**CV practical assignment final report**

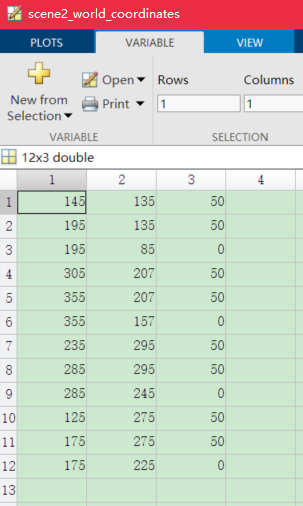
**Group G**

**Memberships:** **Tuo Yang (0589715) Yordanos Alemu (0592896) Subhashree Rautray(0592964)**

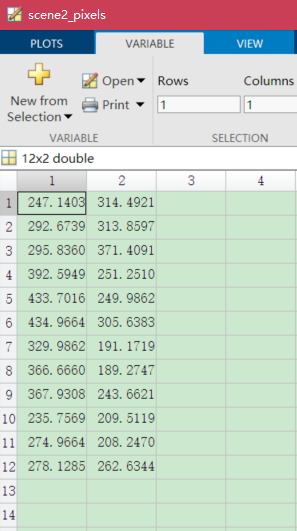
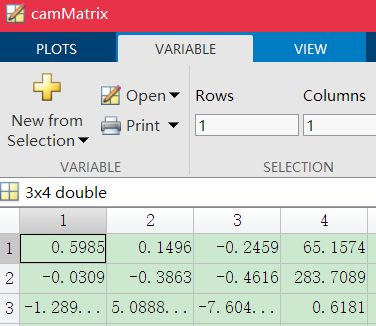
Here we will separately discuss about the main steps of finishing this practical assignment, the steps can be divided into the following parts: camera calibration, back-projection, and localization of cubes in the robot frame. Detailed explanations and experiments result presentation will be provided in the following contents.

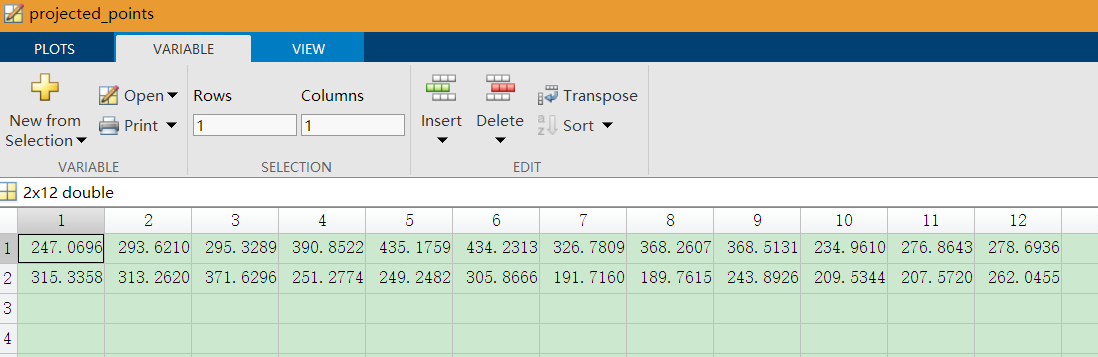
1.**Camera Calibration** (person in charge:Yordanos Alemu)

The main goal is to take advantage of image coordinates and their corresponding world coordinates to get the camera calibration matrix. Here the points we take for calibration are cube’s corner coordinates, more details about picking points will be introduced in the second part. By finding their image and world coordinates, we can get the projection matrix between the world frame and the image frame. Coordinates of the world frame and the image frame we take are shown as follows:

The world coordinates (taken from the image 8130478-2021-04-07-100838-l.png from the second image set)

Their corresponding pixel coordinates (taken from the image 8130478-2021-04-07-100838-l.png from the second image set)

For more details of calibration algorithm, please check the coding file calibrate.m. Here we simply describe the its process. Firstly, doing normalization to image points and world points at the same time. Then using these two sets of points to build the A matrix. Finally, doing the SVD decomposition to the A matrix to get the U, S, V three matrixes. Taking the last column of matrix V to do the reshaping and set the normalization back, the final result we got is one 3 cross 4 matrix:

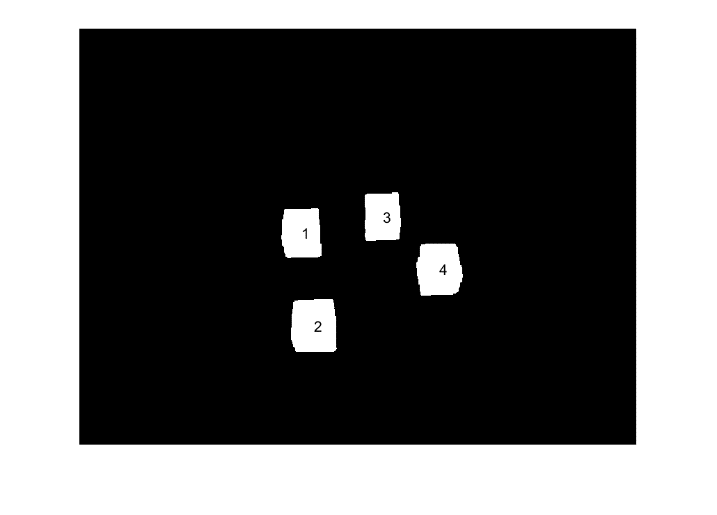
After getting the projection matrix, we can use world coordinates to do projection again, result will be estimated image points, the data of which are shown as follows:

Here we can make comparison between the reprojected pixel coordinates and original coordinates, it can be found that these two sets of points are almost the same but with little errors.

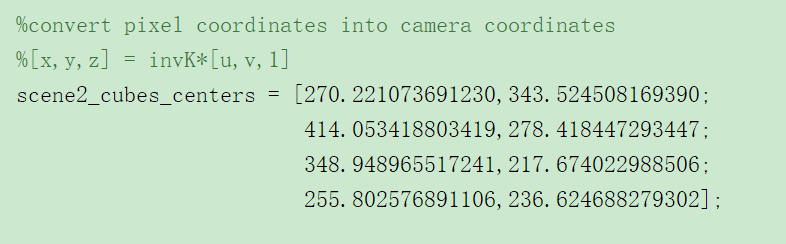
2.**Back Projection and Points Extraction** (person in charge: Tuo Yang and Subhashree Rautray)

The points extraction here refers to extract pixel coordinates and world coordinates for the calibration and back projection task. For the calibration task, like we said in the last part, we extract points from the image 8130478-2021-04-07-100838-l.png from the second image set. The pixel coordinates we extracted are shown as follows:

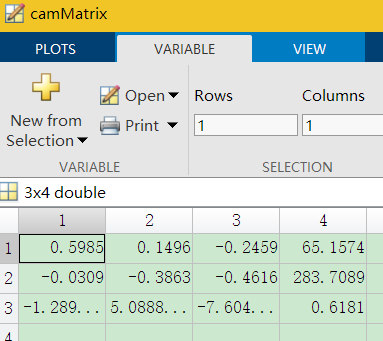
The blue points here will be used as the pixel coordinates for the calibration task, which are picked by using matlab function ginput, their corresponding world coordinates are acquired with calculations by using cube mass centers coordinates in the world frame and cube edge widths.

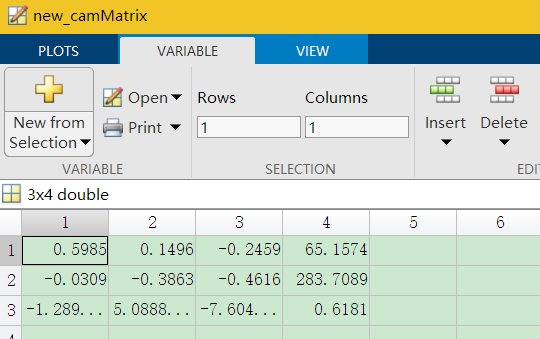
The red points are mass center’s locations in the image frame, acquired by doing the segmentation to this image using RGB or HSV color space. Currently-presented image’s segmentation results are shown as below:

All mass centers in the image frame are marked with numbers, for more segmentation results from second set of images, please check the folder cubes-coords-segmentation.

The mass center coordinates we got in the image frame are:

Then we design the back-projection algorithm, generally it is the conversion from image coordinates to world coordinates, to start with, we need to do decomposition to projection matrix to get three parameters: K (intrinsic matrix), R (rotation matrix), C (camera center)

Projection matrix before decomposition:

Resumed projection matrix after using the formula: M = K\*[R- R\*C]

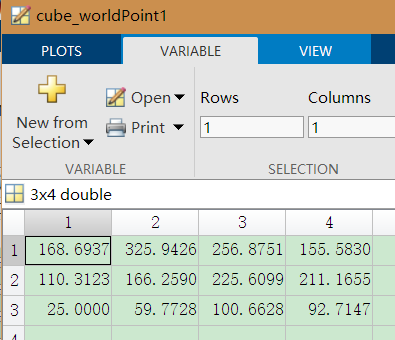
the designed back projection solution is based on this formula:

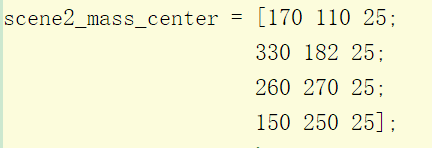


where [u, v] are pixel coordinates, s is the depth, s\*[u v 1] are camera frame coordinates, M is the projection matrix, R is the rotation matrix, t is the translation vector, gained by using the formula t = -R \* C. [X Y Zconst] are world coordinates

We do the transform to the last formula; world coordinates can be got by this:

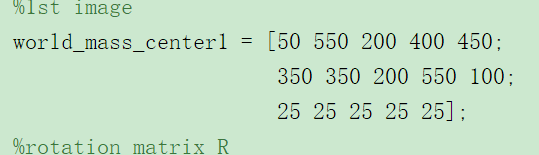


After using the formula below to do the back-projection, the mass centers coordinates we got in the world frame is:

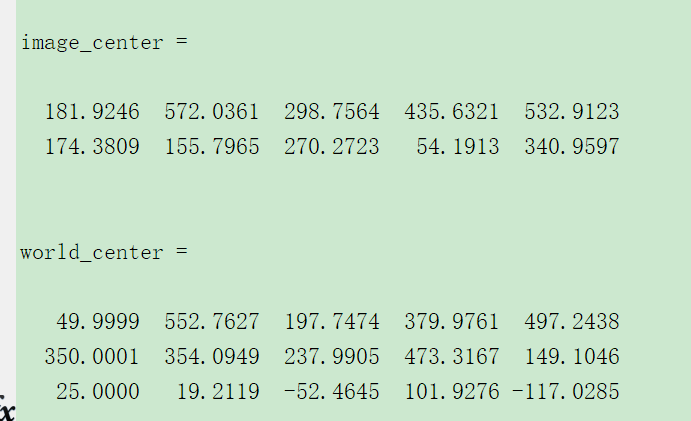
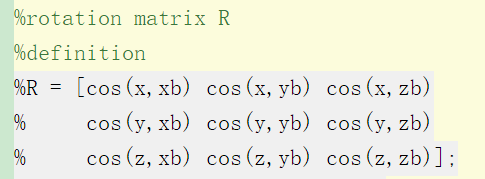
Making comparison with the real mass center coordinates in the world frame:

3. Robot localization (person in charge: Tuo Yang and Subhashree Rautray)

Now we can use the third set of images to do the projection and back-projection process,

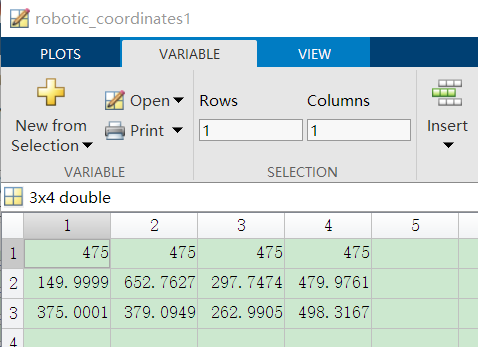
Here we take the image 8130478-2021-04-07-090247-l.png as the example, its mass center coordinates in the world frame are:

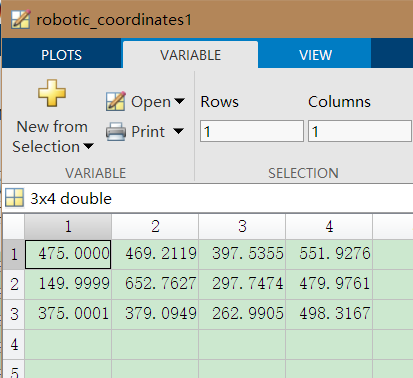
After doing projection and back-projection to these world points, its corresponding pixel coordinates the reprojected world points coordinates are:

Now we need to design the solution of coordinates transformation between world frame and robot frame, transformation matrix could be defined as [R t;0 0 0 1], where R is the rotation matrix, defined as:

Because the transformation happens between two coordinates system, the world frame [x y z] and the robot frame [xb yb zb], so the cos(x, xb) means the cosine value of intersection angle between the world frame x axis and the robot frame x axis，and so on for other elements, Translation vector here we directly take arrow cube coordinates in the world frame [450 100 25]. Therefore, the conversion between world coordinates and robot coordinates can be done by:

PR= RTW\*PW

Where PR is homogenous coordinates in the robot frame, PW is homogenous coordinates in the world frame, RTW is the transformation matrix, we fix the z axis as 25 to reprojected world points, the projected robot coordinates are:

If we don’t fix the z axis as 25 to reprojected world points, the projected robot coordinates are: