Assignment4

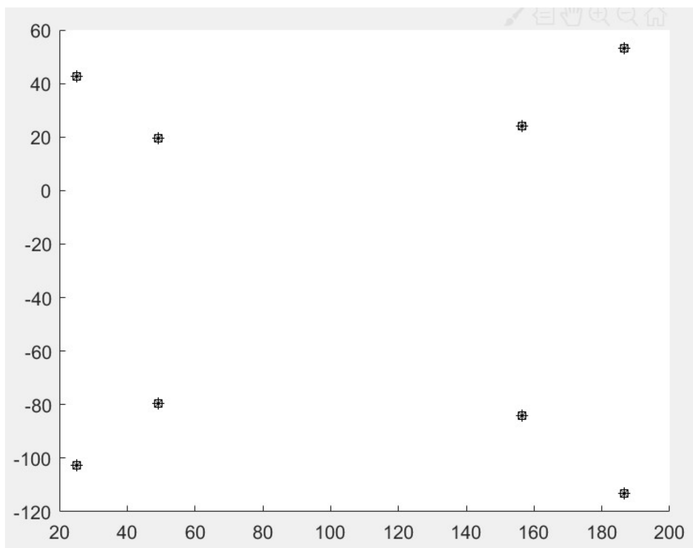
Jingyuan Wang

Question1

For this question, the least squares function for estimating P matrix is implemented in calibrate.m. Basically, build up the matrix A as followed and use SVD function to find the V with smallest singular value.

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ \vdots \\ \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N \end{bmatrix}$$

The results of Q1_Tester.m as well as Q1.m are presented.



c1.jpg

c2.jpg

Question2

When comparing image shifting results of changing K and R matrices, I constructed a matrix

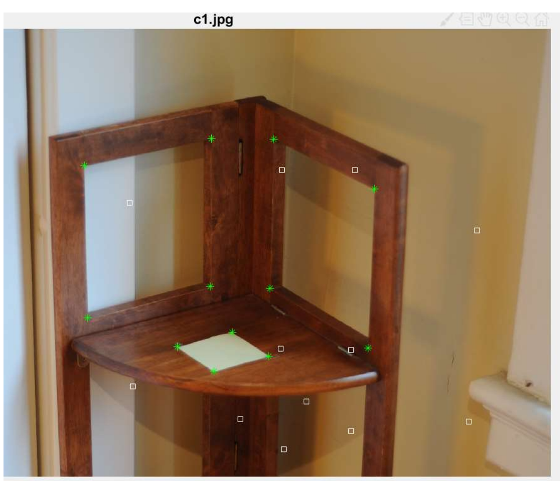
$$M1 = \begin{bmatrix} 1 & 0 & 200 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

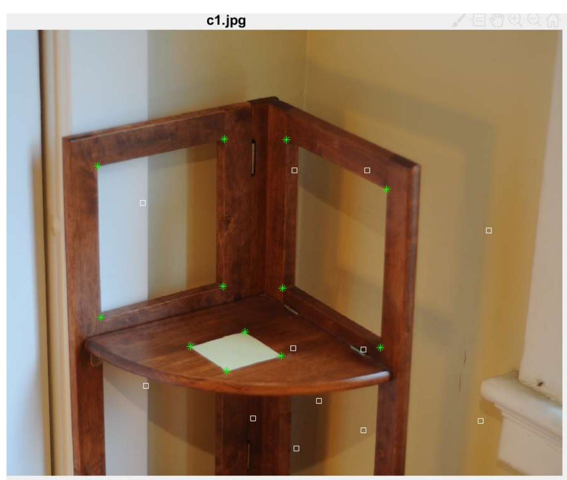
for changing matrix K, which represents to translate the image 200 pixels to the right. After adjusting to R transformation matrix, a matrix M2 was found to change R.

$$M2 = \begin{bmatrix} 1 & 0 & 0.033 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then use this new K matrix and R matrix to build up new P matrices separately and find corresponding points in c1.jpg. As displayed below, the results are approximately the same.

In the same way, a matrix of [1,0,0,-33;0,1,0,3;0,0,1,68;0,0,0,1]; was applied to vector C to get similar transformations for shifting K with 200 pixels and [1.05,-0.07,-0.071;-0.04,1.18,-0.022;-0.06,-0.03,1.17] was applied to [I|-C] for expanding the image to 1.2 times of original size.





Please check code in Q2.m for details. Different P matrices are already coded up but to apply them to the scene, you need to change variable q for question a and b, and variable P for choices of P matrix.

Question3

A)Consider the image shifting case, so for a K matrix, we can apply an general transformation matrix to it.

$$\mathrm{M1} = \begin{bmatrix} 1 & 0 & s \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

Which makes:

$$\mathrm{Knew} = \begin{bmatrix} fm_x & 0 & p_x + s \\ 0 & fm_y & p_y + t \\ 0 & 0 & 1 \end{bmatrix}$$

Indicating the image translates with(s,t). So to find a transformation(M2) for R matrix, we have:

$$\mathrm{M1} * \mathrm{K} * \mathrm{R} * [\,\mathrm{I}\,|-\mathrm{C}] = \mathrm{K} * \mathrm{M2} * \mathrm{R} * [\,\mathrm{I}\,|-\mathrm{C}]$$

i.e.,

$$\mathrm{M2} = \mathrm{inv}(\mathrm{K}) * \mathrm{M1} * \mathrm{K}$$

which makes new transformation P with M1*K, R, C and K, M2*R, C get the same result. And

$$\Delta\mathrm{u} = \mathrm{s} * \mathrm{w}/\mathrm{f}m_x$$

To get an intuition of this, shifting an image can be obtained by scaling the u,v,w unit vectors of R matrix.

Similarly, equal transformation M3 for the third part satisfies:

$$\mathrm{M1} * \mathrm{K} * \mathrm{R} * [\,\mathrm{I}\,|-\mathrm{C}] = K * R * M3 * [\,I\,| - C]$$

Which means:

$$\mathrm{M3} = \mathrm{inv}(\mathrm{R}) * \mathrm{inv}(\mathrm{K}) * \mathrm{M1} * \mathrm{K} * \mathrm{R};$$

So shifting an image, we can change the position of camera along the translating axis. If we want the image shifts to the right for some distance, it has equal effect with camera moving to the left.

B)Construct a new matrix M1 for applying to K and get image expansion effect.

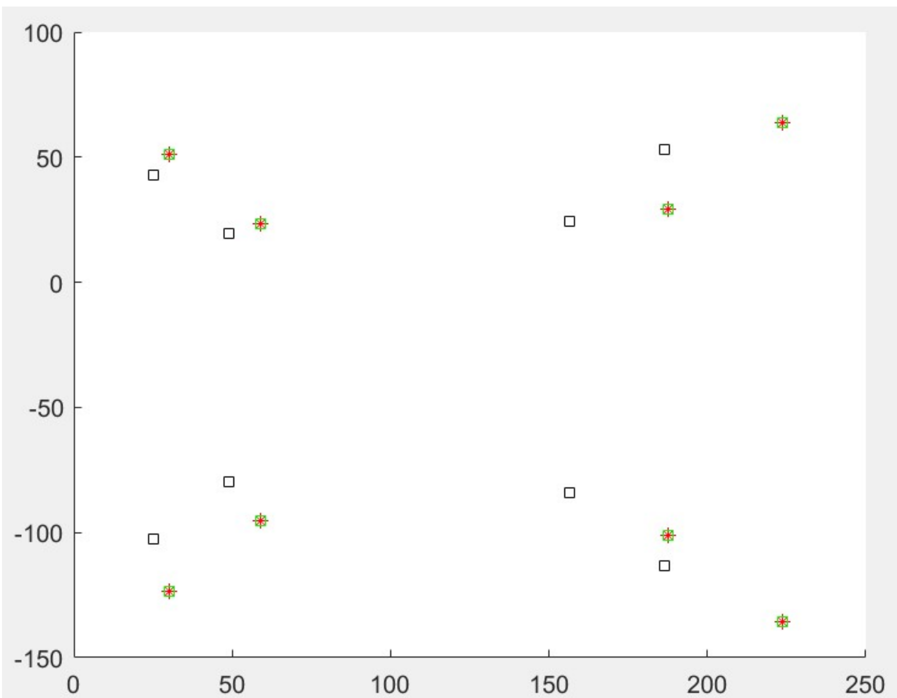
$$M1 = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So change for [I|-C] , matrix M2 still satisfies

$$\mathrm{M2} = \mathrm{inv(R)} * \mathrm{inv(K)} * \mathrm{M1} * \mathrm{K} * \mathrm{R};$$

To expand the image, we can increase focal length in K matrix is equal to move the camera forward along the positive z axis in the left-hand coordinates.

In Q3.m, I constructed the different transformations as discussed and get the same result of the Q1Tester scene. As the red points refer to image expansion through K and green ones refer to image expansion through C.



Question4

In this question, I constructed K matrices for camera 1 and 2. And with a 3d rotation matrix ,

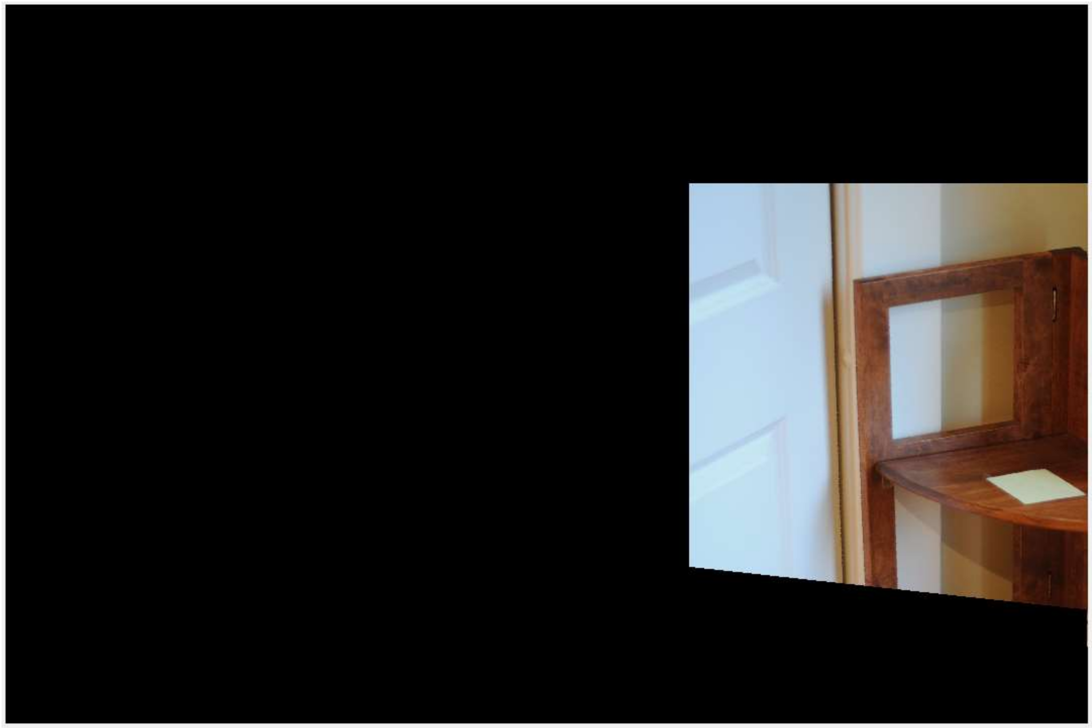
We can easily get the relationship between the projection of a 3d points to a pixel in scene 1 and scene 2. As coded in Q4.m,

$$\begin{bmatrix} i_2 \\ j_2 \\ 1 \end{bmatrix} = K2 * Rotation * K1 * \begin{bmatrix} i_1 \\ j_1 \\ 1 \end{bmatrix}$$

Below are the output with required rotation angle theta:

Theta = 0

Theta = 10



Theta = 20