CMSC 733 Homework 1: AutoCalib Using 1 late day

Naitri Rajyaguru
email: nrajyagu@umd.edu

***Abstract*—Camera converts 3D world into 2D image. Camera Calibration is extensively used in image processing or computer vision to identify the geometric characteristics of the image. This report discusses camera calibration algorithm developed by Zhengyou Zhang.**

I. Introduction

The algorithm developed by Zhang provides both Intrinsic and Extrinsic parameters of a camera. It assumes that one has calibration pattern in Z=0 plane of the world frame and one has images form different viewpoints. Exploiting certain constraints of the geometric relationship between 2D representation of 3D point on an image, we can obtain camera parameters.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathrm{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The intrinsic parameters involves of focal length, distortion coefficients and principle points of a camera.

$$K = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Where α and β are the scale factors in image u a and γ is the parameter describing skewness of the axes. The principal points are given by (u_0, v_0).

And the extrinsic parameters are given by

$$\begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}$$

It describes orientation and location of the camera with respect to some world coordinate system.

Fig. 1: Camera Calibration

II. Data

A checkerboard pattern has been used to compute camera parameters/ The pattern was printed on an A4 paper with each sqaure of 21.5mm. In total 13 images were taken from Google Pixel XL phone.

A. Image and World Points

Image Points are obtained using findChessboardCorners to detect corners of checkerboard. Here we also use cornerSubPix is used as it provides accurate location of corners. World points are obtained by creating mesh grid of nx=9 and ny=6 and it was multiplied by the square size.

III. Parameter Estimation

A. Estimating K

As mentioned in the paper, if Z = 0, we can compute Homography between points and compute K matrix.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [\begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

For 54 image and world points of each image, Homography matrix is computed which results into 13 H matrices.

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

According to the paper, we cannot directly compute or decompose P matrix and hence we compute B matrix.

$$B = K^{-T}K^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - v_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

Again, we have all unknowns and hence to retrieve B we estimate it using homogrpahy matrix by using the following equation.

$$h_i^T B h_j = v_{ij}^T b$$

Taking SVD of V matrix obtained results into b as mentioned and we can easily decompose K.

K =

$$[\begin{bmatrix} 2.05441008\mathrm{e}+03 & 8.87498939\mathrm{e}-02 & 7.53978139\mathrm{e}+02 \end{bmatrix} \\ \begin{bmatrix} 0.00000000\mathrm{e}+00 & 2.04184794\mathrm{e}+03 & 1.35442281\mathrm{e}+03 \end{bmatrix} \\ \begin{bmatrix} 0.00000000\mathrm{e}+00 & 0.00000000\mathrm{e}+00 & 1.00000000\mathrm{e}+00 \end{bmatrix}]$$

B. Estimating R and t

Once K is known, the extrinsic parameters for each image is computed as follows

$$r_1 = \lambda A^{-1} h_1$$
$$r_2 = \lambda A^{-1} h_2$$
$$r_3 = r_1 \times r_2$$
$$t = \lambda A^{-1} h_3$$

R and t for one of the image is [Rt] =

$$\begin{bmatrix} 9.99010922\text{e}-01 & -3.64038942\text{e}-02 & 2.93599273\text{e}-02 \\ 2.72983847\text{e}-02 & 9.82260203\text{e}-01 & 1.88536776\text{e}-01 \\ -3.50995153\text{e}-02 & -1.87444414\text{e}-01 & 9.82282439\text{e}-0 \end{bmatrix}$$

C. Approximate Distortion kc

Initially, kc is assumed to be [0,0].Hence, projection error prior optimization is 0.7041599314205579.

IV. Non-Linear Geometric Error Minimization

Initially we assumed lens distortion to be zero but in reality it is present. Hence to remove lens distortion, we need an optimized estimation of camera parameters. This optimization was done with scipy.optimize function with Levenberg-Marquardat method. The points are projected using K initial on the image and error is computed with original image points.

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \|m_{ij} - \hat{m}\left(A, R_i, t_i, M_j\right)\|^2$$

The new obtained camera calibration matrix K, extrinsic and image points are computed. The result is shown below:

K = $\begin{bmatrix} 2.05440713\text{e}+03 & 8.75674957\text{e}-02 & 7.53980268\text{e}+0 \\ 0.00000000\text{e}+00 & 2.04184162\text{e}+03 & 1.35443870\text{e}+03 \\ 0.00000000\text{e}+00 & 0.00000000\text{e}+00 & 1.00000000\text{e}+00 \end{bmatrix}$

[Rt] =

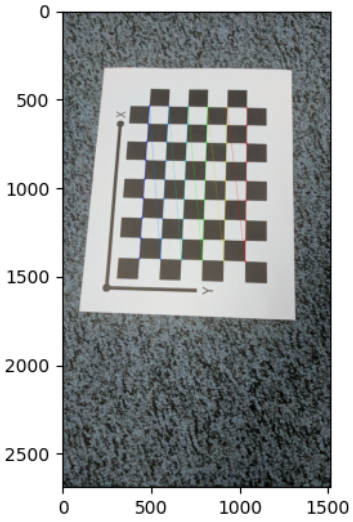
$$\begin{bmatrix} 9.99010915\text{e}-01 & -3.64031326\text{e}-02 & 2.93599302\text{e}-02 \\ 2.72987015\text{e}-02 & 9.82263230\text{e}-01 & 1.88536466\text{e}-01 \\ -3.50994627\text{e}-02 & -1.87444134\text{e}-01 & 9.82285447\text{e}-0 \end{bmatrix}$$

The distortion is kc = (0.014522481675472676, -0.10732518914895216)

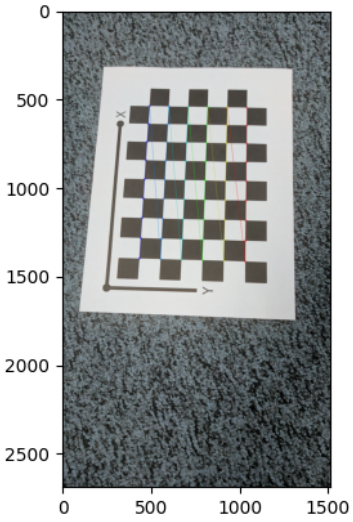
The reprojection error is 2.319669910816789.

Input images and undistorted images are shown in result section.

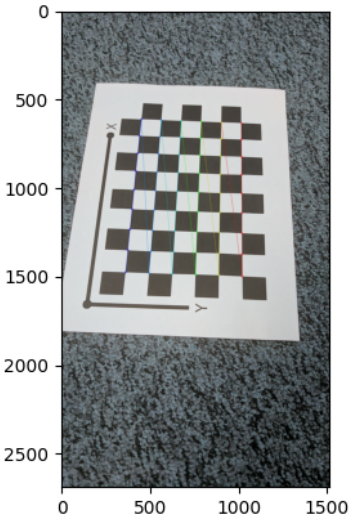
V. Results



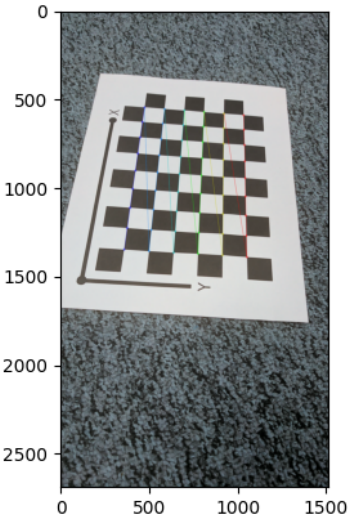
(a) Input image 1



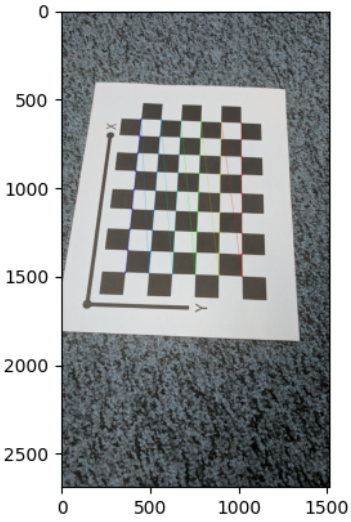
(b) Undistorted image 1



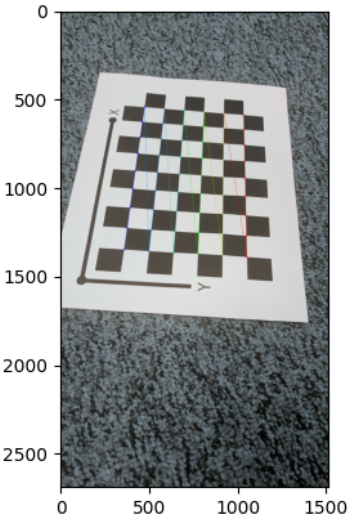
(a) Input image 2



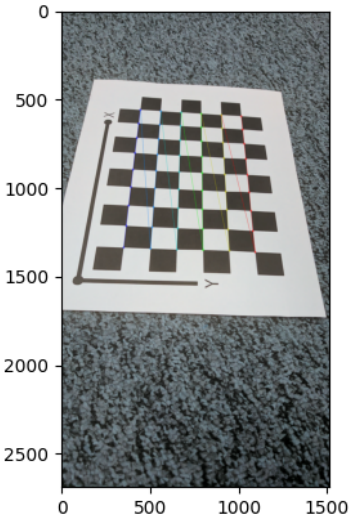
(a) Input image 3



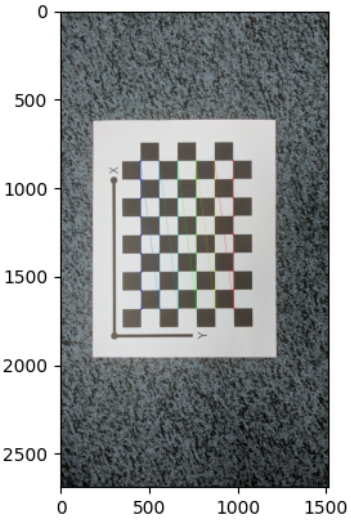
(b) Undistorted image 2



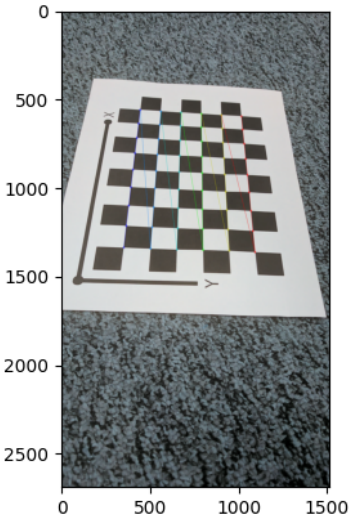
(b) Undistorted image 3



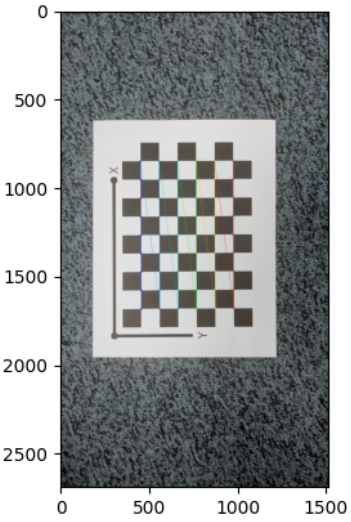
(a) Input image 4



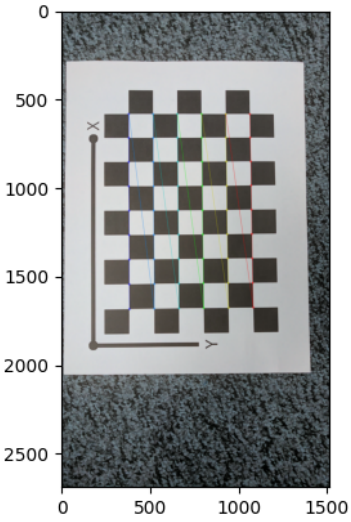
(a) Input image 5



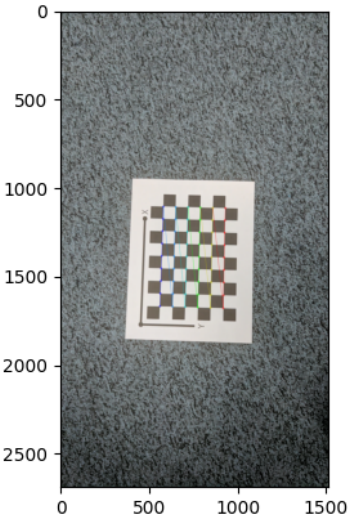
(b) Undistorted image 4



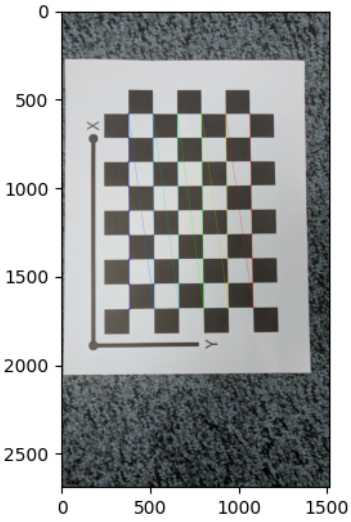
(b) Undistorted image 5



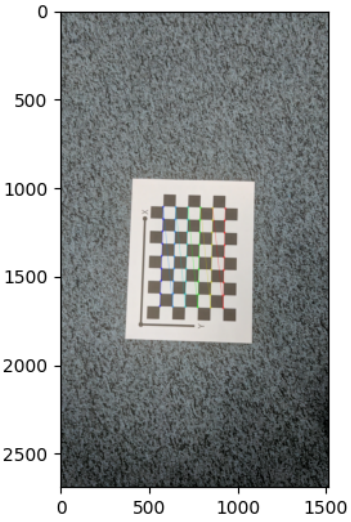
(a) Input image 5



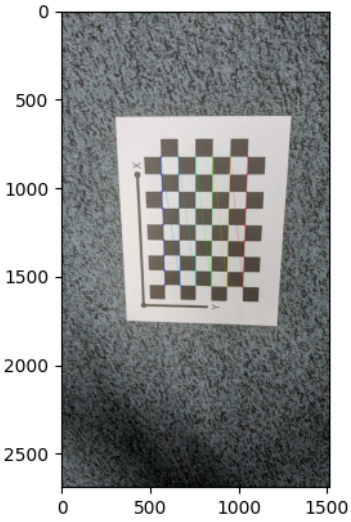
(a) Input image 6



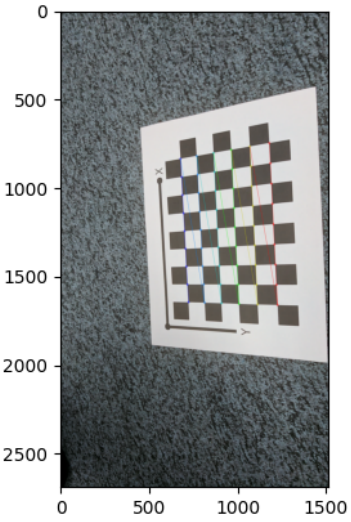
(b) Undistorted image 5



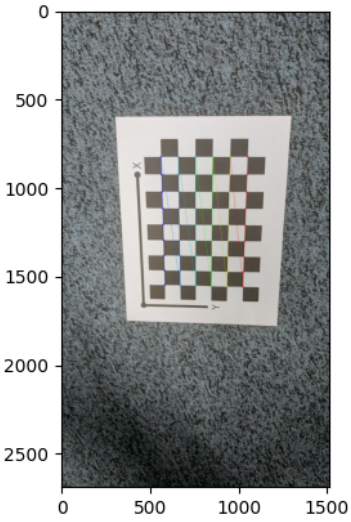
(b) Undistorted image 6



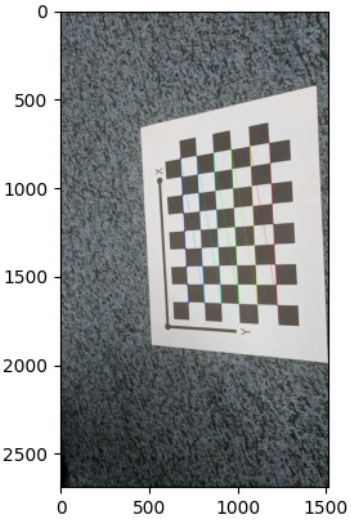
(a) Input image 7



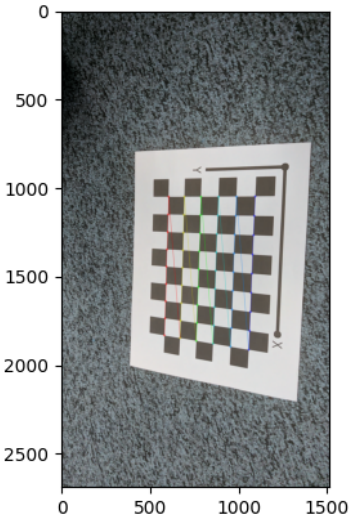
(a) Input image 8



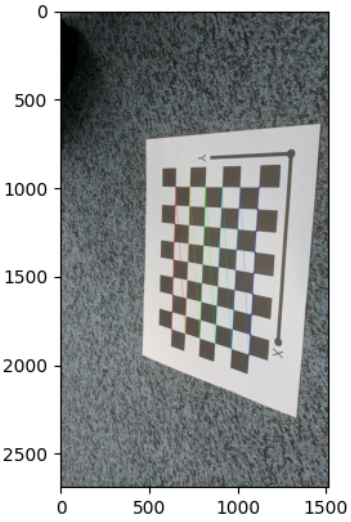
(b) Undistorted image 7



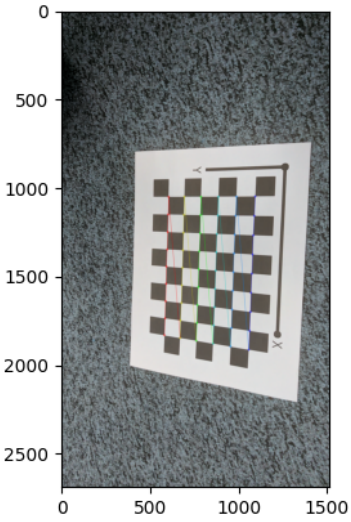
(b) Undistorted image 8



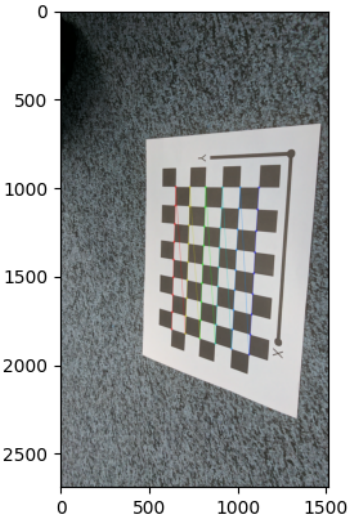
(a) Input image 9



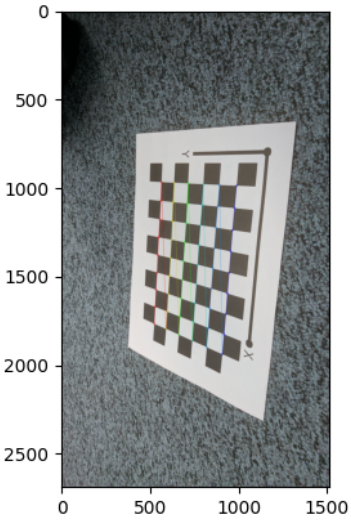
(a) Input image 10



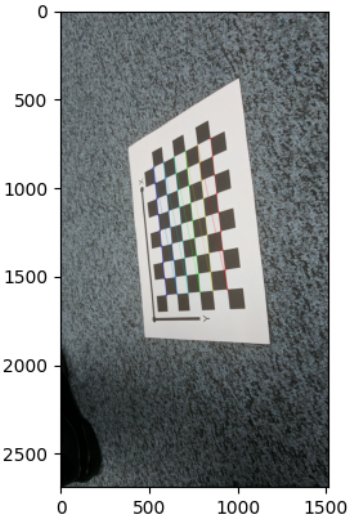
(b) Undistorted image 9



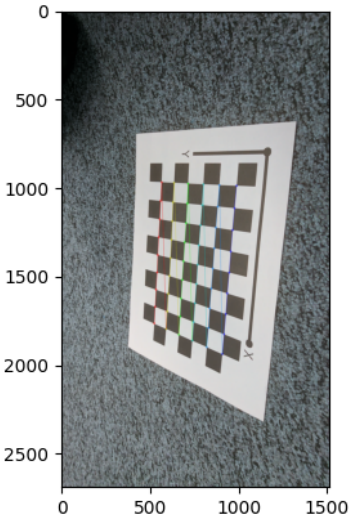
(b) Undistorted image 10



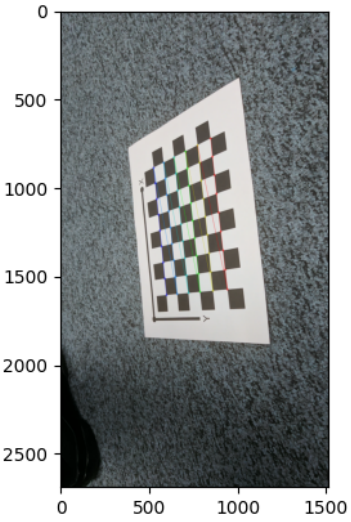
(a) Input image 11



(a) Input image 12



(b) Undistorted image 11



(b) Undistorted image 12