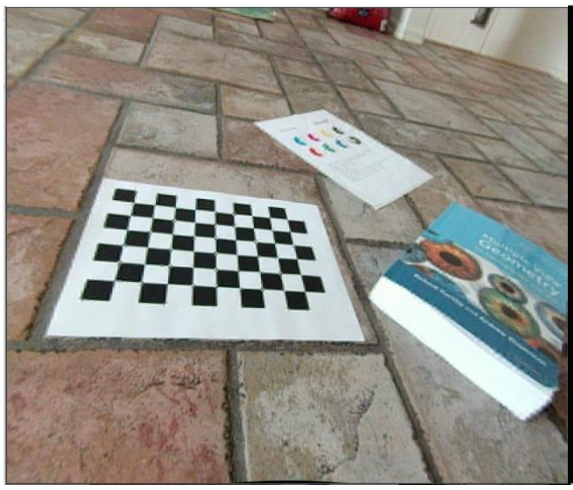
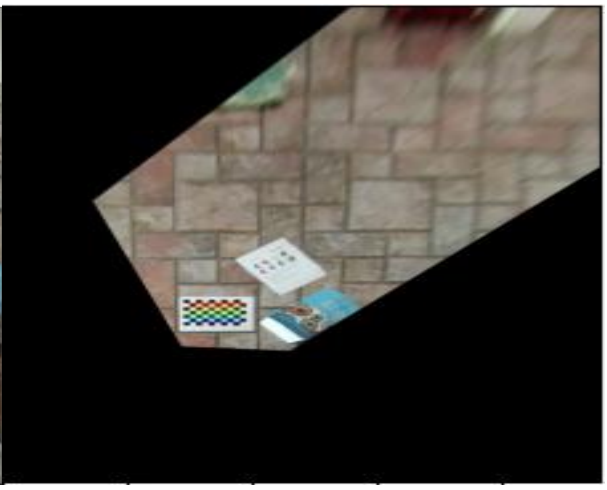
CSE 598 Perception in Robotics Project 2b: Camera Motion and Structure

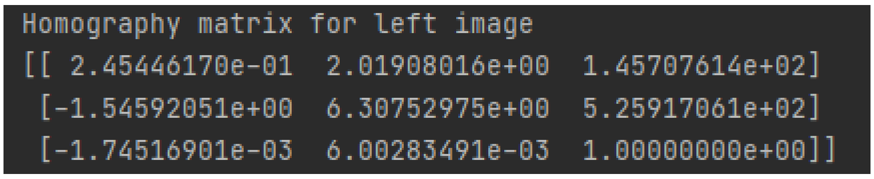
Yongbaek Cho

Task_5: Planar Homography



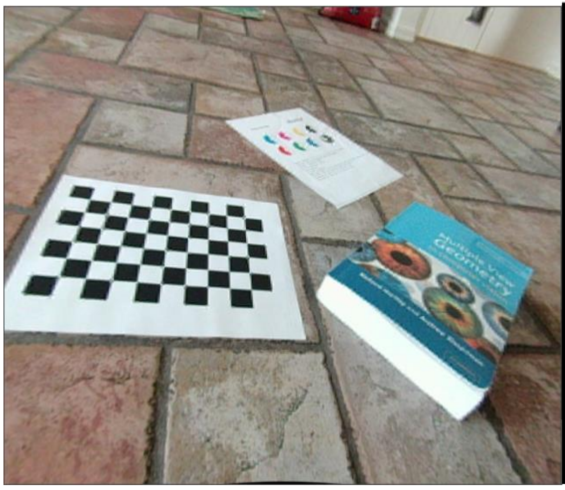


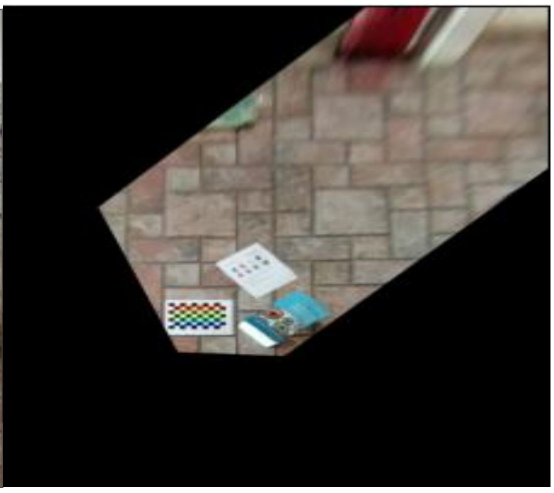
< Undistorted left_0 image >   < Warped left_0 image >



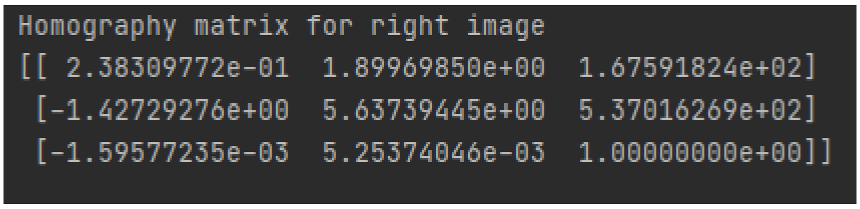
```
Homography matrix for left image
[[ 2.45446170e-01  2.01908016e+00  1.45707614e+02]
 [-1.54592051e+00  6.30752975e+00  5.25917061e+02]
 [-1.74516901e-03  6.00283491e-03  1.00000000e+00]]
```

< Homography matrix for left_0 image>





< Undistorted right_0 image >   < Warped right_0 image >



```
Homography matrix for right image
[[ 2.38309772e-01  1.89969850e+00  1.67591824e+02]
 [-1.42729276e+00  5.63739445e+00  5.37016269e+02]
 [-1.59577235e-03  5.25374046e-03  1.00000000e+00]]
```

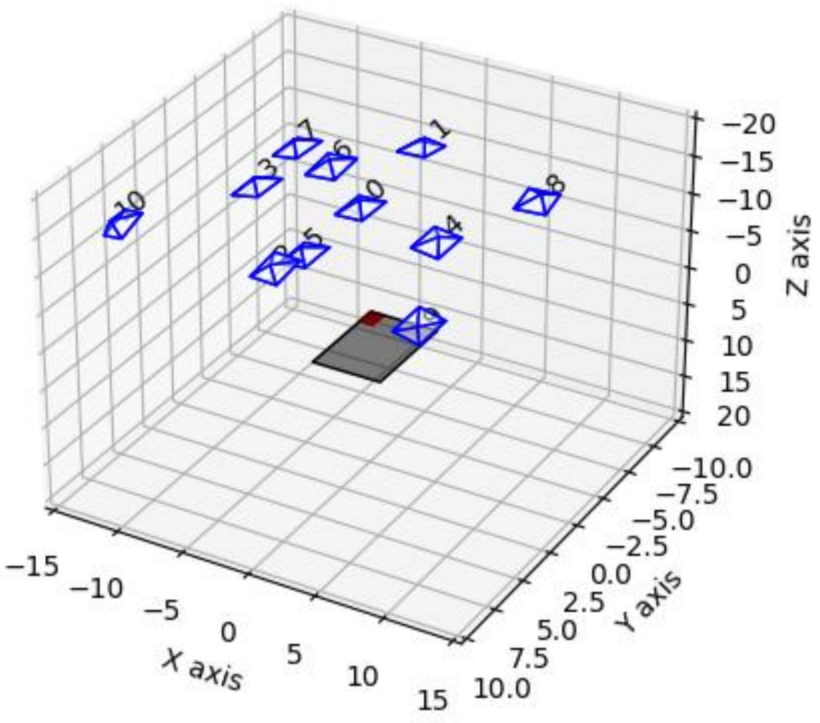
< Homography matrix for right_0 image >

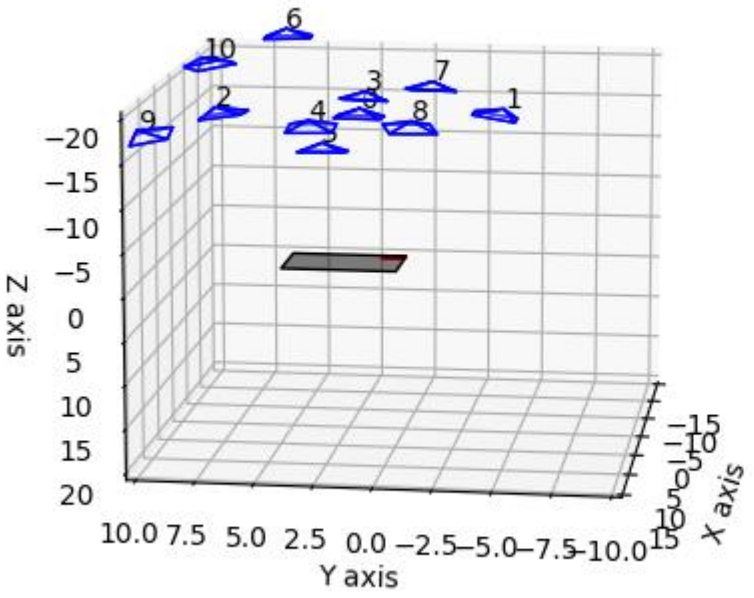
The warpPerspective() library function applies a perspective transformation to images. This function transforms the src(source image) using the following matrix equation

$$\texttt{dst}(x,y) = \texttt{src}\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right)$$

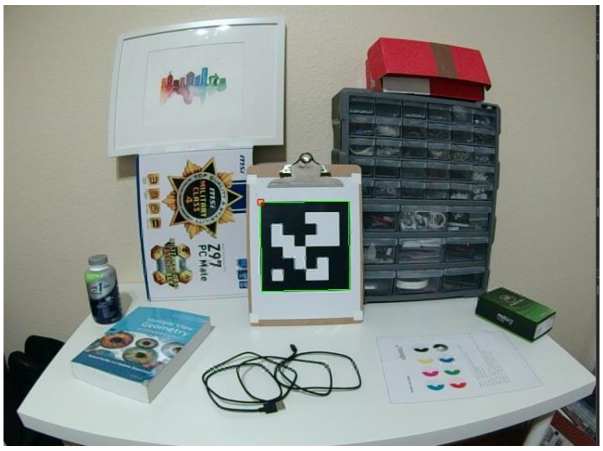
The parameters are src: input image, dst : output image that has the size dsize and the same type as src , M : 3 X 3 transformation matrix, dsize : size of the output image and has a flag WARP_INVERSE_MAP if set. Otherwise, the transformation is first inverted and put in the formula instead of M which is transformation matrix.

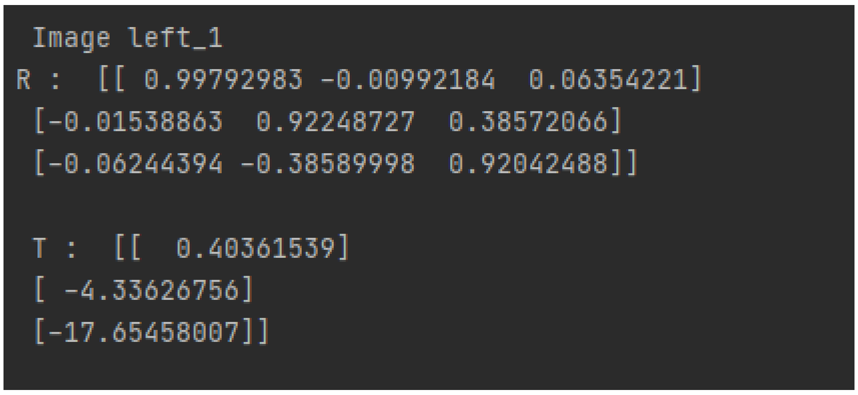
Task 6: Camera Pose Estimation from a Single View



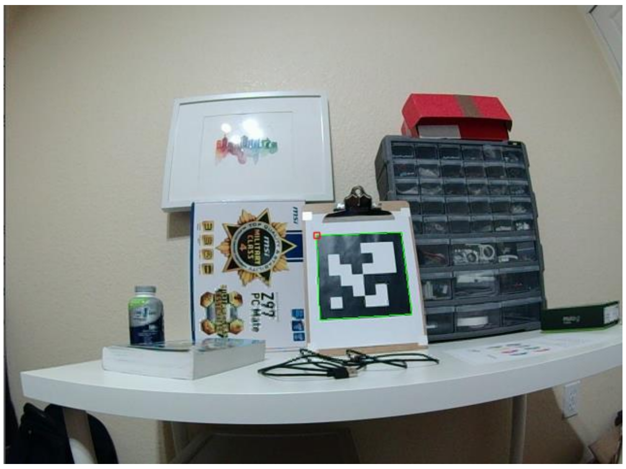


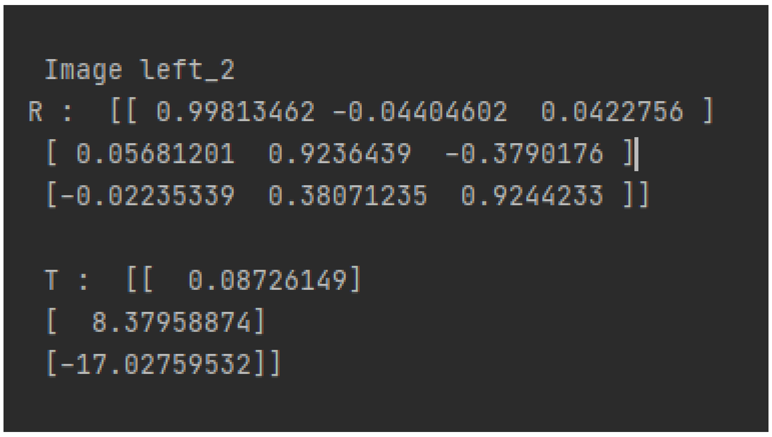
< The pose of at least 5 views using five different images taken by the same camera, similar to Figure 6. >



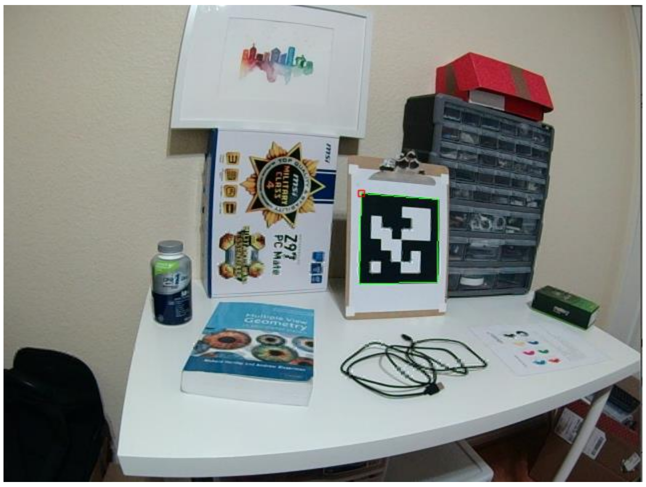


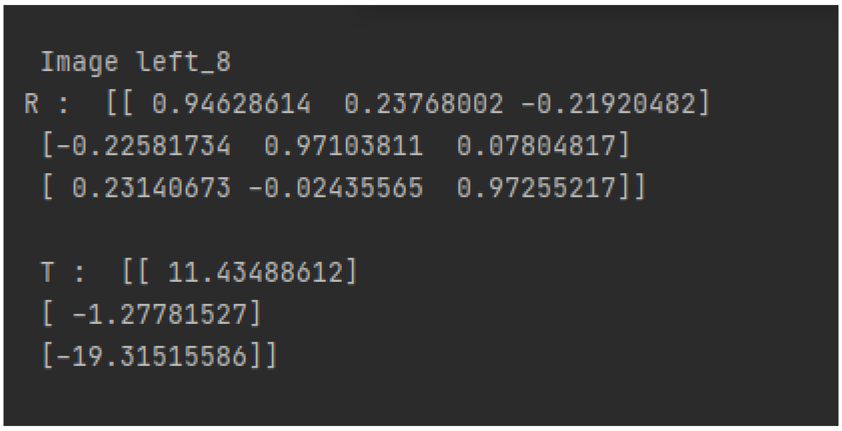
```
 Image left_1
R :  [[ 0.99792983 -0.00992184  0.06354221]
 [-0.01538863  0.92248727  0.38572066]
 [-0.06244394 -0.38589998  0.92042488]]

 T :  [[  0.40361539]
 [ -4.33626756]
 [-17.65458007]]
```



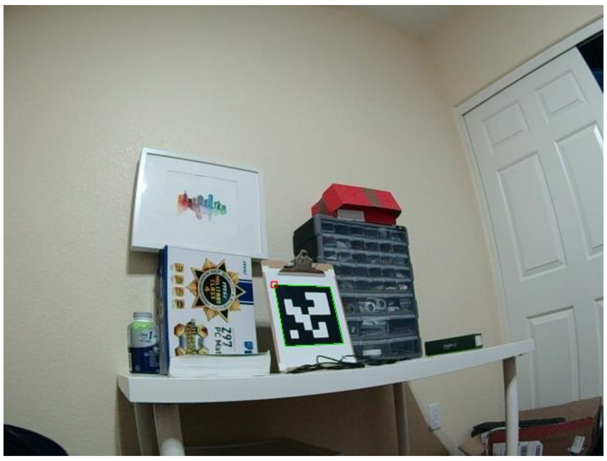


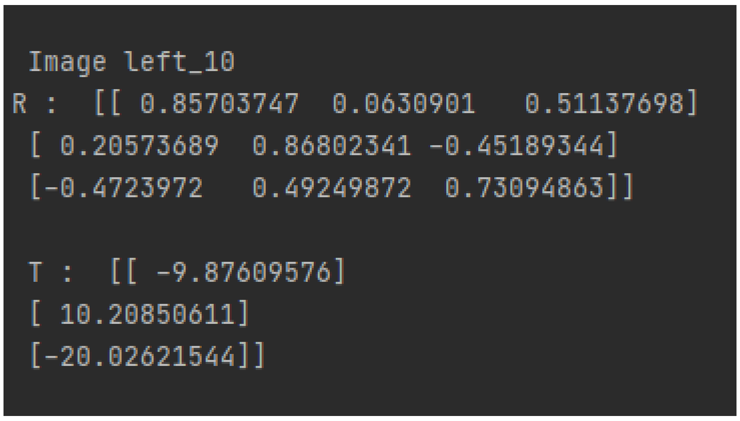
```
 Image left_2
R :  [[ 0.99813462 -0.04404602  0.0422756 ]
 [ 0.05681201  0.9236439  -0.3790176 ]
 [-0.02235339  0.38071235  0.9244233 ]]

 T :  [[  0.08726149]
 [  8.37958874]
 [-17.02759532]]
```



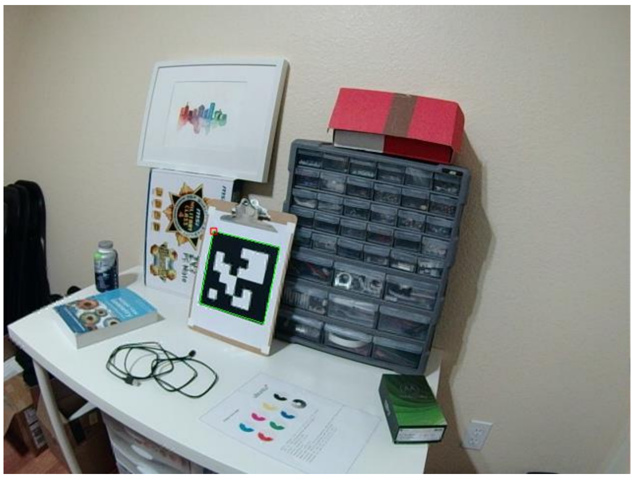


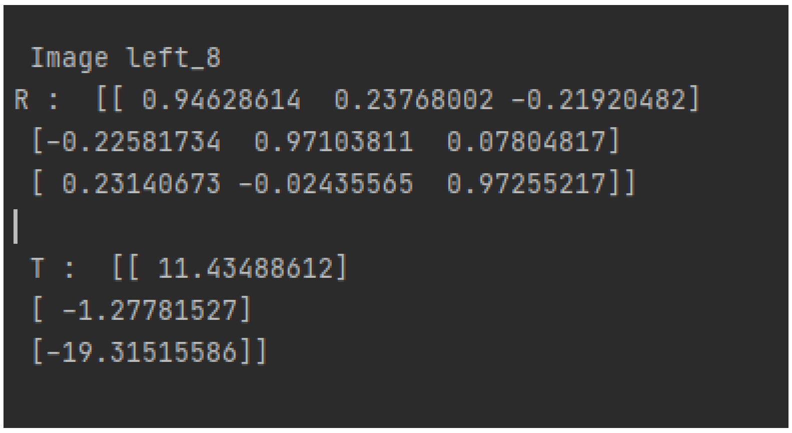
```
 Image left_8
R :  [[ 0.94628614  0.23768002 -0.21920482]
 [-0.22581734  0.97103811  0.07804817]
 [ 0.23140673 -0.02435565  0.97255217]]

 T :  [[ 11.43488612]
 [ -1.27781527]
 [-19.31515586]]
```



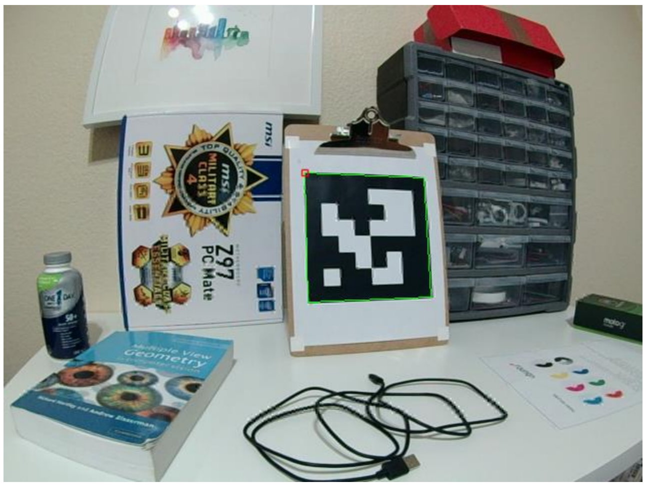


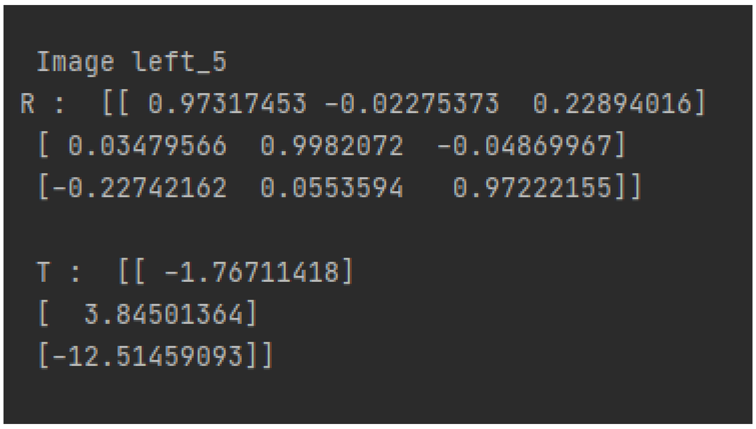
```
 Image left_10
R :  [[ 0.85703747  0.0630901   0.51137698]
 [ 0.20573689  0.86802341 -0.45189344]
 [-0.4723972   0.49249872  0.73094863]]

 T :  [[ -9.87609576]
 [ 10.20850611]
 [-20.02621544]]
```





```
 Image left_8
R :  [[ 0.94628614  0.23768002 -0.21920482]
 [-0.22581734  0.97103811  0.07804817]
 [ 0.23140673 -0.02435565  0.97255217]]

 T :  [[ 11.43488612]
 [ -1.27781527]
 [-19.31515586]]
```





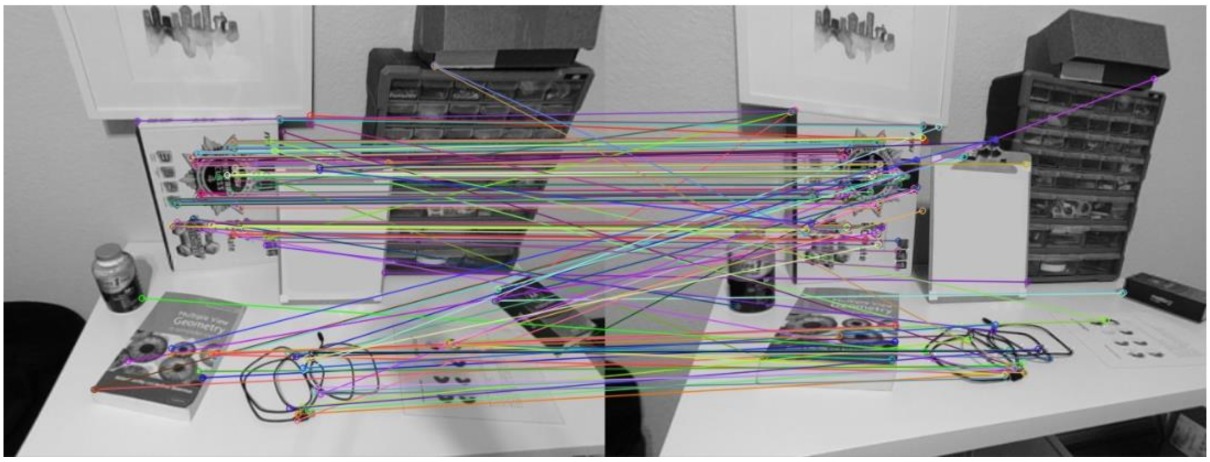
```
 Image left_5
R :  [[ 0.97317453 -0.02275373  0.22894016]
 [ 0.03479566  0.9982072  -0.04869967]
 [-0.22742162  0.0553594   0.97222155]]

 T :  [[ -1.76711418]
 [  3.84501364]
 [-12.51459093]]
```

< Calculated pose results (R matrix and t vector for each view >

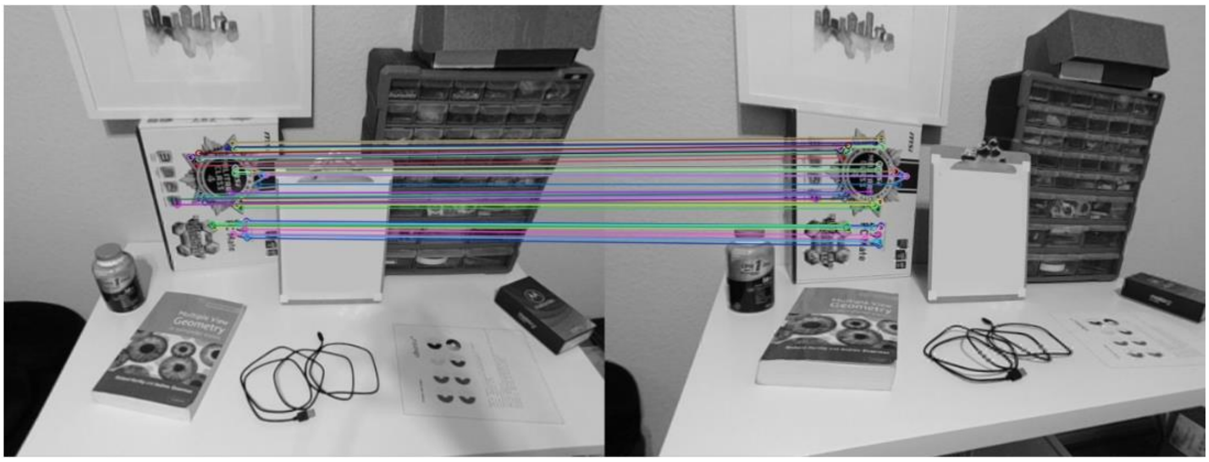
Solvepnp() function returns the rotation and the translation vectors that transform a 3D point expressed in the object coordinate frame to the camera coordinate frame. The parameters are objectpoints, imagepoints, input camera intrinsic matrix, input vector of distortion coefficients etc. This function estimates the object pose given object points and their corresponding image projections, as well as camera intrinsic matrix & distortion coefficients.

Task 7: Camera Pose Estimation and Reconstruction from Two Views

Matching results of at least two pairs of images

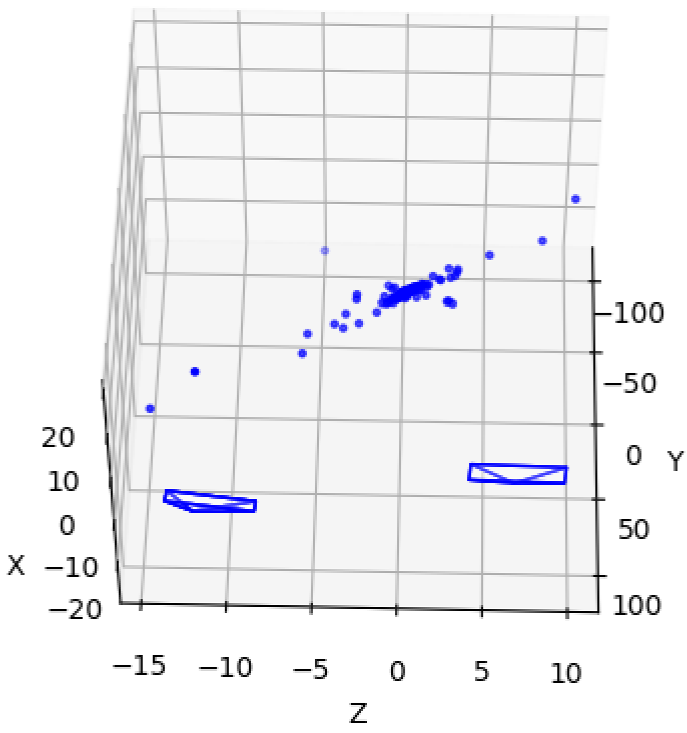


< An example of matches of feature points on the two views >

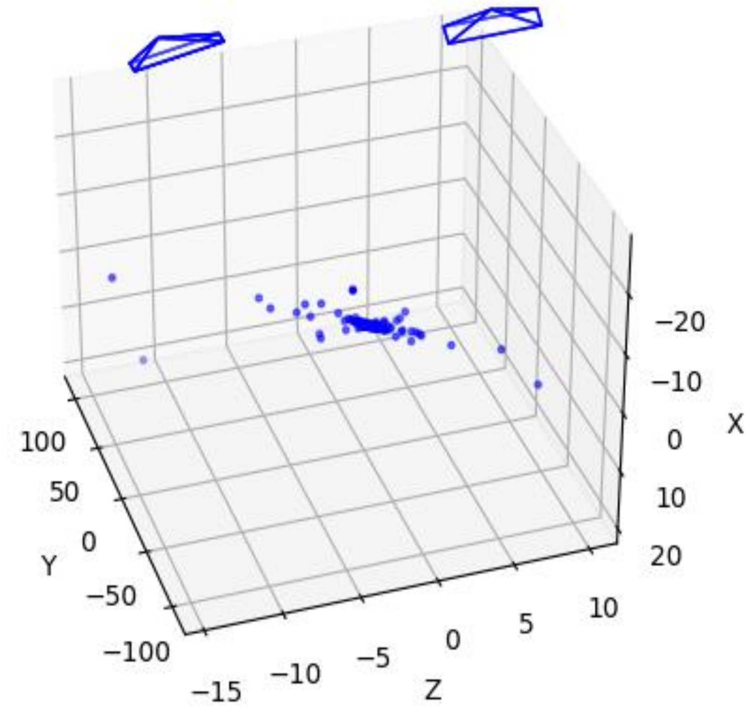


< An example of inliers of matched feature points by calculating the essential matrix with RANSAC >

3D point



3D point



findEssentialMat() – This function calculates an essential matrix from the corresponding points in two images. The parameters take porints1 from the first image, points2 from the second image as well as same size and format of points 1 and the camera intrinsic matrix.

rcoverPose() – This function recovers the relative camera rotation and the translation from an estimated essential matrix and the corresponding points in two images. The parameters take the input essential matrix, points from the first image , points2 from the second image, the camera intrinsic matrix, output rotation matrix, output translation vector and input and output mask for inliers in points1 and points2.