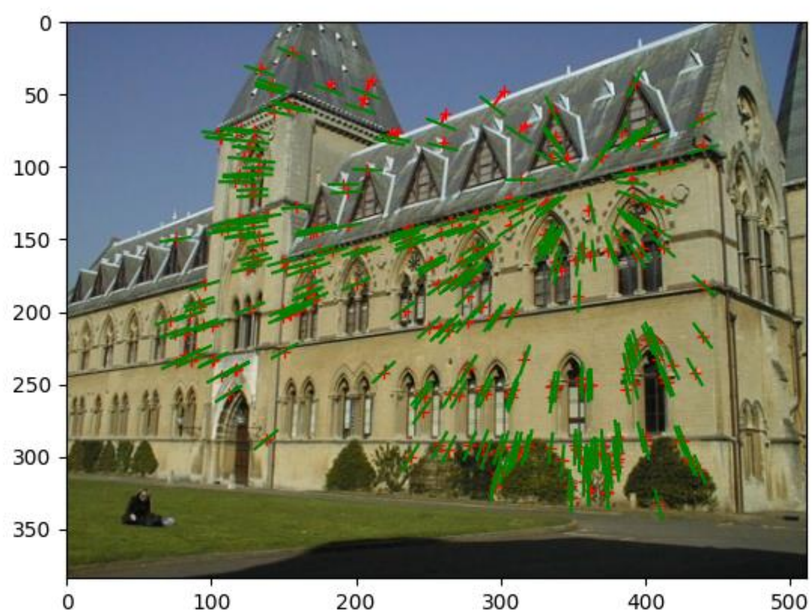
Part 2 Fundamental Matrix Estimation, Camera Calibration, Triangulation:

For the lab and library image pairs, display your result (points and epipolar lines) and report your residual for both unnormalized and normalized fundamental matrix estimation.

Normalized

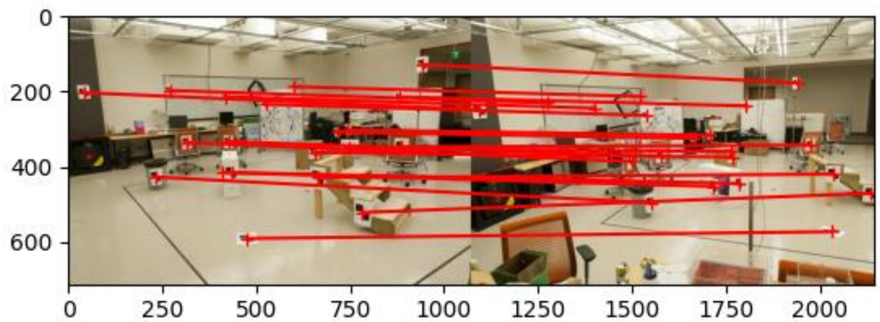


Not Normalized

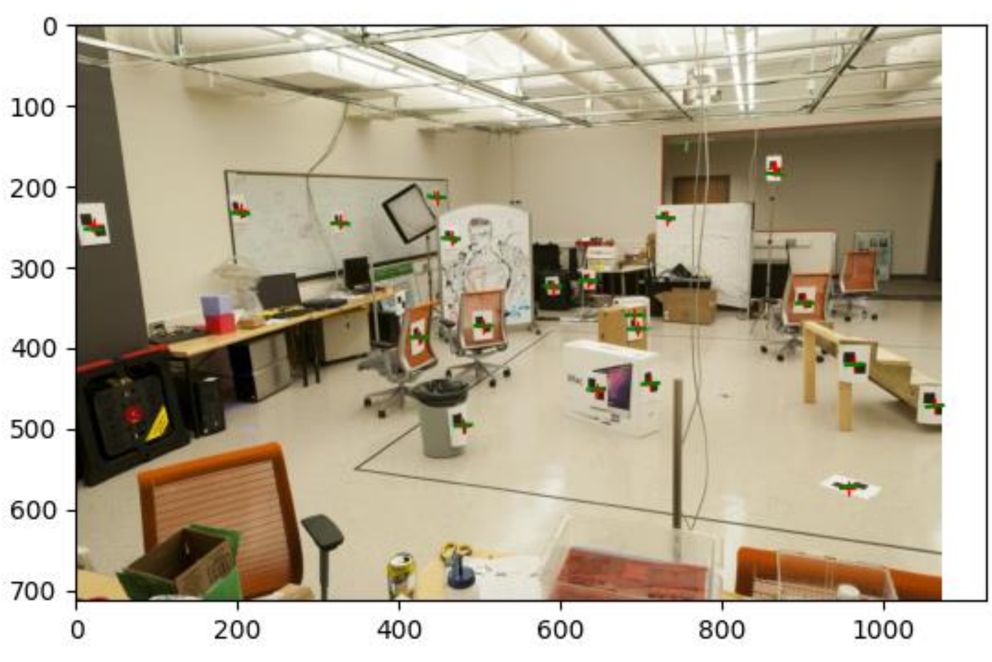


The top is the residual for normalized.
Bottom is the residual for not normalized.

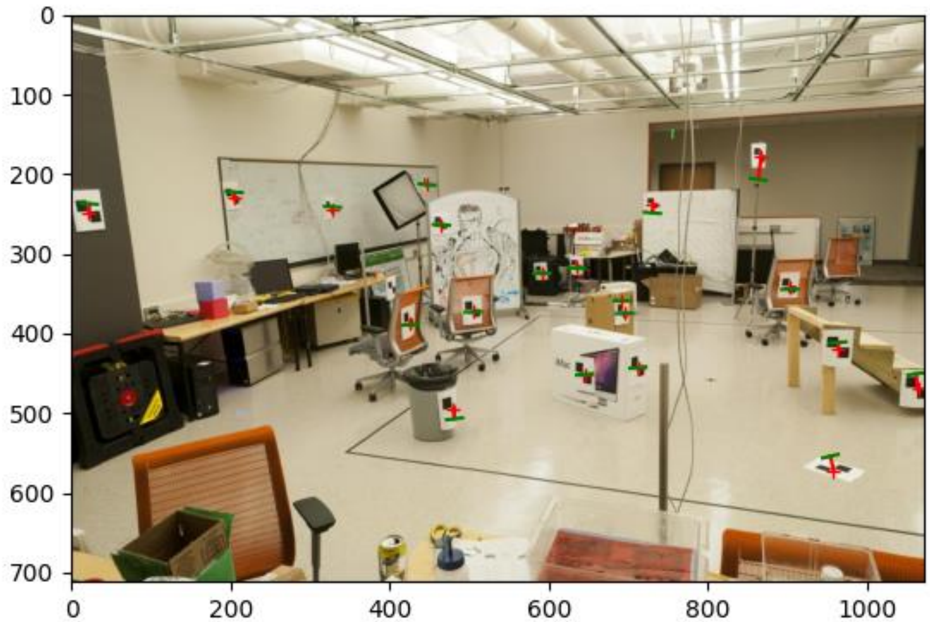
```
residual of method: 0.15550691145382828
residual of method: 0.13563748159971029
```


0
200
400
600
0
250
500
750
1000
1250
1500
1750
2000

Normalized


0
100
200
300
400
500
600
700
0
200
400
600
800
1000

Not Normalized



The top is the residual for normalized.
Bottom is the residual for not normalized.

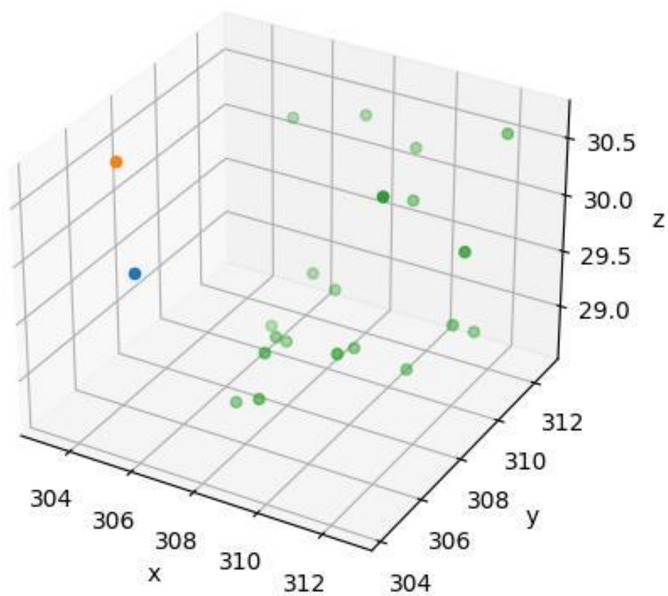
```
residual of method: 0.32929111614544426
residual of method: 1.3551018693181622
```

For the lab image pair, show your estimated 3x4 camera projection matrices. Report the residual between the projected and observed 2D points.

```
[[-3.09963996e-03 -1.46204548e-04  4.48497465e-04  9.78930678e-01]
 [-3.07018252e-04 -6.37193664e-04  2.77356178e-03  2.04144405e-01]
 [-1.67933533e-06 -2.74767684e-06  6.83964827e-07  1.32882928e-03]]
[[ 6.93154686e-03 -4.01684470e-03 -1.32602928e-03 -8.26700554e-01]
 [ 1.54768732e-03  1.02452760e-03 -7.27440714e-03 -5.62523256e-01]
 [ 7.60946050e-06  3.70953989e-06 -1.90203244e-06 -3.38807712e-03]]
13.545832895785042
15.544953461804898
```

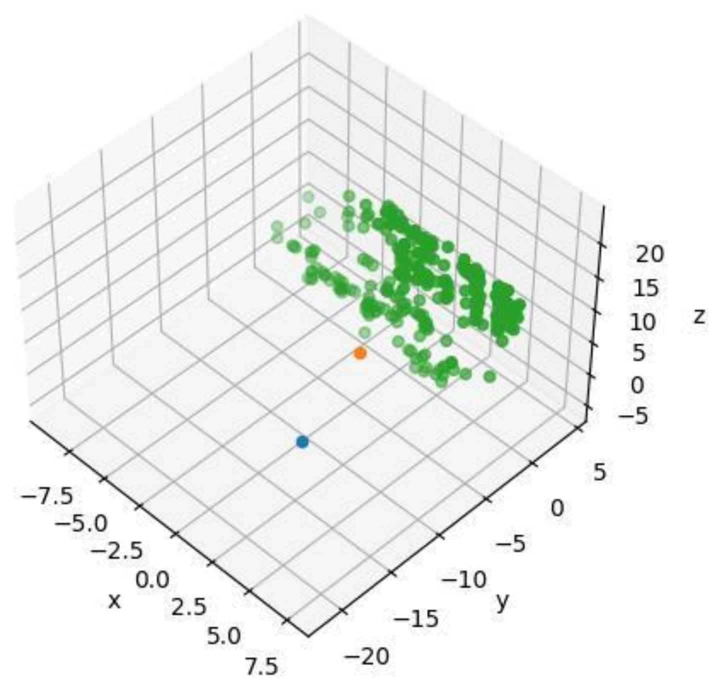
For the lab and library image pairs, visualize 3D camera centers and triangulated 3D points.

Lab



30.5
30.0
29.5
29.0
z
304
306
308
310
312
x
312
310
308
306
304
y

Library

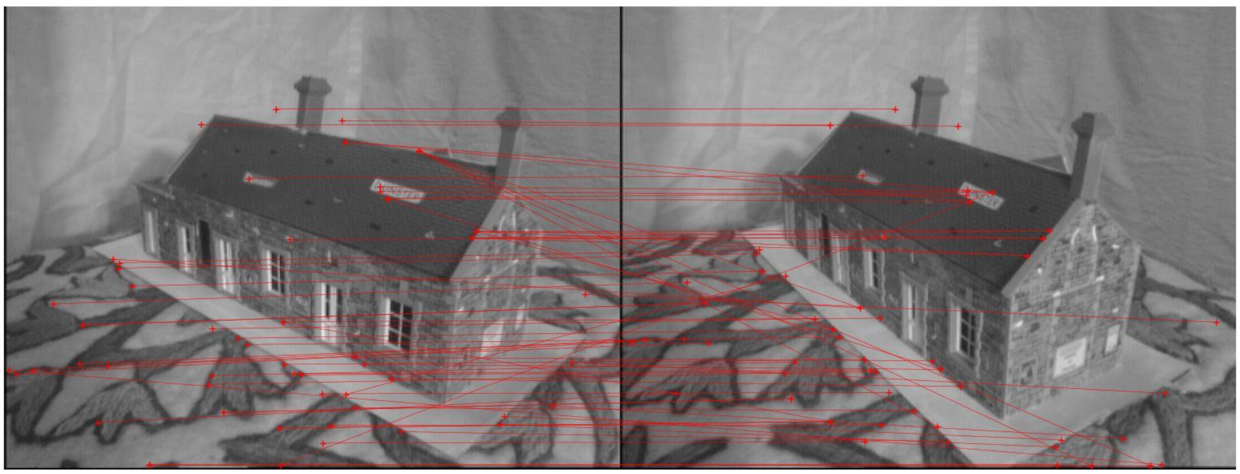


20
15
10
5
0
−5
z
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
x
5
0
−5
−10
−15
−20
y

Below shows the camera center coordinates and the error for the Lab and Library respectively

```
[305.83276769 304.20103826  30.13699243   1.        ]
[303.10003925 307.18428016  30.42166874   1.        ]
[  7.28863053 -21.52118112  17.73503585   1.        ]
[  6.89405488 -15.39232716  23.41498687   1.        ]
Mean 3D reconstuction error for the lab data:  0.00079
2D reprojection error for the lab 1 data:  24.318252955295634
2D reprojection error for the lab 2 data:  3.92328078899706
2D reprojection error for the library 1 data:  51.53227122136155
2D reprojection error for the library 2 data:  81.28132134985131
```

For the house and gaudi image pairs, display your result and report your number of inliers and average inlier residual for normalized estimation without ground truth matches.



```
Average residual: 0.43898617803870543
Number of inliers: 68
```

```
Average residual: 1.0394351411145666
Number of inliers: 141
```