ELL793

Computer Vision

Prof. Brejesh Lall

Assignment 1:
Camera Calibration

Submitted by:
Raghav Gupta (2017EE10544)
Prakhar Kanchan (2017EE30543)

Goal

To find intrinsic and extrinsic camera calibration parameters of a mobile phone camera using a checkerboard object on orthogonal walls.

Setup

- A 10X7 Checkerboard shown in Figure 1 was printed on 2 A4-sized sheets.

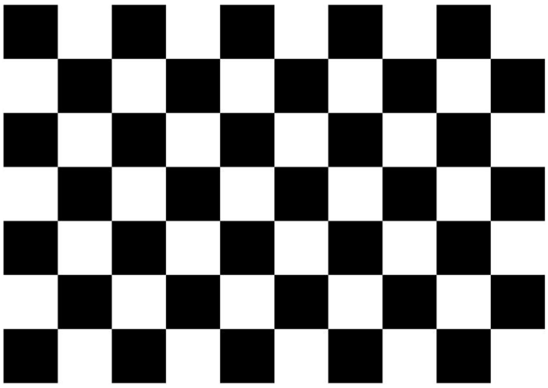

Fig. 1: 10X7 Checkerboard

- The printed checkerboards were pasted on a corner wall, and some points were marked with a red marker, as shown in Figure 2.

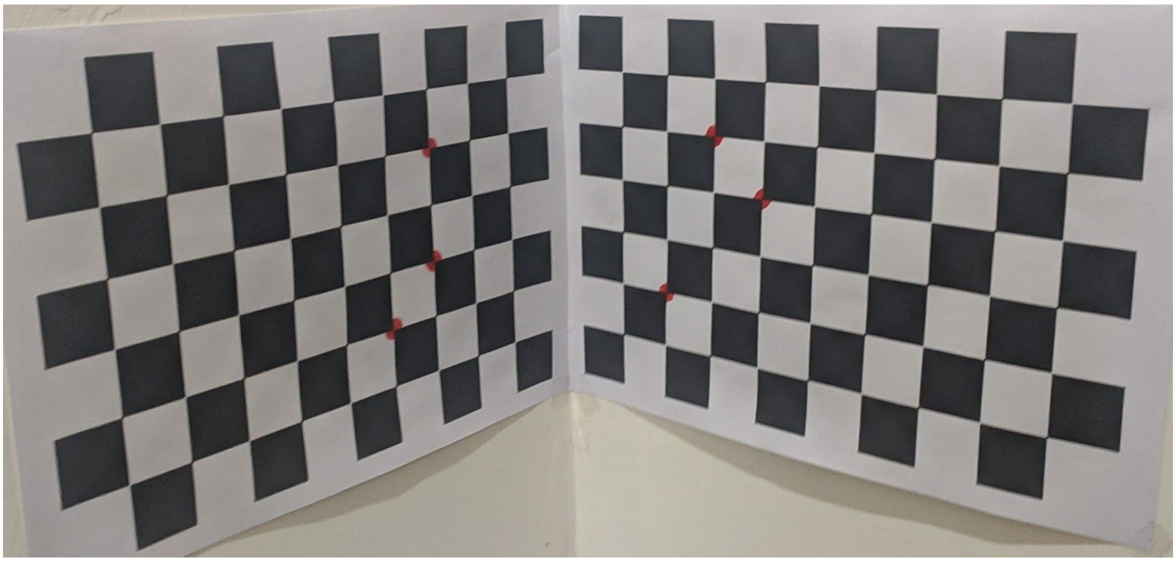

Fig. 2: Checkerboards on orthogonal walls

- The camera used has the following specs -
    - Google Camera version 7.2.010
    - Aperture $f/1.75$
    - Exposure Time 1/33
    - Focal length 4.77mm
    - ISO 1210

Dataset Creation

- 3D coordinates were measured using a scale on the checkerboards with the origin marked, as shown in Figure 3.
- The sides of the squares on the board measure 2.7 cm each. The white space around the squares measures 1.1 cm around the origin.
- These measures were then used to calculate the 3D coordinates of each of the points marked.
- To find the 2D coordinates of points on the image, OpenCV's setMouseCallback method was used for left mouse click and displayed on the shell. These were then saved into a CSV file.
- Figure 3 displays the 2D coordinates on the image.

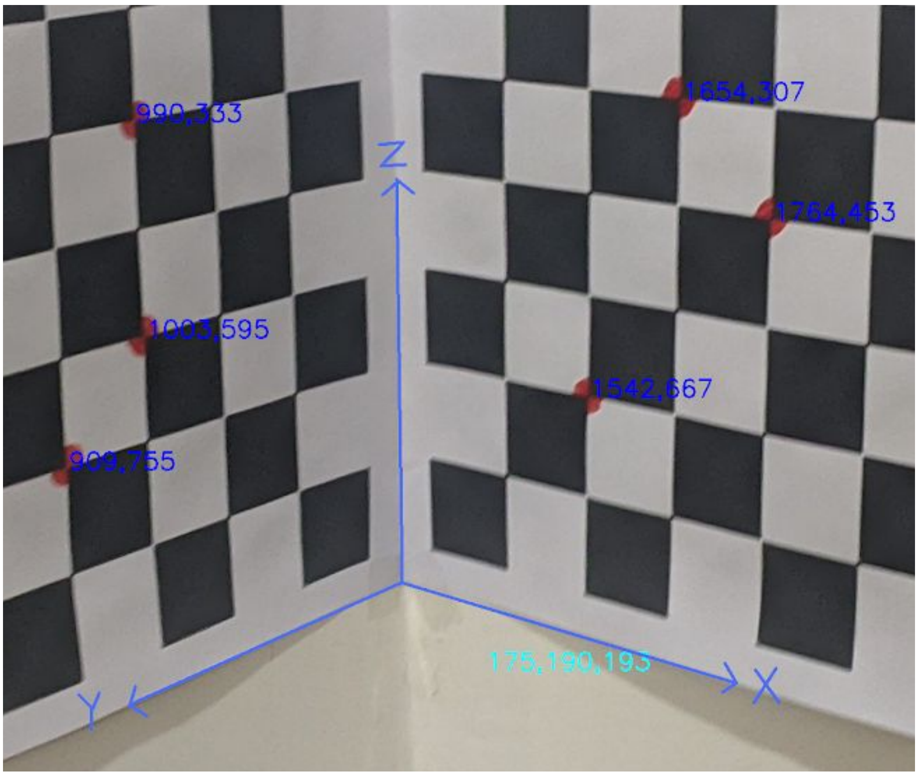

Fig. 3: 2D coordinates of marked points on the image

3D Points			2D Points	
x coordinate	y coordinate	z coordinate	x coordinate	y coordinate
0	9.2	14.6	990	333
0	9.2	9.2	1003	595
0	11.9	6.5	909	755
6.5	0	6.5	1542	667
11.9	0	11.9	1764	453
9.2	0	14.6	1654	307

Table 1: Dataset created

Procedure

- These points were normalized by finding T and U matrices such that ^x = Tx and ^X = UX gives the normalized points with centroid at origin and the average distance from origin equal to $\sqrt{2}$ and $\sqrt{3}$ for 2D and 3D points, respectively. The matrices T and U are given as below:

$$T = \begin{bmatrix} \frac{1}{d_{2d}} & 0 & \frac{-x_{centroid}}{d_{2d}} \\ 0 & \frac{1}{d_{2d}} & \frac{-y_{centroid}}{d_{2d}} \\ 0 & 0 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} \frac{1}{d_{3d}} & 0 & 0 & \frac{-X_{centroid}}{d_{3d}} \\ 0 & \frac{1}{d_{3d}} & 0 & \frac{-Y_{centroid}}{d_{3d}} \\ 0 & 0 & \frac{1}{d_{3d}} & \frac{-Z_{centroid}}{d_{3d}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$(x_{centroid}, y_{centroid})$ is the centroid of 2D points

$(x_{centroid}, y_{centroid}, z_{centroid})$ is the centroid of 3D points

$$d_{2d} = \frac{\text{Average distance from centroid of 2D points}}{\sqrt{2}}$$

$$d_{3d} = \frac{\text{Average distance from centroid of 3D points}}{\sqrt{3}}$$

Why Normalization?

By normalizing the 2D and 3D points, we are reducing the variance in our dataset. Since units of measurement are different for 2D and 3D coordinates, the dataset would have high variance. By reducing variance, the RMSE between the 2D coordinates computed using the projection matrix and the ground truth 2D points would be less.

- After computing the normalized coordinates, we compute the projection matrix by the following equation:

$$\mathcal{P}\boldsymbol{m} = 0,$$

$$\mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -x_1\boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -y_1\boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -x_n\boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -y_n\boldsymbol{P}_n^T \end{pmatrix} \quad \text{and} \quad \boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix} = 0.$$

The projection matrix m is obtained by finding the eigenvector of $P^T P$ corresponding to the smallest eigenvalue. We then denormalize the projection matrix m using the following equation:

$$m = T^{-1} m U$$

- After computing m we find the intrinsic and extrinsic parameters using the following set of equations:

$$\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b}).$$

$$\begin{aligned} &\rho = \varepsilon / ||\boldsymbol{a}_3||, \\ &\boldsymbol{r}_3 = \rho \boldsymbol{a}_3, \\ &x_0 = \rho^2(\boldsymbol{a}_1 \cdot \boldsymbol{a}_3), \\ &y_0 = \rho^2(\boldsymbol{a}_2 \cdot \boldsymbol{a}_3), \end{aligned}$$

$$
\begin{aligned}
\cos\theta &= -\frac{(\boldsymbol{a}_1 \times \boldsymbol{a}_3)\cdot(\boldsymbol{a}_2 \times \boldsymbol{a}_3)}{||\boldsymbol{a}_1 \times \boldsymbol{a}_3||\,||\boldsymbol{a}_2 \times \boldsymbol{a}_3||},\\
\alpha &= \rho^2||\boldsymbol{a}_1 \times \boldsymbol{a}_3||\sin\theta,\\
\beta &= \rho^2||\boldsymbol{a}_2 \times \boldsymbol{a}_3||\sin\theta,
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{r}_1 &= \frac{\rho^2\sin\theta}{\beta}(\boldsymbol{a}_2 \times \boldsymbol{a}_3) = \frac{1}{||\boldsymbol{a}_2 \times \boldsymbol{a}_3||}(\boldsymbol{a}_2 \times \boldsymbol{a}_3),\\
\boldsymbol{r}_2 &= \boldsymbol{r}_3 \times \boldsymbol{r}_1.
\end{aligned}
$$

$$
\mathcal{K} \stackrel{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha\cot\theta & x_0 \\ 0 & \dfrac{\beta}{\sin\theta} & y_0 \\ 0 & 0 & 1 \end{pmatrix}.
$$

$$
\boldsymbol{t} = \rho\mathcal{K}^{-1}\boldsymbol{b}.
$$

Figure 4 shows the code flow diagram.

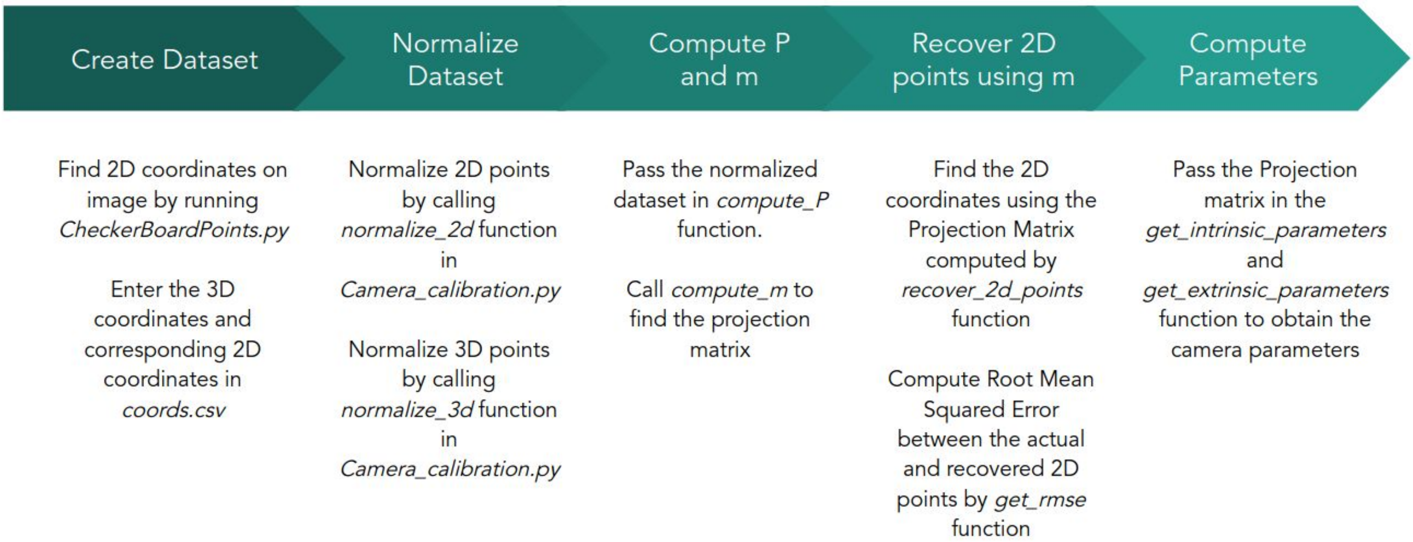

Fig. 4: Code flow diagram

Results

Using the dataset shown in Table 1 and the procedure mentioned above, projection matrix m obtained is as follows:

$$m = \begin{bmatrix} -1.71827428e+01 & 3.12075749e+01 & 4.75921141e+00 & -9.81900974e+02 \\ -1.25426669e+00 & -2.51010383e+00 & 3.26293545e+01 & -6.63527845e+02 \\ 5.77660173e-03 & 7.48176032e-03 & 3.24309813e-03 & -7.47753556e-01 \end{bmatrix}$$

The points recovered are plotted on the image below:

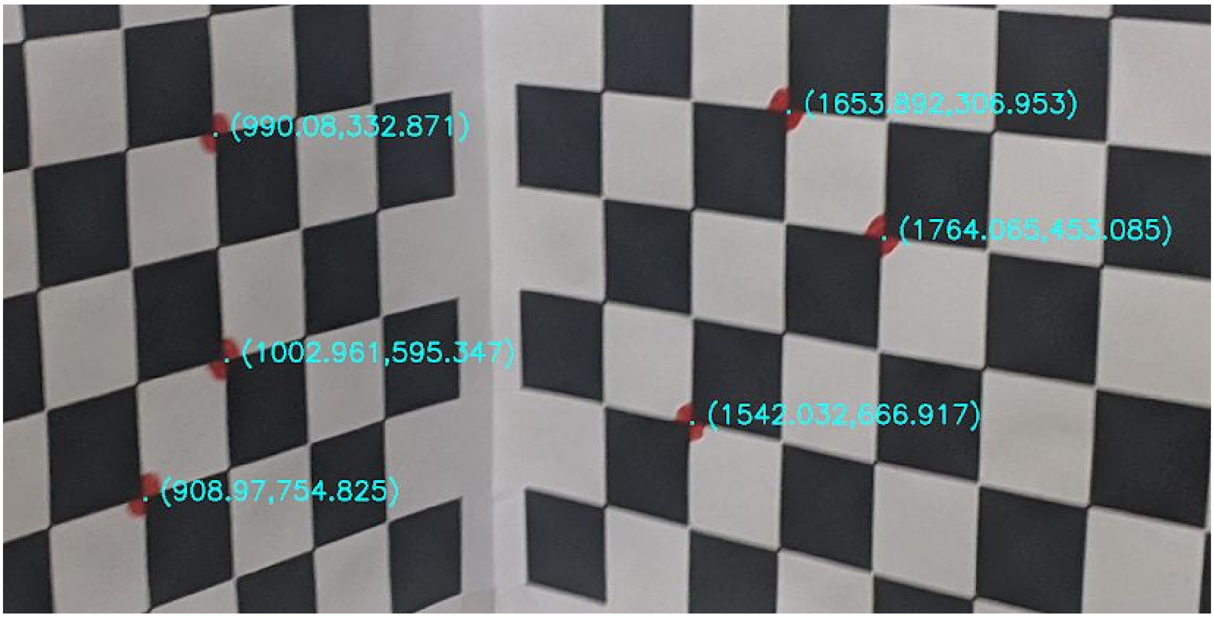

Fig. 4: Recovered Points marked on the image

The Root Mean Squared Error between the points recovered is <u>0.132</u>
Scaling Factor (rho) = 100.068

Intrinsic Parameters

Intrinsic Parameter	Value
α	3268.8328
β	3177.6579
θ	88.836°
x_0	1498.6881
y_o	799.0383

$$K = \begin{bmatrix} 3.26883285e+03 & -6.64152459e+01 & 1.49868819e+03 \\ 0 & 3.17831377e+03 & 7.99038327e+02 \\ 0 & 0 & 1 \end{bmatrix}$$

We obtained k and l from the EXIF tags of the JPEG images. From that the focal length was computed using the following equations:

$$\alpha = kf \qquad \beta = lf$$

Where $1/k \, * \, 1/l$ is the dimension of pixel

Focal Length was found out to be <u>4.8mm</u>

Extrinsic Parameters

$$R = \begin{bmatrix} -0.79479317 & 0.60666633 & 0.01611778 \\ -0.18481504 & -0.26725222 & 0.94573762 \\ 0.57805468 & 0.748687 & 0.3245313 \end{bmatrix}$$

The Euler angles obtained through this R matrix is as follows:
[x: -71.0677783°, y: 0.939456°, z: -142.6566923°]

$$t = \begin{bmatrix} 4.20523744 \\ -2.07937932 \\ -74.82642318 \end{bmatrix}$$