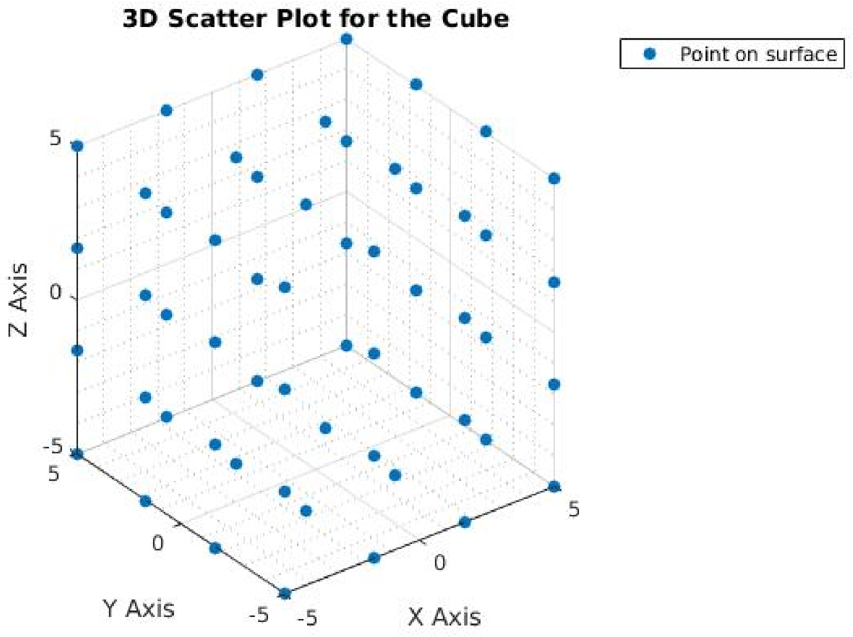
Moile Robotics
Assignment 1

Gokul B. Nair - 201502034
Pranav Kamojjhala - 201501213

1 Solution to Part 0:

The script for the required implementation is provided in the file 'a01.m' as required. The function is written in the script 'generateCube.m'. The final 3D scatter plot is obtained as the following:



The figure above shows the result of the function generateCube(4,10) with the same parameters.

2 Solution to Part 1:

The script is saved as 'a02.m' as required. Given the initial position of the camera as (25,0,0), we can obtain the remaining seven positions by rotating this point about the z-axis by 45°, each time. These positions are stored as a 3X8 matrix 'cam_pos'. We obtain this matrix for the given situation as:

$$\begin{bmatrix} 25.0 & 17.67 & 0.00 & -17.67 & -25.0 & -17.67 & 0.00 & 17.67 \\ 0.00 & 17.67 & 25.0 & 17.67 & 0.00 & -17.67 & -25.0 & -17.67 \\ 0.00 & 0.000 & 0.00 & 0.000 & 0.00 & 0.000 & 0.00 & 0.000 \end{bmatrix}$$

Initially, The camera coordinates are such that its z-axis points towards the world origin along the world x-axis, that is, into the scene. The camera y-axis points perpendiculary upwards and the camera x-axis points to the right. We have taken right hand system here. Hence the rotations carried out in the right order to transform a point from world coordinates to camera coordinates are as follows:

- About X-Axis in clockwise direction by 90°.
- About new Y-Axis in clockwise direction by 90°.

After these, the required rotation 'θ' and translation 'T' is appllied to the camera to obtain the 8 required positions. We obtain the final rotation matrix as:

$$\begin{bmatrix} sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \\ cos(\theta) & -sin(\theta) & 0 \end{bmatrix}$$

The translation matrix 'T' is obtained as:

$$\begin{bmatrix} 0 \\ 0 \\ -25 \end{bmatrix}$$

The final Homogenous Transformation Matrix is obtained as:

$$\begin{bmatrix} sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ cos(\theta) & -sin(\theta) & 0 & -25 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The above matrix can be represented generally as:

$$\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

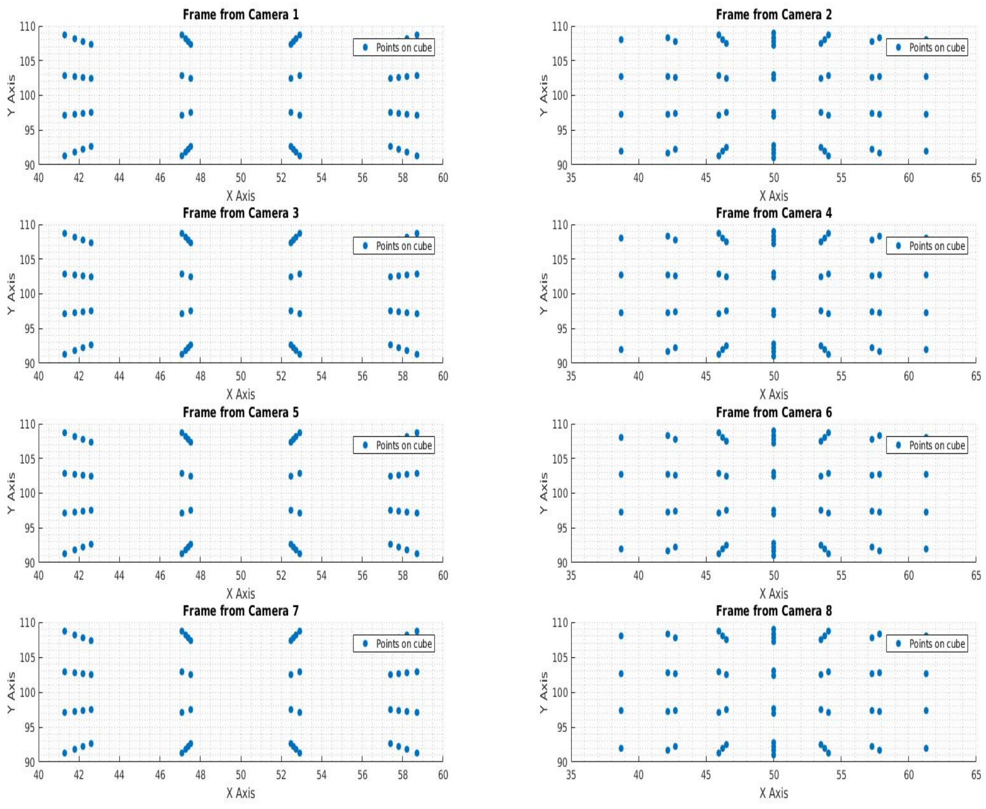
Here, 'R' is the 3X3 rotation matrix, 'T' is the 3X1 translation matrix, '0' is the 1X3 zero matrix and 1 the scalar. World coordinates each point on the cube generated is converted to corresponding camera coordinates using the equation:

$$cam\_coordinates = R * world\_coordinates + T$$

The frame coordinates for each camera for each point on the cube can be obtained by the equation:

$$frame\_coordinates = K * cam\_coordinates$$

Here, 'K' is the given intrinsic parameter of the camera used. We obtain the 2D scatter plots from all the 8 views as follows:



3 Part 3: Camera Calibration Toolbox

The functions used here are as follows:

- 'a03.m' contains the main script.
- 'homo.m' to get the homographies.
- 'getv.m' to generate 'V' matrix.
- 'getb.m' to get 'B' while solving for VB=0.
- 'nearestSPD.m' to get the nearest positive definite matrix so that we can find the proper cholesky decomposition. (Third Party Function)

The points taken for reference in each image are as follows:

- image1 - (0,0),(1,1),(2,1),(2,2),(3,2).
- image2 - (0,0),(1,1),(2,1),(3,1),(3,3).
- image3 - (0,0),(2,0),(2,1),(3,1),(4,1).

3.1 Accuracy

We have used reprojection errors by taking the pattern key points and the world points of the corresponding key points and checking the distances.