Q1.

The dataset Obtained from the scene:
X-homogenous coordinates of the 3D World coordinates of control points
x-their respective homogenous Image coordinates

```
X =

    2.0000    3.0100    1.0000   -0.0100   -0.0100    0.0100         0    2.0000
         0         0         0    2.0000    3.0000    2.0000    3.0000         0
    5.0000    3.0000    7.0000    2.0000    5.0000    6.0000    8.0000    1.0000
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000


x =

   1.0e+03 *

    2.0010    2.3130    1.7309    1.0352    0.7654    1.0421    0.7753    2.0219
    2.0982    1.4235    2.6934    1.1214    2.1316    2.4390    3.1496    0.7849
    0.0010    0.0010    0.0010    0.0010    0.0010    0.0010    0.0010    0.0010
```

The transformation matrices T and U mentioned in the question for the above dataset is :

```
T =

    0.0015         0   -2.2634
         0    0.0015   -3.0686
         0         0    1.0000


U =

    0.6324         0         0   -0.6324
         0    0.6324         0   -0.7905
         0         0    0.6324   -2.9247
         0         0         0    1.0000
```

WHY NORMALISATION ?:

The matrix M will contain entries xi , yj - the image coordinates. Since xi and yi are measured in pixels (as we can see from above) the values are in thousands. But the third homogeneous coordinate is 1 and therefore the matrix M contains coefficient of highly varying magnitude. This can make the matrix Trans(M)*M poorly conditioned resulting buildup of numerical errors. The numerics can be greatly improved by translating the coordinates such that their “center of mass” is zero and then rescaling the coordinates to be 1. Data normalisation improves the accuracy of the results ensuring that the algorithm will be invariant to arbitrary choices in scale and coordinate frame.

Following the steps in the DLT method as given in the slides we get the intrinsics and extrinsics.

```
X0 =

    9.4165
   10.5695
    4.2853


K =

 -151.7633 -118.1228 -140.1675
   71.1600 -223.6864   54.2658
   -0.0313   -0.0290    0.0290


R =

    0.0856    0.7939    0.6020
    0.0793    0.5969   -0.7984
   -0.9932    0.1160   -0.0119
```

The RMSE is calculated in cm.

```
RMSE =

    0.0350
```

The RMSE (in cm) obtained is 0.0350.This value is a little high than expected because of manual errors occurred while collecting the dataset for image points.

Note:

The same algorithm was run for another dataset obtained online.The RMSE obtained for that dataset is of the order 10^-6.So there is a lot of manual error occurred in the dataset while the points were observed on the image.

The RMSE of the points is calculated using the standard formula $RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$ wheredistances are used instead of y.

BLUE - ORIGINAL CONTROL POINTS;RED - OBTAINED

BLUE - ORIGINAL CONTROL POINTS;RED - OBTAINED

SHOWING BOTH THE ORIGINAL POINTS AND OBTAINED POINTS:

BLUE - ORIGINAL CONTROL POINTS;RED - OBTAINED

$^{O}B_{J}$

Reference : https://www.mathworks.com

The code already contains the dataset stored in the variables.Just run the myMainScript.m file.

getImagePoints.m is a helper file that was used to get the image points (not nneeded for thee mainscript to run).