

Sai Surya Salalith Mantha 9463553709

saisurym@usc.edu

Steps:

1. Find the Sift features.
2. Matches before RANSAC
3. Matches using RANSAC
4. Essential Matrix.
5. Rotation and Translation Matrix
6. Projection Matrix and Triangulation.
7. plot 3D points.

```
In [3]: # import neccessary Libraries
import cv2
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
```

```

In [58]: # SFM() function takes two images as Input
# shows the matches, Essential matrix, Rotation and translation matrix and 3d-
points.
def SFM(image1,image2):
    MIN_MATCH_COUNT = 10

    #TODO: Load Different Image Pairs
    img1=cv2.imread(image1)
    img2=cv2.imread(image2)

    #TODO: Replace K with given Intrinsic Matrix
    K = np.array([[518.86, 0., 285.58],
                  [0., 519.47, 213.74],
                  [0., 0., 1.]])

    #####
    #1----SIFT feature matching---
    #####

    #detect sift features for both images
    sift = cv2.xfeatures2d.SIFT_create()
    kp1, des1 = sift.detectAndCompute(img1,None)
    kp2, des2 = sift.detectAndCompute(img2,None)

    #use flann to perform feature matching
    FLANN_INDEX_KDTREE = 0
    index_params = dict(algorithm = FLANN_INDEX_KDTREE, trees = 5)
    search_params = dict(checks = 50)

    flann = cv2.FlannBasedMatcher(index_params, search_params)

    matches = flann.knnMatch(des1,des2,k=2)

    # store all the good matches as per Lowe's ratio test.
    good = []
    for m,n in matches:
        if m.distance < 0.7*n.distance:
            good.append(m)

    if len(good)>MIN_MATCH_COUNT:
        p1 = np.float32([ kp1[m.queryIdx].pt for m in good ]).reshape(-1,1,2)
        p2 = np.float32([ kp2[m.trainIdx].pt for m in good ]).reshape(-1,1,2)

        draw_params = dict(matchColor = (0,255,0), # draw matches in green color
                           singlePointColor = None,
                           flags = 2)

        img_siftmatch = cv2.drawMatches(img1,kp1,img2,kp2,good,None,**draw_params)

    ##### Display matchings before Ransac
    print("Before RANSAC:")
    plt.figure(num=None, figsize=(10, 10), dpi=80, facecolor='w', edgecolor=
'k')
    plt.imshow(img_siftmatch)

```

```

plt.show()

#####
#2---essential matrix--#
#####
E, mask = cv2.findEssentialMat(p1, p2, K, cv2.RANSAC, 0.999, 1.0);

matchesMask = mask.ravel().tolist()

draw_params = dict(matchColor = (0,255,0), # draw matches in green color
                    singlePointColor = None,
                    matchesMask = matchesMask, # draw only inliers
                    flags = 2)

img_inliermatch = cv2.drawMatches(img1,kp1,img2,kp2,good,None,**draw_params)

##### Display matchings After Ransac
print("After RANSAC:")
plt.figure(num=None, figsize=(10, 10), dpi=80, facecolor='w', edgecolor=
'k')
plt.imshow(img_inliermatch)
plt.show()
print ("Essential matrix:")
print (E)

#####
#3---recoverpose--#
#####

points, R, t, mask = cv2.recoverPose(E, p1, p2)
print ("Rotation:")
print (R)
print ("Translation:")
print (t)
# p1_tmp = np.expand_dims(np.squeeze(p1), 0)
p1_tmp = np.ones([3, p1.shape[0]])
p1_tmp[:2,:] = np.squeeze(p1).T
p2_tmp = np.ones([3, p2.shape[0]])
p2_tmp[:2,:] = np.squeeze(p2).T
print ((np.dot(R, p2_tmp) + t) - p1_tmp)

#####
#4---triangulation---#
#####

#calculate projection matrix for both camera
M_r = np.hstack((R, t))
M_l = np.hstack((np.eye(3, 3), np.zeros((3, 1))))

P_l = np.dot(K, M_l)
P_r = np.dot(K, M_r)

# undistort points
p1 = p1[np.asarray(matchesMask)==1,:,:]
p2 = p2[np.asarray(matchesMask)==1,:,:]
p1_un = cv2.undistortPoints(p1, K, None)

```

```

p2_un = cv2.undistortPoints(p2, K, None)
p1_un = np.squeeze(p1_un)
p2_un = np.squeeze(p2_un)

#triangulate points this requires points in normalized coordinate
point_4d_hom = cv2.triangulatePoints(M_l, M_r, p1_un.T, p2_un.T)
point_3d = point_4d_hom / np.tile(point_4d_hom[-1, :], (4, 1))
point_3d = point_3d[:3, :].T

#####
#5---output 3D pointcloud--#
#####
#TODO: Display 3D points
print("3-D points")
print("-----Multiple Views-----"
)

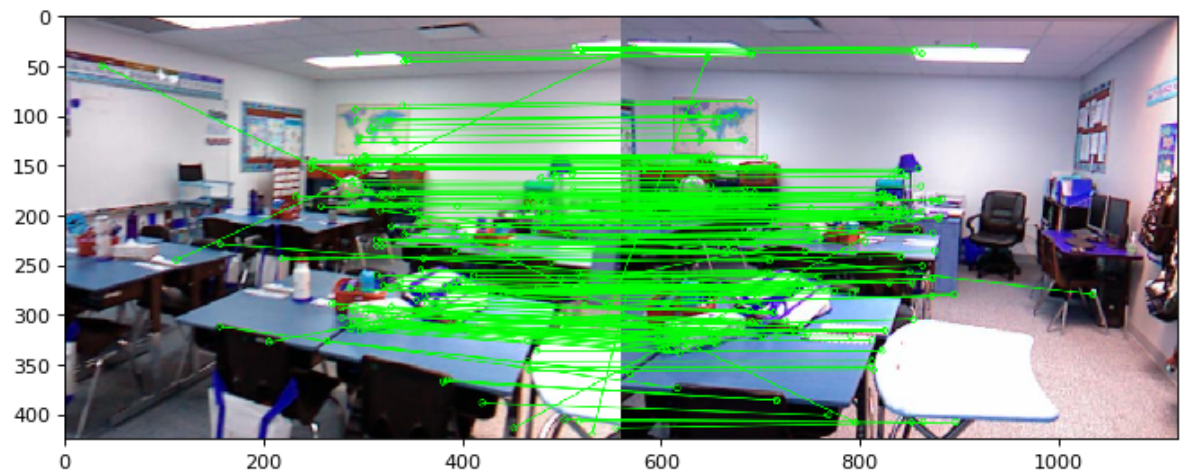
fig = plt.figure(num=None, figsize=(14, 9), dpi=80, facecolor='w', edgecolor='k')
ax1 = fig.add_subplot(2, 2, 1, projection='3d')
ax1.scatter(point_3d[:,0], point_3d[:,1], point_3d[:,2])
ax1.view_init()
ax2 = fig.add_subplot(2, 2, 2, projection='3d')
ax2.scatter(point_3d[:,0], point_3d[:,1], point_3d[:,2])
ax2.view_init(20,20)
ax3 = fig.add_subplot(2, 2, 3, projection='3d')
ax3.scatter(point_3d[:,0], point_3d[:,1], point_3d[:,2])
ax3.view_init(20,40)
ax4 = fig.add_subplot(2, 2, 4, projection='3d')
ax4.scatter(point_3d[:,0], point_3d[:,1], point_3d[:,2])
ax4.view_init(20,60)
plt.show()

# fig = plt.figure()
# ax = Axes3D(fig)
# ax.autoscale(enable=True,axis='both',tight=None)
# ax.scatter(point_3d[:,0], point_3d[:,1], point_3d[:,2])
# # ax.view_init(20,60)
# plt.show()

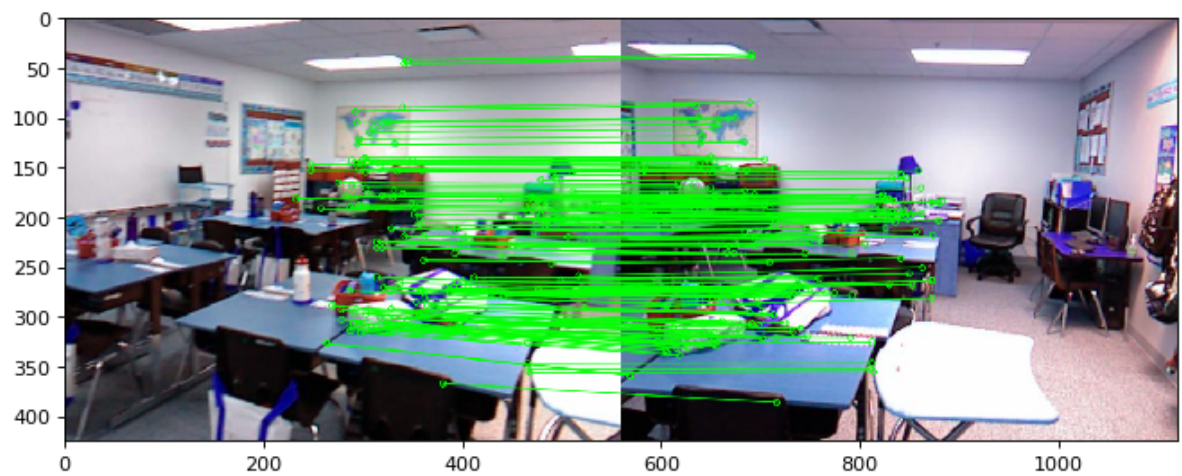
```

```
In [59]: SFM('HW4_data/a1.png', 'HW4_data/a2.png')
```

Before RANSAC:



After RANSAC:



Essential matrix:

```
[[ 0.03528635 -0.64288702  0.00591349]
 [ 0.49341659  0.02860999 -0.50550657]
 [-0.0264357  0.29126999  0.00798598]]
```

Rotation:

```
[[ 0.93091564  0.04776478 -0.3620975 ]
 [-0.04716958  0.99883182  0.01048912]
 [ 0.36217552  0.0073155  0.93208121]]
```

Translation:

```
[[ -0.41331871]
 [ -0.01921728]
 [ -0.91038362]]
```

```
[[ 2.01093932e+02 -1.00079762e+02 -8.64155536e+01  7.58626155e+01
   1.02391160e+02  8.60376854e+01 -2.17275593e+02 -2.19148232e+02
  -2.19497506e+02 -2.20954578e+02 -2.16717648e+02 -2.19676020e+02
  -2.39285400e+02 -2.39285400e+02 -2.31986772e+02 -2.14733153e+02
  -2.31372076e+02 -2.30190552e+02 -2.12960135e+02 -2.29142655e+02
  -2.30760233e+02 -2.10340313e+02 -2.28887388e+02 -2.14436623e+02
  -2.14191896e+02 -2.28607618e+02 -2.29520346e+02 -2.18574369e+02
  -2.30209497e+02 -2.12923531e+02 -2.16655371e+02  3.70077667e+01
  -2.15273890e+02 -2.28015057e+02 -2.08082235e+02 -2.15096528e+02
  -2.12147634e+02 -2.27875776e+02 -2.13017857e+02 -2.11478218e+02
  -2.12049743e+02 -2.12049743e+02 -2.11325631e+02 -2.10403697e+02
  -2.28982839e+02 -2.27219050e+02 -2.09231242e+02 -2.12360370e+02
  -2.15080955e+02 -2.15570924e+02 -2.24099734e+02 -2.04625539e+02
  -2.03145805e+02 -2.03145805e+02 -2.26844565e+02 -2.10134251e+02
   1.37270575e+02 -2.24873766e+02 -2.02864499e+02 -2.07734701e+02
  -2.15345623e+02 -2.22845029e+02 -2.22383272e+02 -2.21687140e+02
  -2.22264896e+02 -2.24116354e+02 -2.15003328e+02 -2.22179591e+02
  -2.21887083e+02 -2.23660535e+02 -2.20225862e+02 -2.03341328e+02
  -2.07132428e+02 -2.07963590e+02 -2.12058952e+02 -2.10781577e+02
  -2.19802817e+02 -2.21197757e+02 -2.15960768e+02 -2.07554911e+02
  -2.07554911e+02 -2.18505599e+02 -2.18505599e+02 -2.07294432e+02
  -2.22110233e+02 -2.19095672e+02 -2.11784129e+02 -2.06158169e+02
  -2.09599866e+02 -2.05343969e+02 -2.05919514e+02 -2.20531131e+02
  -2.04746899e+02 -2.20547242e+02 -2.20547242e+02 -2.17600483e+02
  -2.04236320e+02 -2.10325131e+02 -2.08779577e+02 -2.19083424e+02
  -2.09151556e+02 -2.11586334e+02 -2.09505919e+02 -2.17506139e+02
  -2.19767768e+02 -2.11813794e+02 -2.17185827e+02 -2.17473891e+02
  -2.16988748e+02 -2.16988748e+02 -2.15083167e+02 -2.17454009e+02
  -2.16000986e+02 -2.19231472e+02 -2.21849761e+02 -2.16701757e+02
  -2.16195483e+02 -2.15868835e+02 -2.15584646e+02 -2.15584646e+02
  -2.10140532e+02 -2.10749733e+02 -2.17227387e+02 -2.09181722e+02
  -2.15449808e+02 -2.15589181e+02 -2.11312408e+02 -2.18472022e+02
  -2.10176951e+02 -2.14578213e+02 -2.16407098e+02 -2.10822289e+02
  -2.14818788e+02 -2.07743137e+02 -2.16882691e+02 -2.17587664e+02
  -2.13026679e+02 -2.12951557e+02 -2.13045311e+02 -2.21235245e+02
  -2.12687704e+02 -1.94878817e+02 -2.12129303e+02 -2.15253380e+02
  -2.15318057e+02 -2.15149370e+02 -2.16668217e+02 -2.15228760e+02
  -2.15562627e+02 -2.16763736e+02 -2.16763736e+02 -2.16388651e+02
  -2.15056795e+02 -2.16591411e+02 -2.15444121e+02 -2.17609300e+02
  -2.13993526e+02 -2.17770404e+02 -2.17770404e+02 -2.17592806e+02
  -2.17592806e+02 -2.19124347e+02 -2.17971767e+02 -2.17971767e+02
  -2.17093211e+02 -2.17897840e+02 -2.17106509e+02 -2.19174046e+02
  -2.22504542e+02 -2.22504542e+02 -2.19368719e+02 -2.19586905e+02
  -2.21664040e+02 -2.19217542e+02 -2.19417554e+02 -2.17687033e+02
  -2.15494336e+02 -2.13214233e+02 -2.12203699e+02 -2.20585686e+02]
```

-2.25185381e+02 -2.45572949e+02 -2.23410908e+02 -2.37705928e+02
-2.23831713e+02 -2.23165034e+02 -2.15561198e+02 -2.37768049e+02
-2.37768049e+02 -2.19093650e+02 -2.20768161e+02 -2.20768161e+02
-2.25295188e+02 -2.24194073e+02 -5.11613215e+02 -2.26844811e+02
-2.27988363e+02 -2.27080861e+02 -4.58094866e+02 -2.20444109e+02
-4.66283343e+02 -2.22741727e+02]
[3.46355160e+02 -2.14085458e+02 5.82802197e+01 3.41873899e+01
-1.41926272e+02 -2.31754337e+01 2.51107083e+00 -1.10748041e+00
-1.30088556e+00 -2.11767176e+00 -4.96733156e+00 2.57720729e+00
3.13101693e+01 3.13101693e+01 1.96194133e+01 -1.35321987e+00
1.71049150e+01 1.94703817e+01 -1.47764952e+00 1.98264761e+01
1.25100305e+01 -9.73943979e-01 1.98269661e+01 -3.50090026e+00
-4.68669659e+00 1.87537091e+01 1.57385465e+01 -9.71057253e+00
1.46147831e+01 -4.53331542e+00 -8.97560869e+00 -2.43252053e+01
-5.65754718e+00 1.90220265e+01 -1.67915033e+00 -7.32765547e+00
-4.47910826e+00 1.92139447e+01 -5.42783572e+00 -5.48063666e+00
-5.50726290e+00 -5.50726290e+00 -4.43308422e+00 -4.90363031e+00
1.45208162e+01 1.62145992e+01 -3.18674840e+00 -5.75565507e+00
-7.87217987e+00 -8.32623884e+00 1.41110140e+01 -4.68847583e-02
-9.52830904e-01 -9.52830904e-01 5.36484887e+00 -6.78420544e+00
6.16230910e+01 1.19321029e+01 -1.45076745e-01 -4.25875034e+00
-9.40993269e+00 5.57981496e+00 1.22598619e+01 1.15280743e+01
1.10033068e+01 1.05247400e+01 -9.31102347e+00 5.12194153e+00
1.07433938e+01 1.23502827e+01 7.23969013e+00 -2.87242460e+00
-5.51044166e+00 -5.42715832e+00 -8.15730799e+00 -6.66036749e+00
2.85575390e+00 2.37741496e+00 -1.08101846e+01 -5.90211082e+00
-5.90211082e+00 -1.28393519e+01 -1.28393519e+01 -4.59591423e+00
-1.27813578e+01 6.40757155e+00 -3.55491132e+00 -6.00969899e+00
-8.67688824e+00 -5.25452507e+00 -6.50006778e+00 3.80568340e+00
-5.61335858e+00 2.96301011e+00 2.96301011e+00 2.41177376e-01
-6.03454681e+00 -5.26704947e+00 -9.50808142e+00 4.93588265e-01
-8.86942480e+00 -4.93562993e+00 -5.60610871e+00 3.34184876e-01
3.14685971e+00 -6.28947853e+00 4.18756471e+00 8.67141447e-01
-2.85146582e-01 -2.85146582e-01 1.60257316e+00 1.08205754e+01
-8.82050885e-01 -3.60662158e+00 1.25000798e+01 -4.27359511e+00
-4.34153298e+00 -1.27478322e+00 -6.24508063e+00 -6.24508063e+00
-8.89260446e+00 -9.36487079e+00 -6.90624252e+00 -7.49929922e+00
-1.70894031e+00 -2.21034412e+00 -9.75707067e+00 -7.81918144e+00
-1.05374954e+01 -2.16921940e+00 -7.05691492e+00 -1.07373776e+01
-8.84548694e+00 2.04440743e+00 -9.29852378e+00 -1.09286269e+01
-1.15352093e+01 -1.36411138e+01 -1.22281221e+01 -1.03583397e+01
-1.53329213e+01 -2.15022801e+02 -1.44457190e+01 -1.51438159e+01
-1.25637905e+01 -1.30650144e+01 -1.30026034e+01 -1.55734664e+01
-1.37577568e+01 -1.49515076e+01 -1.49515076e+01 -1.49945322e+01
-1.64963272e+01 -1.65194779e+01 -1.29863507e+01 -1.56846456e+01
-1.79790414e+01 -1.46839875e+01 -1.46839875e+01 -1.63542315e+01
-1.63542315e+01 -1.42395153e+01 -1.59868141e+01 -1.59868141e+01
-1.70404424e+01 -1.71179552e+01 -1.93687135e+01 -1.73300978e+01
-1.46235377e+01 -1.46235377e+01 -1.76224164e+01 -1.70949506e+01
-1.57024481e+01 -1.80755275e+01 -1.82658321e+01 -2.04599820e+01
-2.07684738e+01 -2.23724888e+01 -1.14074860e+01 -1.94053962e+01
-1.60219988e+01 -1.67137267e+01 -1.75367208e+01 -1.10456854e+01
-1.88655074e+01 -2.27222230e+01 -1.41175405e+01 -1.02880652e+01
-1.02880652e+01 -2.49906055e+01 -2.50553880e+01 -2.50553880e+01
-1.89083556e+01 -2.10691492e+01 -1.48733653e+02 -2.08480345e+01
-1.95588954e+01 -1.94625897e+01 -2.77357134e+02 -2.73433650e+01
-3.70410714e+00 -2.25267191e+01]

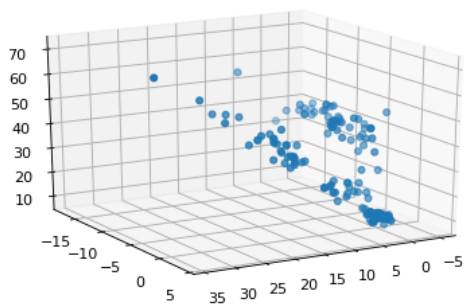
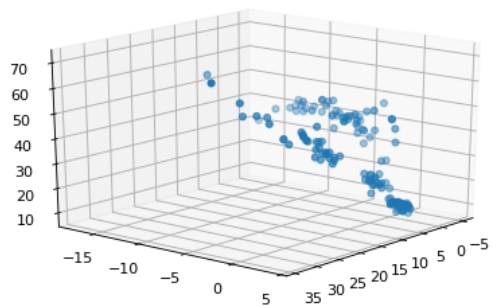
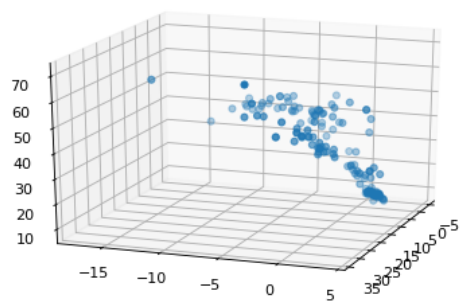
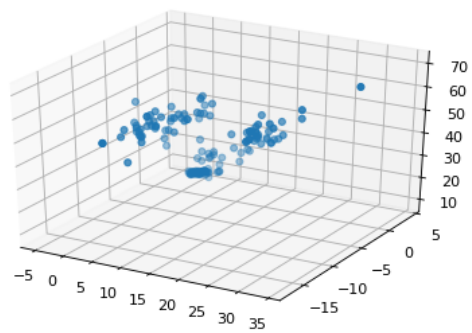

```

[ 8.73624060e+01 3.64868628e+00 2.21792801e+01 8.66514780e+01
 1.16773374e+02 1.15014570e+02 3.23386334e+00 8.66033071e+00
 8.66033071e+00 8.30020014e+00 1.10488825e+01 1.20345195e+01
 4.85523376e+00 4.85523376e+00 1.02102811e+01 2.08412155e+01
 1.41626890e+01 1.45068535e+01 2.45708773e+01 1.75853594e+01
 1.76568670e+01 2.70477704e+01 1.82922208e+01 2.61785399e+01
 2.67724868e+01 1.97720102e+01 1.98086738e+01 2.68849531e+01
 2.02638804e+01 2.88880684e+01 2.81265053e+01 1.27917814e+02
 2.87201087e+01 2.18621075e+01 3.12442485e+01 2.93843443e+01
 3.03545993e+01 2.27618938e+01 3.13922174e+01 3.23510744e+01
 3.26590117e+01 3.26590117e+01 3.28370541e+01 3.34927556e+01
 2.46301584e+01 2.55950068e+01 3.45890834e+01 3.40929569e+01
 3.38530143e+01 3.52015932e+01 3.04306597e+01 3.91651211e+01
 3.98454677e+01 3.98454677e+01 3.02064229e+01 3.91555797e+01
 1.72874429e+02 3.17817408e+01 4.19380590e+01 4.08741583e+01
 3.88365197e+01 3.42590928e+01 3.45629697e+01 3.55544552e+01
 3.55018606e+01 3.52669547e+01 4.12627332e+01 3.72257828e+01
 3.69696438e+01 3.63305858e+01 3.80416148e+01 4.57020206e+01
 4.51492617e+01 4.52592727e+01 4.46535949e+01 4.58091185e+01
 4.14352624e+01 4.14477146e+01 4.64298988e+01 4.88665771e+01
 4.88665771e+01 4.66919191e+01 4.66919191e+01 4.90998064e+01
 4.65898095e+01 4.47192163e+01 4.99599510e+01 5.27047276e+01
 5.22935797e+01 5.40700165e+01 5.41167581e+01 4.77869814e+01
 5.60128779e+01 4.89803051e+01 4.89803051e+01 5.05052093e+01
 5.66439159e+01 5.39541574e+01 5.59163999e+01 5.05052093e+01
 5.59163999e+01 5.48771667e+01 5.59974800e+01 5.22320365e+01
 5.15203782e+01 5.53404654e+01 5.28188258e+01 5.33019062e+01
 5.66826175e+01 5.66826175e+01 5.78234623e+01 5.84572021e+01
 6.08119943e+01 6.00105613e+01 5.84572021e+01 6.20353867e+01
 6.29885798e+01 6.37784580e+01 6.47768115e+01 6.47768115e+01
 6.77036002e+01 6.82541682e+01 6.52554262e+01 6.94153334e+01
 6.66416796e+01 6.66416796e+01 6.94591493e+01 6.70750108e+01
 7.18702992e+01 6.90925400e+01 6.97393148e+01 7.32047358e+01
 7.30538481e+01 7.73561738e+01 7.56608846e+01 7.87014085e+01
 8.50070444e+01 8.46930000e+01 8.50884224e+01 8.48178036e+01
 8.95090379e+01 9.69082148e+01 9.13835771e+01 9.36265364e+01
 9.24403297e+01 9.29422477e+01 9.26837977e+01 9.50509262e+01
 9.36779305e+01 9.57565873e+01 9.57565873e+01 9.63130594e+01
 9.69082148e+01 9.75930935e+01 9.65218417e+01 9.78812119e+01
 9.86161232e+01 9.73813213e+01 9.73813213e+01 9.88970697e+01
 9.88970697e+01 9.87148658e+01 9.92579964e+01 9.92579964e+01
 1.00490065e+02 1.01048677e+02 1.02713002e+02 1.02768204e+02
 1.02034844e+02 1.02034844e+02 1.03011280e+02 1.02977747e+02
 1.03925433e+02 1.04788987e+02 1.05699817e+02 1.05829782e+02
 1.06967003e+02 1.07776132e+02 1.07765939e+02 1.08044013e+02
 1.08448471e+02 1.00602431e+02 1.09527135e+02 1.05920001e+02
 1.09988505e+02 1.10761195e+02 1.12160309e+02 1.08808186e+02
 1.08808186e+02 1.14340482e+02 1.14278209e+02 1.14278209e+02
 1.13576422e+02 1.14142602e+02 3.65664730e+00 1.16266420e+02
 1.16052482e+02 1.17353337e+02 3.07776093e+01 1.22365699e+02
 3.07776093e+01 1.23895581e+02]]

```

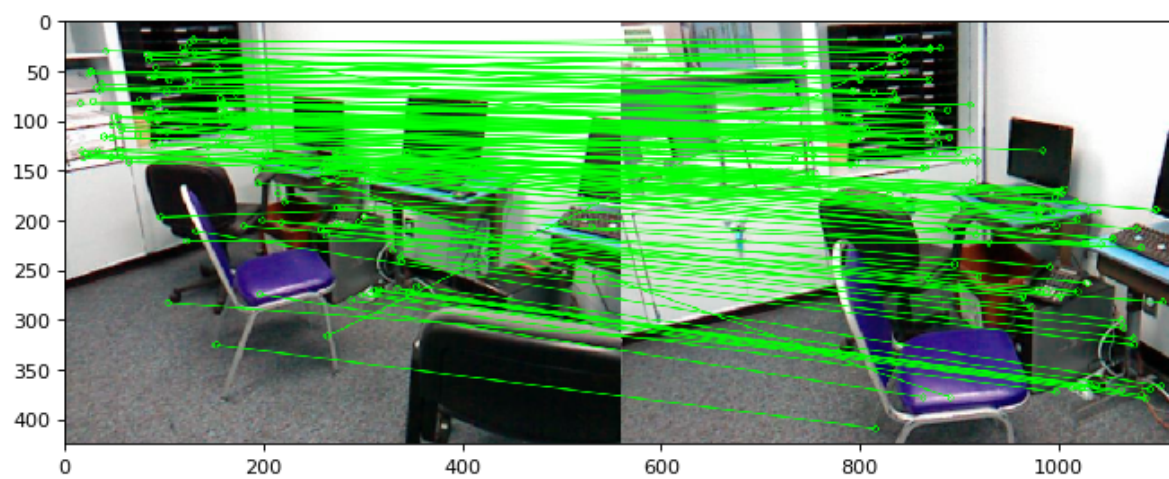
3-D points

-----Multiple Views-----

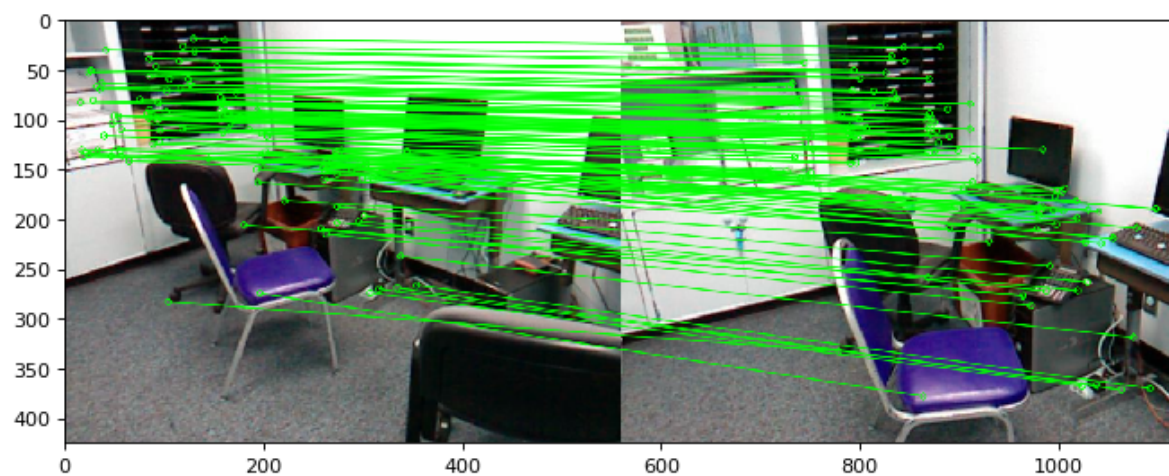


```
In [60]: SFM('HW4_data/b1.png', 'HW4_data/b2.png')
```

Before RANSAC:



After RANSAC:



Essential matrix:

```
[[ 0.02594348 -0.59135081 -0.27692398]
 [ 0.64062882 -0.06831098 -0.07001068]
 [ 0.2957447  0.23831386  0.09334175]]
```

Rotation:

```
[[ 0.94272111 -0.10665497  0.31607219]
 [ 0.10759025  0.99408894  0.0145439 ]
 [-0.31575505  0.02029545  0.94862366]]
```

Translation:

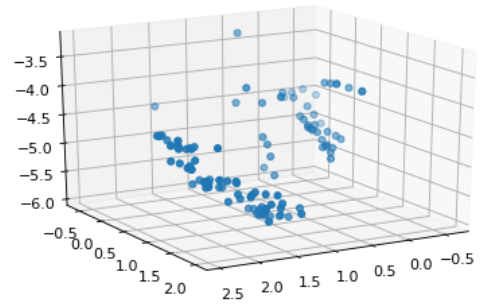
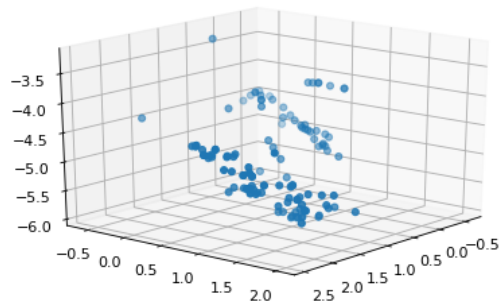
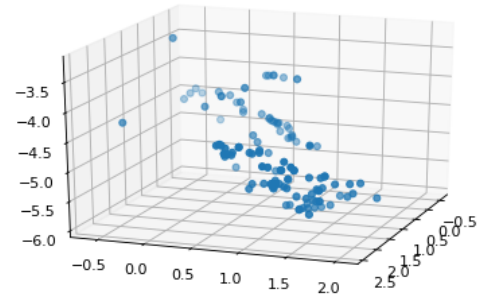
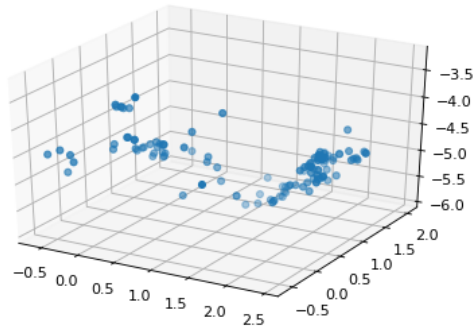
```
[[ 0.38195359]
 [-0.40006709]
 [ 0.83310131]]
```

```
[[ 123.34609333 108.93571589 107.47021546 107.79233068 128.90350374
 106.87405328 129.76228186 129.76228186 122.74973904 125.98714594
 108.24373106 125.19498899 124.74137917 110.6782712 110.6782712
 128.0789267 107.12209375 107.12209375 113.92997837 77.971339
 115.03098914 108.09969341 114.58411408 112.10188019 107.93308882
 111.83057385 106.85122064 105.76806603 110.77253225 123.55071711
 129.40675038 129.40675038 119.83676156 119.83676156 118.78867967
 130.41762124 130.41762124 120.49587762 126.7643935 118.25289251
 113.5985867 118.75947897 200.62548344 200.62548344 128.80302481
 118.86526876 118.86526876 120.94661373 120.94661373 112.99183729
 115.6322489 95.31134284 120.07380704 126.25669911 117.46588187
 67.38383877 127.40797301 131.02462708 126.25322249 116.57150592
 134.33282649 169.4902448 96.21745951 294.73902281 127.55807585
 165.43029301 165.43029301 136.18803072 126.76368745 126.76368745
 136.35222725 134.68430958 341.86029136 130.04293023 130.04293023
 117.55699233 130.12540978 91.45786073 45.91698144 132.64817593
 124.65429289 117.10073969 126.05783339 126.05783339 123.34715106
 122.39107769 118.76861755 125.13735646 139.37242158 125.06429395
 108.32558178 119.07398356 123.36120342 121.36586555 128.64861004
 123.59598955 123.59598955 84.15399199 121.11450256 123.19272265
 130.536511 104.35462066 125.65924361 95.07681794 50.43939035
 110.27152724 121.85803398 91.73005044 107.10966461 114.0522823
 -34.35806038 120.04607742 120.04607742 99.31227467 94.30324359
 104.64725193 129.79958826 94.29998383 94.29998383 114.3823687
 111.17035446 114.41084232 115.75751161 95.57976212 116.37950074
 5.81272267 118.75141251 121.16455098 122.84666157 107.27264588
 101.0101966 116.78959033 124.69266512 121.45764269 124.53798223
 104.62609139 126.37614849 85.63566811 122.80457668 109.2196782
 126.06938229 90.06314541 117.85170133 113.07786529 128.9663125
 91.14628113 121.88460347 94.80595189 96.06377696 127.10949508
 127.10949508 -90.81123502 115.37046577 114.51302692 136.09171885
 100.74371048 114.26525425 113.41061462 143.56658865 143.56658865
 107.12590799 107.58747797 109.0548499 -11.02204266 -12.42064665
 -34.87239851 -60.22610719]
 [ 31.27797882 35.38721379 36.22234349 36.46431118 30.86129519
 37.07345635 30.92789308 30.92789308 32.91714618 32.76531379
 38.07047396 32.94689758 34.0534324 38.85803873 38.85803873
 31.86917548 40.0399439 40.0399439 38.06181835 22.23071133
 37.08566116 40.34072466 39.07923492 66.48701503 41.06831402
 39.6734287 42.53824223 43.43246111 1.08714719 37.5362106
 61.14848095 61.14848095 40.57914698 40.57914698 41.1675863
 36.2026017 36.2026017 40.58787039 38.017057 41.55621986
 42.96099991 41.62610978 4.73827212 4.73827212 37.00105511
 42.39913522 42.39913522 40.93037529 40.93037529 44.31614906
 43.63322208 -3.54968873 42.52483951 40.73544099 -27.57874072
```

70.70958502	38.99435761	39.63933475	42.77410067	47.15413146
38.29084158	6.82237475	74.69852849	50.70926562	42.33654754
7.08120577	7.08120577	25.51534346	43.34086417	43.34086417
38.63089443	39.67654461	210.73519052	18.70012684	18.70012684
45.48847835	52.60662139	67.0814847	108.41699443	44.73684325
49.07277534	289.31958692	48.63966405	48.63966405	50.21382148
50.57943172	53.20238096	49.63263507	41.47218628	50.04238275
56.22318198	53.4249106	51.28433695	52.64105945	50.85926764
53.19078579	53.19078579	83.92072561	56.70308945	59.7233945
53.02012589	72.07407989	55.13018669	79.95726688	132.99154164
67.99880488	60.31430442	78.72353165	72.5436771	66.80587373
114.59858895	61.79504124	61.79504124	78.60092489	91.37099509
79.38590385	74.50928795	108.60196825	108.60196825	88.85483856
93.33619255	90.9439523	88.82038538	113.8656418	87.86522578
-255.59398867	90.72748027	85.71197982	85.27221582	103.92693525
112.57479851	95.10011206	87.02155886	91.48659206	85.18862977
117.25008474	90.59197619	136.31603547	96.45057856	116.7886475
95.86981972	138.90462856	112.54790392	114.93481067	97.61590917
140.46402721	115.53052907	142.98782773	145.3105809	116.28233894
116.28233894	58.87529344	128.61360398	129.95280244	116.67231918
153.44222032	134.50112232	136.8506626	114.35035141	114.35035141
156.9283074	159.71658899	158.05057465	70.249692	70.28553591
97.31735311	75.12406214]			
[-47.03628066	-43.45274925	-43.71159662	-44.56126337	-51.25095514
-45.67751147	-52.25470302	-52.25470302	-51.19684298	-53.16050809
-48.6776048	-54.07286948	-54.4229348	-51.41329162	-51.41329162
-56.40875551	-53.02016068	-53.02016068	-55.07270406	-43.09608091
-56.0017962	-55.05583857	-57.12523919	-57.72063373	-56.3929216
-57.55229795	-56.52457985	-58.58563072	-62.19138422	-67.12962167
-71.28198915	-71.28198915	-68.76032435	-68.76032435	-68.84964271
-72.05085767	-72.05085767	-69.6359074	-71.28198915	-69.78076932
-69.02513189	-70.30578729	-96.77762901	-96.77762901	-73.81597383
-71.92730594	-71.92730594	-72.55733733	-72.55733733	-70.98633673
-72.00286674	-65.85542857	-75.01415153	-77.145454	-73.81597383
-61.55465184	-78.01085478	-82.21116318	-81.48324796	-79.64465489
-84.37995112	-96.77762901	-73.19586596	-142.18699036	-84.38971795
-96.77762901	-96.77762901	-86.95536738	-85.07679765	-85.07679765
-88.43116987	-88.47333196	-163.40327659	-88.12426973	-88.12426973
-85.85422506	-92.89687971	-80.85918361	-71.76085059	-95.76118012
-94.16047591	-96.15319407	-95.20655402	-95.20655402	-94.81470253
-95.08914009	-94.18155708	-96.10392912	-100.04832108	-96.57740258
-91.78510483	-95.98623268	-97.19538114	-97.48109753	-101.14554982
-101.05127111	-101.05127111	-91.51908053	-103.54891965	-105.85479706
-108.30664263	-101.44163561	-107.75835506	-99.17450433	-87.50461522
-104.64679895	-108.33977001	-100.15598962	-104.41156376	-107.44544991
-57.55062763	-109.52031071	-109.52031071	-107.53089046	-107.57027694
-111.55527188	-130.62476232	-121.2200933	-121.2200933	-127.19885595
-126.88696733	-128.11421861	-128.51020978	-123.55128102	-129.05904501
-89.96929723	-132.65231117	-133.38593001	-134.2927956	-130.46786056
-128.77060199	-133.26368033	-135.69493431	-135.67381292	-136.59401815
-133.20721124	-140.09359427	-129.81603709	-140.04699931	-138.7482673
-143.65566811	-134.84644683	-142.18699036	-141.91894447	-146.99395446
-138.16006158	-147.08633735	-140.7740706	-142.74241901	-151.06096448
-151.06096448	-80.74530679	-151.99797083	-152.6450507	-158.73643098
-150.51551097	-155.43926349	-155.52163312	-165.32994091	-165.32994091
-156.75791695	-158.28448591	-159.77032964	-135.62364404	-135.69493431
-165.55152769	-159.9070056]]			

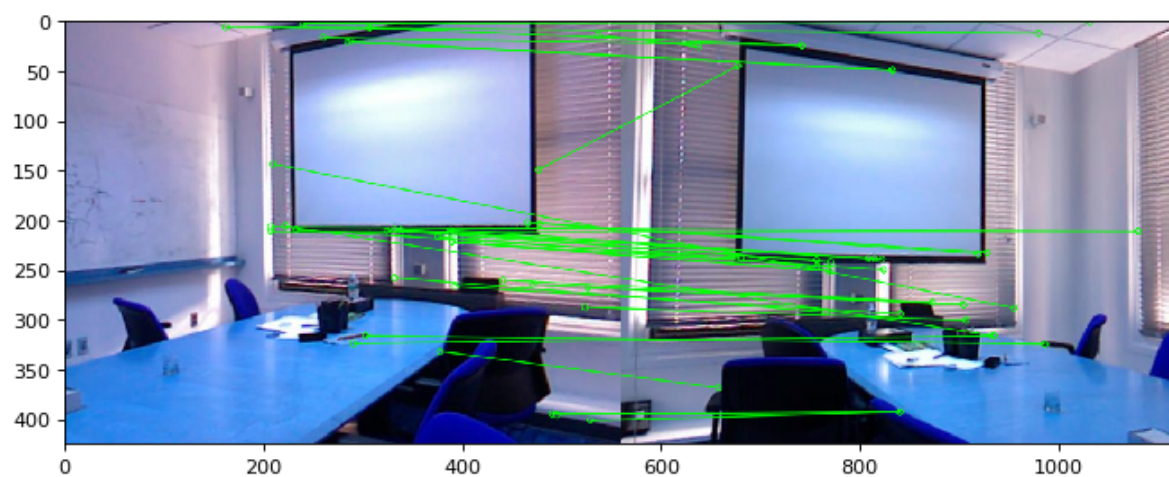
3-D points

-----Multiple Views-----

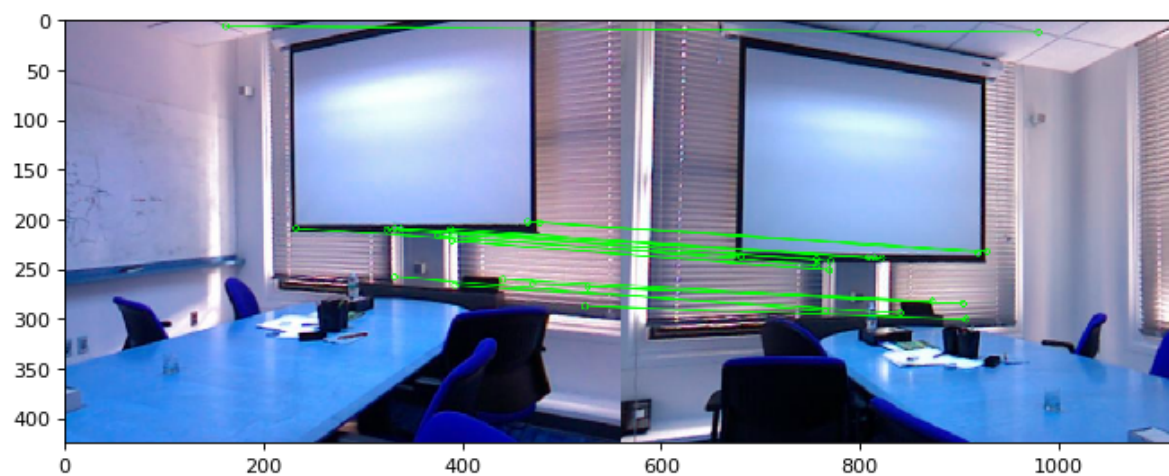


```
In [61]: SFM('HW4_data/c1.png', 'HW4_data/c2.png')
```


Before RANSAC:



After RANSAC:



Essential matrix:

```
[[ 0.00473978  0.31054486  0.00603249]
 [ 0.39721785 -0.03937412  0.58356854]
 [-0.02344382 -0.6340427  -0.03253751]]
```

Rotation:

```
[[ 0.98971631 -0.01233748 -0.14251108]
 [-0.02016788 -0.99835704 -0.05363276]
 [-0.14161524  0.05595537 -0.98833907]]
```

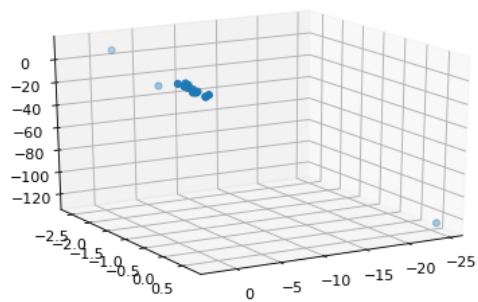
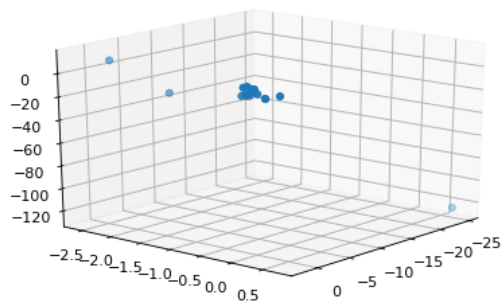
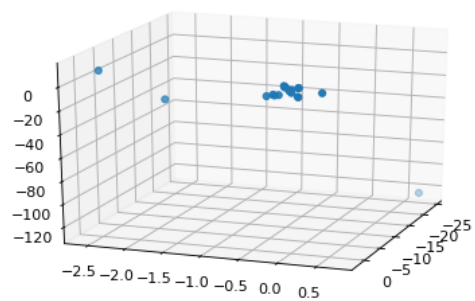
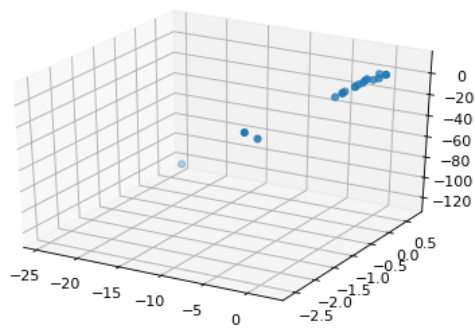
Translation:

```
[[0.89833516]
 [0.01519329]
 [0.43904795]]
```

```
[[ 254.16463803  305.27205932  305.09656799  179.27481552  145.6831153
 -115.59387625  227.74827855  -80.2156471  -13.95595786  -17.65960166
  77.50463325  116.14446292 -126.27579725 -131.61464713 -131.67502499
 -88.23975696  -88.10324334 -183.15595437 -284.55666144 -181.84326039
 -130.9417333  -130.46835824 -184.43690297 -184.43690297 -117.57086148
 -133.65320043 -112.47464312 -132.63487612 -360.33746922 -115.69036789
 -216.808904  -221.82889301 -183.02221441 -188.42973696 -255.33307392
 -459.68615832]
 [ -30.02040456 -429.95548317 -434.9502494  -441.16830446 -530.86961285
 -451.17702105  -16.06799168  -47.77756839  -78.6934831  -78.04416059
 -648.60761944 -650.05281099  -38.16576152 -456.4039462  -555.65010178
 -454.60412717 -454.58580578 -468.34917674 -702.8217101  -460.86960864
 -455.92873805 -478.48190455 -478.89993141 -478.89993141 -567.65941177
 -550.2422465  -445.1710836  -557.87338647 -199.14176043 -444.52172092
 -793.37876377 -794.34345493 -595.9366067  -560.88303002 -800.00547276
 -44.5684775 ]
 [ -60.22288799  -63.27925004  -63.27925004  -41.36811194  -36.90139519
  -5.2423673  -68.17158411  -25.80445278  -37.36354146  -37.36354146
 -36.90139519  -43.73860275  -25.80445278  -15.90895302  -14.21119388
 -23.35303962  -24.23413022  -15.62050197   5.22831562  -17.75625954
 -25.23174918  -24.918104  -17.11356037  -17.11356037  -24.83428201
 -30.04488397  -39.26365225  -34.29479031  -15.58343108  -40.64246899
 -19.26720414  -19.26720414  -33.83520464  -34.29479031  -19.26720414
 -10.9937539 ]]
```

3-D points

-----Multiple Views-----



SFM:

Structure from motion pipeline consists of finding 3D matching points. Initially find the matching points and then filter the weak matches using RANSAC. Compute the Essential matrix and decompose it to Rotation and Translation matrices. Calculate the 3D points from triangulation and matching. Add subsequent views and perform above steps repeatedly and do bundle adjustments to get the dense points reconstruction of the 3D scene.

Matchings:

Matchings are relatively better after performing RANSAC and filtering the weak matches.

Translation Matrix:

Translation vector is recovered along with the rotation matrix from the essential matrix. Translation matrix in this context refers to the position of the camera of one image with respect to another image. from the translation vector, we can estimate the positions of cameras with respect to each other. we can observe that the first value represents the lateral movement of the camera, second value represents the relative change in height and third value represents the relative change in depth stating whether the camera is moved closer or farther away from the scene.

Result:

If we observe the 3D plots we can observe that the matched points were plotted very well. Even the depth of the images were also relatively accurate in the 3d-plots from images. If we observe, in first two cases the matchings in the images are of varied depth and in the 3D plot we can observe that the points were distant along Z-axis accounting the depth. In the last image pair, the matching was present relatively at the same depth around the projector screen, we can see a cluster of points in the last graph without much variation along the z-axis.

Finally, we can need more feature matches to construct the above images completely using 3D, as a result, more camera angles.