Important Note: Submit all your codes and the resulting images you used in the lab to SUCourse as a single zip file. Deadline for submission to SUCourse is **until the end of the lab**.

Things to do:

- In this lab we will recover the pose of a camera from two images taken from different viewpoints by estimating the Essential Matrix (E) using 8-point algorithm. To do that we will make use of epipolar geometry. First run "`lab8.m`" file to generate 3D world points and obtain their corresponding image points from two different views (Figure 1).

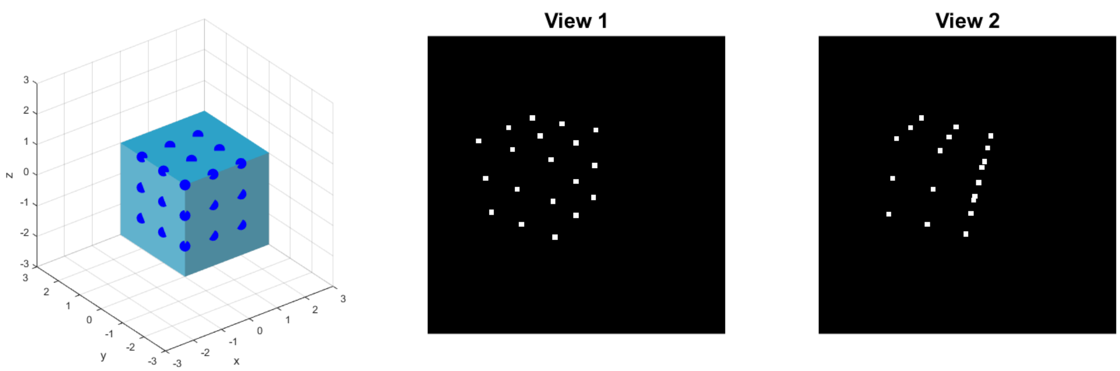

Figure 1: 3D world points and their corresponding image points from two different views

- Note that the matrix K which includes intrinsic camera parameters is already known (calibrated two-view problem). Thus, start your implementation by transforming points in pixel coordinates in both views by multiplying them with the inverse of K matrix.

- Initially, select 8 point pairs properly to avoid degenerate configuration for estimating a reliable Essential Matrix. For each point pair, obtain the vector a appropriately and stack 8 of them in the rows of X matrix as follows:

$$a = [x_1x_2 \quad x_1y_2 \quad x_1z_2 \quad y_1x_2 \quad y_1y_2 \quad y_1z_2 \quad z_1x_2 \quad z_1y_2 \quad z_1z_2]^T \tag{1}$$

$$X = [a_1^T;\ a_2^T;\ \ldots\ a_8^T] \tag{2}$$

- Since $XE^s = 0$, where E^s is the stacked version of Essential Matrix, the eigenvector associated with the smallest eigenvalue of X^TX gives the solution in stacked form. Now, obtain the Essential Matrix E and cure it to satisfy Essential Matrix Characterization theorem, and obtain the normalized essential matrix.

  Hint: Replace the singular values of the estimated E with $\sigma_1 = \sigma_2 = 1, \sigma_3 = 0$ and verify that $U, V \in SO(3)$ for $E = U\Sigma V^T$.

- Next, find the epipoles (e_1, e_2) and the coefficients of epipolar lines for a selected point pair. Then, **verify** that the image point and the epipole for one view are on the epipolar line.

  Hint: To find epipoles, you can use `null(E)` function, which gives the nullspace from the right.

- Finally, recover the rotation and the translation of the camera. Possible poses are recovered as follows:

$$(\hat{T}_1, R_1) = (UR_z(\frac{\pi}{2})\Sigma U^T, UR_z^T(\frac{\pi}{2})V^T) \qquad (\hat{T}_2, R_2) = (UR_z(-\frac{\pi}{2})\Sigma U^T, UR_z^T(-\frac{\pi}{2})V^T) \tag{3}$$

Your results should look like as follows:

```
True E =                                 Estimated E =
         0   -1.0000         0              0.0025   -0.3198    0.0025
   -0.3615         0   -3.1415             -0.1249    0.0035   -0.9922
         0    3.0000         0             -0.0059    0.9475    0.0054

True Rc2c1 =                             Estimated Rc2c1 =
    0.9063         0   -0.4226              0.9001    0.0037   -0.4356
         0    1.0000         0             -0.0023    1.0000    0.0038
    0.4226         0    0.9063              0.4356   -0.0024    0.9001

True Tc2c1 =                             Estimated Tc2c1 =
     3                                      0.9475
     0                                      0.0041
     1                                      0.3198
```

Post Lab

- Select 8 points on the same plane and recover the pose of the camera by estimating the essential matrix. Repeat the same procedure for each plane.
- Now select all the points and recover the pose of the camera by estimating the essential matrix.

Provide the images of two views for each case and compare the results. Explain all the procedure that you follow. Comment on which recovered pose would you pick. Discuss your results and verifications.

Deadline for post lab report submission to SUCourse: **17 December 2018, 23:55**.