ENPM673

Perception for Autonomous Robots — PROJECT-3

17 April 2023

Vamshi Kalavagunta

119126332

vamshik@umd.edu

A. James Clark School of Engineering

University of Maryland College Park - 20742

1 Problem-1

Calibrate the camera (Find the intrinsic matrix K)

1. **What is the minimum number matching points to solve this mathematically?**
   The minimum number of matching points to solve this mathematically is 6.

2. **What is the pipeline or the block diagram that needs to be done in order to calibrate this camera given the image above?**
   Pipeline followed to calibrate this camera for the given image is below

   - **STEP 1:**Collect image and world point correspondences.
   - **STEP 2:**Compute the A matrix using these correspondences.
   - **STEP 3:**Compute the P matrix using the SVD method.
   - **STEP 4:**Compute the M matrix using the P matrix.
   - **STEP 5:**Decompose the M matrix into rotation and intrinsic matrices using the RQ factorization method.
   - **STEP 6:**Compute the reprojection error for each point using the intrinsic matrix and the world points.

3. **First write down the mathematical formation for your answer including steps that need to be done to find the intrinsic matrix K.**

   Here we perform camera calibration using a known set of 3D world coordinates and their corresponding 2D coordinates in the camera image. The following are the math formulas used in the code:

   - We are given image and world co-ordinates so we calculate P using the below equation

$$\begin{bmatrix} x_{image} \\ y_{image} \\ 1 \end{bmatrix} = \mathrm{P} \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{bmatrix}$$

   - To compute P lets assume ${P_1}^T, {P_2}^T, {P_3}^T$ are three row vectors of P. We compute A as given in equation .
     Where AP = 0 and w = 1

$$A = \begin{bmatrix} 0_4^T & -w_i'\mathrm{X}_i^T & v_i'\mathrm{X}_i^T \\ -w_i'\mathrm{X}_i^T & 0_4^T & -u_i'\mathrm{X}_i^T \\ -v_i'\mathrm{X}_i^T & -u_i'\mathrm{X}_i^T & 0_4^T \end{bmatrix}$$

   - We solve for P by computing SVD of A. $A = UDV^T$, P is last row of V obtained by SVD of A reshaped to $(3, 4)$. We divide P by its last element to get 1 at last position.

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$$

     Left 3x3 submatrix M of P is of form $M = KR$.

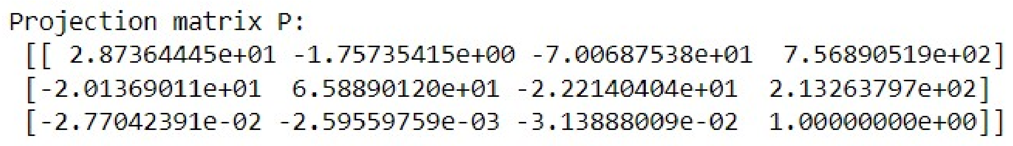
     K is an upper triangular matrix. R is an orthogonal matrix.

- The projection matrix (P) is decomposed into intrinsic parameters (K) and extrinsic parameters (R and t) using the RQ decomposition. The intrinsic matrix (K) is obtained by extracting the upper triangular part of the RQ decomposition and normalizing the diagonal elements to 1. The rotation matrix (R) and translation vector (t) are obtained from the lower triangular part of the RQ decomposition.

- The reprojection error is calculated by projecting each world point onto the image plane using the projection matrix, normalizing the projected point by its homogeneous coordinate, and comparing it with the corresponding image point. The reprojection error is the Euclidean distance between the normalized projected point and the image point. The mean reprojection error is the average of the reprojection errors over all the points.

- The formula for reprojection error is the Euclidean distance between the projected point and the corresponding image point in pixel coordinates:
  Reprojection error $= ||(u, v) - (u', v')||$
  where (u, v) is the image point in pixel coordinates, and (u', v') is the projected point obtained by multiplying the world point by the projection matrix P and normalizing the resulting homogeneous coordinates.

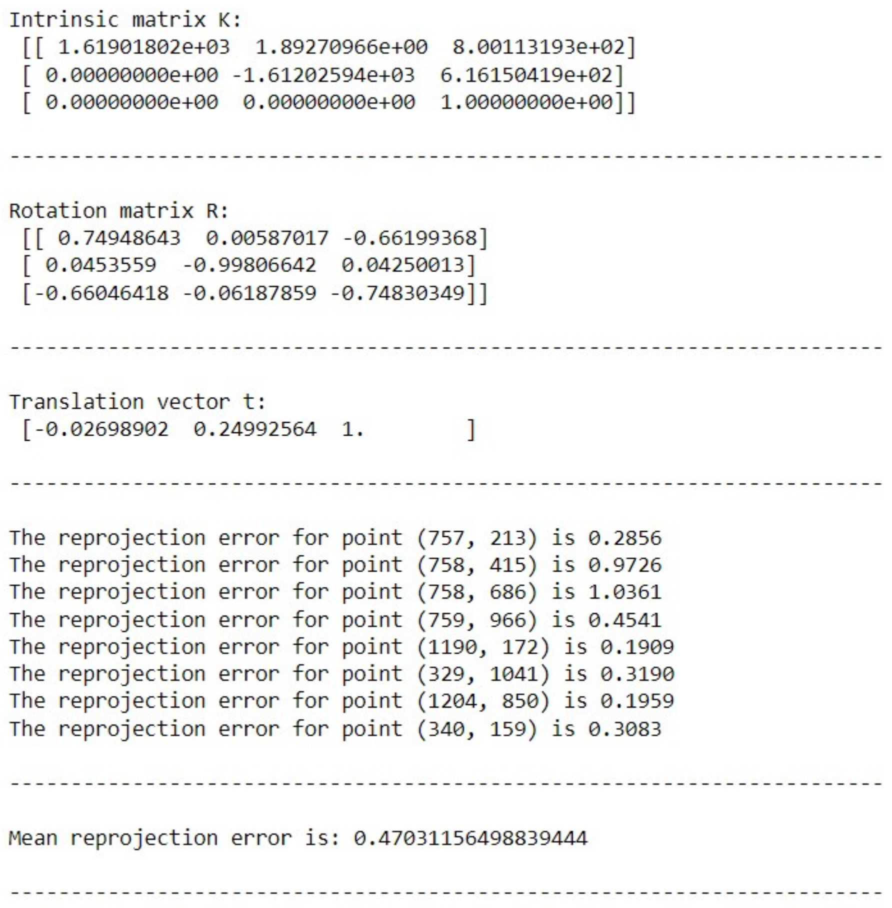
4. **Find the P matrix.**



```
Projection matrix P:
 [[ 2.87364445e+01 -1.75735415e+00 -7.00687538e+01  7.56890519e+02]
 [-2.01369011e+01  6.58890120e+01 -2.22140404e+01  2.13263797e+02]
 [-2.77042391e-02 -2.59559759e-03 -3.13888009e-02  1.00000000e+00]]
```

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.**

The camera intrinsic matrix, Rotation and Translation matrix and Reprojection errors for each points obtained is given below



```
Intrinsic matrix K:
 [[ 1.61901802e+03  1.89270966e+00  8.00113193e+02]
 [ 0.00000000e+00 -1.61202594e+03  6.16150419e+02]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]

------------------------------------------------------------------

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

------------------------------------------------------------------

Translation vector t:
 [-0.02698902  0.24992564  1.        ]

------------------------------------------------------------------

The reprojection error for point (757, 213) is 0.2856
The reprojection error for point (758, 415) is 0.9726
The reprojection error for point (758, 686) is 1.0361
The reprojection error for point (759, 966) is 0.4541
The reprojection error for point (1190, 172) is 0.1909
The reprojection error for point (329, 1041) is 0.3190
The reprojection error for point (1204, 850) is 0.1959
The reprojection error for point (340, 159) is 0.3083

------------------------------------------------------------------

Mean reprojection error is: 0.47031156498839444

------------------------------------------------------------------
```

2 Problem 2:

This was printed on an A4 paper and the size of each square is 21.5 mm. Note that the Y axis has an odd number of squares and X axis has an even number of squares. It is a general practice to neglect the outer squares (extreme squares on each side and in both directions). Thirteen images taken from a Google Pixel XL phone with focus locked can be downloaded from here which you will use to calibrate. For this question, you are allowed to use any in-built function.

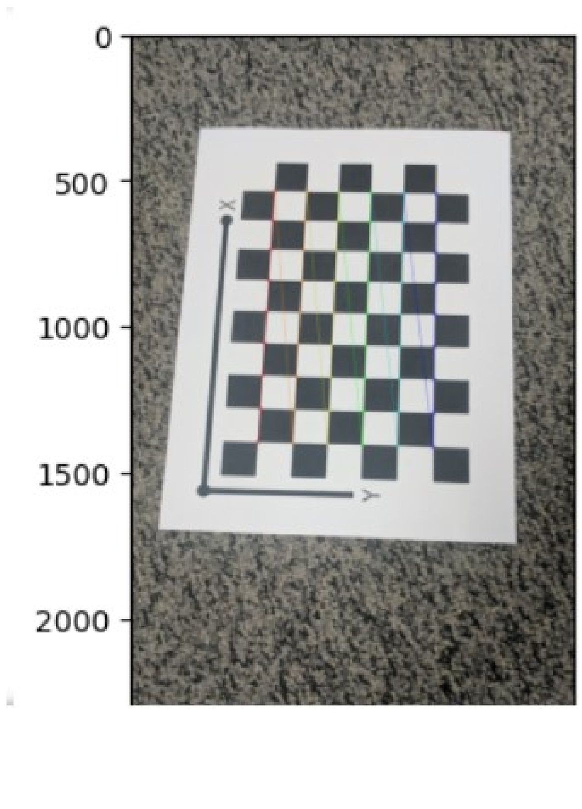
1. **Find the checkerboard corners using any corner detection method (inbuilt OpenCV functions such as findChessboardCorners are allowed) and display them for each image.**

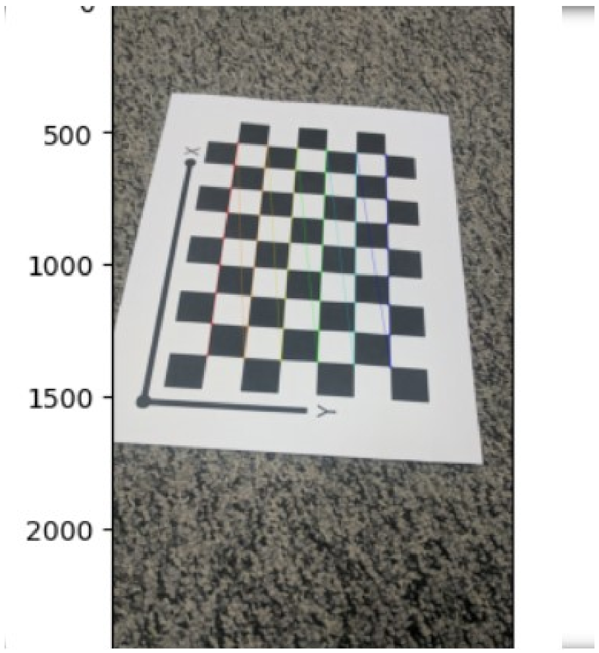
2.1 Pipeline

- **STEP:1–Choose a calibration target:** A common calibration target is a chessboard pattern with known dimensions of each square.
- **STEP:2–Capture multiple images of the calibration target:** Take multiple images of the calibration target from different angles and positions. It is recommended to capture images with varying lighting conditions and orientations to improve the accuracy of the calibration.

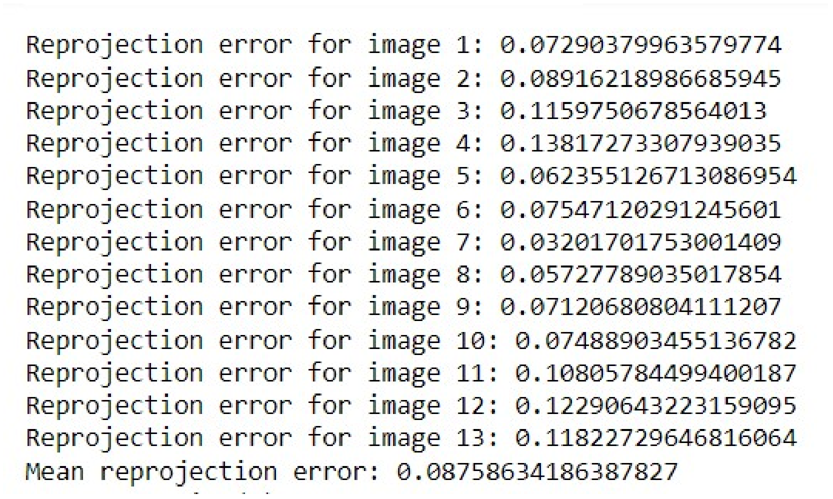
- **STEP:3–Detect the corners of the calibration target:** Use a corner detection algorithm, such as the Harris corner detector, or OpenCV's findChessboardCorners() function to detect the corners of the chessboard in each image.
- **STEP:4–Compute the object points:** Calculate the 3D coordinates of each corner of the chessboard using the known dimensions of each square.
- **STEP:5–Compute the image points:** Use the detected corner coordinates to calculate the 2D coordinates of each corner in the image.
- **STEP:6–Estimate the camera matrix and distortion coefficients:** Use OpenCV's calibrateCamera() function to estimate the camera matrix (K) and distortion coefficients (D) that best describe the relationship between the 3D world coordinates and 2D image coordinates.
- **STEP:7–Evaluate the calibration:** Evaluate the calibration by computing the reprojection error, which measures the difference between the projected image points and the observed image points. The lower the reprojection error, the more accurate the calibration.
- **STEP:8–Use the camera matrix and distortion coefficients:** Once the camera matrix and distortion coefficients are obtained, they can be used to correct for the distortions and obtain accurate measurements and positions of objects in the image.

Corner detection for some images is displayed below



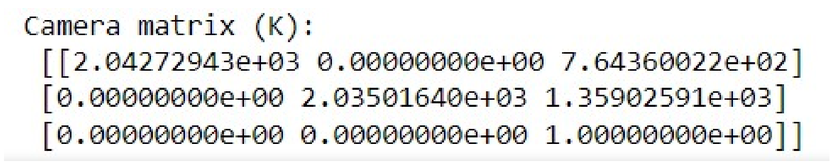


2. **Compute the Reprojection Error for each image using built-in functions in OpenCV**



```
Reprojection error for image 1: 0.07290379963579774
Reprojection error for image 2: 0.08916218986685945
Reprojection error for image 3: 0.1159750678564013
Reprojection error for image 4: 0.13817273307939035
Reprojection error for image 5: 0.062355126713086954
Reprojection error for image 6: 0.07547120291245601
Reprojection error for image 7: 0.03201701753001409
Reprojection error for image 8: 0.05727789035017854
Reprojection error for image 9: 0.07120680804111207
Reprojection error for image 10: 0.07488903455136782
Reprojection error for image 11: 0.10805784499400187
Reprojection error for image 12: 0.12290643223159095
Reprojection error for image 13: 0.11822729646816064
Mean reprojection error: 0.08758634186387827
```

3. **Compute the K matrix ?**



```
Camera matrix (K):
 [[2.04272943e+03 0.00000000e+00 7.64360022e+02]
 [0.00000000e+00 2.03501640e+03 1.35902591e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

4. **How can we improve the accuracy of the K matrix?** To improve the accuracy of the K matrix, we can:

   (a) **Increase the number of images used for calibration:** Using more images will provide more data to improve the accuracy of the calibration process.

   (b) **Use images with different orientations and lighting conditions:** Using images with different orientations and lighting conditions will provide a better representation of the real-world scenarios and help to avoid overfitting to a particular set of images.

   (c) **Increase the size of the chessboard or use a more accurate measuring device to determine the size of each square:** Using a larger chessboard or a more accurate measuring device to determine the size of each square will provide more accurate object points and improve the accuracy of the calibration process.

   (d) **Use a more accurate corner detection method, such as the Harris corner detector, or manually label the corners:** The accuracy of the corner detection method can affect the accuracy of the calibration process. Using a more accurate corner detection method, such as the Harris corner detector, or manually labeling the corners can improve the accuracy of the image points used in the calibration process.