Report

Omkar. A. Chittar

Problem 1. Calibrate the Camera using the following image:

1. What is the minimum number matching points to solve this mathematically?
2. What is the pipeline or the block diagram that needs to be done to calibrate this camera given the image above.
3. First write down the mathematical formation for your answer including steps that need to be done to find the intrinsic matrix K.
4. Find the P matrix.
5. Decompose the P matrix into the Translation, Rotation and Intrinsic matrices using the Gram–Schmidt process and compute the reprojection error for each point.

1. We need a minimum of 6 matching points to solve the above problem mathematically since each point produces two equations and the projection matrix is 11 DOF.

2. Pipeline:

1. Import the req libraries - **NumPy**, **SciPy**.
2. Define image points and world points as input.

```
world_pts = [[0,0,0,1], [0,3,0,1], [0,7,0,1], [0,11,0,1], [7,1,0,1], [0,11,7,1], [7,9,0,1], [0,1,7,1]]
image_pts = [(757,213), (758,415), (758,686), (759,966), (1190,172), (329,1041), (1204,850), (340,159)]
```

3. Create a **homogenous system of linear equations** using the image points and world points.

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & & & \vdots & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix}$$

4. Use Singular Value Decomposition (**SVD**) on the **homogeneous** matrix to calculate the projection matrix. We obtain three matrices (U, S, V^T) from the operation.
   a. The **last row** entries of the $\mathbf{V^T}$ matrix obtained represent the values of the **projection matrix P**.
   b. Reshape these values to form a **3x4** matrix **P**.

```
Camera Projection Matrix:
 [[ 3.62233659e-02 -2.21521080e-03 -8.83242915e-02  9.54088881e-01]
 [-2.53833189e-02  8.30555704e-02 -2.80016309e-02  2.68827013e-01]
 [-3.49222322e-05 -3.27184809e-06 -3.95667606e-05  1.26053750e-03]]
```

5. Take the values from the first 3 rows and columns from the P matrix to form a **3x3** matrix **M**.

```
[[ 3.62233659e-02 -2.21521080e-03 -8.83242915e-02]
 [-2.53833189e-02  8.30555704e-02 -2.80016309e-02]
 [-3.49222322e-05 -3.27184809e-06 -3.95667606e-05]]
```

6. Carry out RQ decomposition on M using the function '**scipy.linalg.rq()**' where R represents the Camera Intrinsic matrix(K).

```
Camera Intrinsic Matrix:
 [[ 8.56060403e-02  1.00077564e-04  4.23062136e-02]
 [ 0.00000000e+00 -8.52363321e-02  3.25791294e-02]
 [ 0.00000000e+00  0.00000000e+00  5.28752857e-05]]
```

7. Calculate the extrinsic matrix (E) as the product of the inverse of the intrinsic matrix and the projection matrix.

```
Extrinsic Matrix:
 [[ 7.49486429e-01  5.87017071e-03 -6.61993681e-01 -6.43413492e-01]
 [ 4.53559043e-02 -9.98066421e-01  4.25001254e-02  5.95818335e+00]
 [-6.60464181e-01 -6.18785893e-02 -7.48303485e-01  2.38398239e+01]]
```

8. Extract the rotation and translation matrices from the extrinsic matrix. The Extrinsic matrix is of the following form:

$$[R \mid t] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right]$$

```
Rotation:
 [[ 0.74948643  0.00587017 -0.66199368]
 [ 0.0453559  -0.99806642  0.04250013]
 [-0.66046418 -0.06187859 -0.74830349]]

Translation:
 [-0.64341349  5.95818335 23.83982388]
```

9. Compute the projected points by multiplying the projection matrix with the world points in homogenous coordinates.
10. Compute the **reprojection error** by calculating the **Euclidean distance** between the **projected points** and the corresponding **image points**.
11. Calculate the average error by dividing the total error by the number of image points.
12. Output the projection matrix, intrinsic matrix, extrinsic matrix, rotation matrix, translation matrix, and average error.

3. Mathematical Formation:

1. Convert the coordinates into homogeneous form.

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & & & \vdots & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix}$$

2. Performing Singular Value Decomposition on this Homogeneous matrix to obtain the following matrices:

$$\boxed{C_{m \times n}} = \boxed{U_{m \times r}} \times \boxed{\Sigma_{r \times r}} \times \boxed{V^{T}_{r \times n}}$$

   a. Columns of **U** are left singular vectors.
   b. **Sigma** is a Diagonal matrix.
   c. Rows of $\mathbf{V^T}$ are right singular vectors.
3. The last row of $\mathbf{V^T}$ represents the elements of the camera projection matrix P.
   a. Reshape the matrix in the order 3x4.
4. Take the values from the first 3 rows and columns from the P matrix to form a **3x3** matrix **M**.
5. Use RQ decomposition on the M matrix.
   a. Using the Gram-Schmidt process to yield two matrices R and Q.

b. R is a upper triangular matrix and Q is a n orthogonal matrix.

$$Q = [\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_n]$$

and

$$R = \begin{bmatrix} \langle \mathbf{e}_1, \mathbf{a}_1 \rangle & \langle \mathbf{e}_1, \mathbf{a}_2 \rangle & \langle \mathbf{e}_1, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{e}_1, \mathbf{a}_n \rangle \\ 0 & \langle \mathbf{e}_2, \mathbf{a}_2 \rangle & \langle \mathbf{e}_2, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{e}_2, \mathbf{a}_n \rangle \\ 0 & 0 & \langle \mathbf{e}_3, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{e}_3, \mathbf{a}_n \rangle \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \langle \mathbf{e}_n, \mathbf{a}_n \rangle \end{bmatrix}$$

6. The matrix R is the camera intrinsic matrix(K).

4. P Matrix:

```
Camera Projection Matrix:
 [[ 3.62233659e-02 -2.21521080e-03 -8.83242915e-02  9.54088881e-01]
 [-2.53833189e-02  8.30555704e-02 -2.80016309e-02  2.68827013e-01]
 [-3.49222322e-05 -3.27184809e-06 -3.95667606e-05  1.26053750e-03]]
```

Results:

```
Reproj error for point  1  is:  0.28561276727805496
Reproj error for point  2  is:  0.9725828452229532
Reproj error for point  3  is:  1.036081784374865
Reproj error for point  4  is:  0.45408628677326207
Reproj error for point  5  is:  0.19089831889735914
Reproj error for point  6  is:  0.3189920832714891
Reproj error for point  7  is:  0.19594240534327106
Reproj error for point  8  is:  0.30829602844222703

Mean Error:
 0.47031156495043525
```