# RBE/CS 549 - HW1: AutoCalib

Cole Parks
Worcester Polytechnic Institute
cparks@wpi.edu

***Abstract*—Automatic camera calibration is a critical step in computer vision research. This paper presents a method to estimate the camera intrinsic parameters and radial distortion coefficients using a set of images of a calibration target, based off of the work of Z. Zhang and is regarded as one of the hallmark papers in camera calibration. The method is tested on a set of images of a checkerboard pattern taken from a Google Pixel XL phone with focus locked. The results are compared to the ground truth values and the accuracy of the method is evaluated. The method is found to be accurate and robust for the given set of images.**

## I. Initial Parameter Estimate

### A. Camera Intrinsic Matrix

First, we need to estimate the camera intrinsic matrix, $K$. This matrix is defined as follows:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $f_x$ and $f_y$ are the focal lengths in the x and y directions, respectively, and $c_x$ and $c_y$ are the principle points in the x and y directions, respectively. We can estimate the camera intrinsic matrix using the `cv2.findChessboardCorners` function in OpenCV to find the corners of the checkerboard pattern in the images. The checkerboard pattern used in this project is shown in Figure 1.

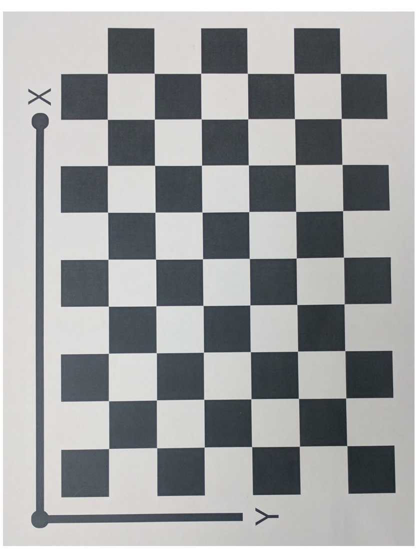


Fig. 1: Checkerboard Pattern

The pattern dimensions were chosen to be 9x6, since the grid is 10x7 and we neglect the outermost row and column of squares. After finding the corners of the checkerboard, we calculated the real world coordinates of the corners using the known pattern size (normalized to 1 since the actual size is not important). From these coordinates, the homography matrices for each image were calculated, and used to generate the closed-form solution to the camera intrinsic matrix, $K$. This was done by creating a matrix of stacked correspondences between the real world coordinates and the image coordinates, with size of $2n \times 9$, where $n$ is the number of corners found in the image. Along with the stacked homographies, $H$, we solved this using SVD, and then reshaped the resulting vector into the camera intrinsic matrix, $K$. This was done by calculating $vo$, $lambda$, $alpha$, $beta$, and $gamma$ from the SVD results, and then calculating the camera intrinsic matrix, $K$, using the following equations:

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2}$$

$$\lambda = B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}}$$

$$\alpha = \sqrt{\frac{\lambda}{B_{11}}}$$

$$\beta = \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}}$$

$$\gamma = -B_{12}\alpha^2\beta/\lambda$$

$$u_0 = \frac{\gamma v_0}{\beta} - \frac{B_{13}\alpha^2}{\lambda}$$

where $B_{ij}$ is the $i$th row and $j$th column of the homography matrix, $H$. The results are shown in Figure 2, in the format of $K$ from Equation 1.

```
Camera Intrinsic Matrix K:
 [[ 2.06189191e+03 -2.85059170e+00  7.76002424e+02]
 [ 0.00000000e+00  2.04779869e+03  1.36323995e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Fig. 2: Camera Intrinsic Matrix, $K$

### B. Camera Extrinsic Parameters

Next, we needed to estimate the camera extrinsic parameters, $R$ and $t$. The rotation matrix, $R$, and the translation vector, $t$, represent the rotation and translation of the camera in the world frame. We can estimate these parameters using the same homography matrices used to estimate the camera intrinsic matrix, $K$. The rotation matrix, $R$, and the translation vector, $t$, can be estimated using the following equations:

$$H = K[R|t]$$

where $H$ is the homography matrix, $K$ is the camera intrinsic matrix, $R$ is the rotation matrix, and $t$ is the translation vector. We can solve for $R$ and $t$ using the following equations:

$$h_1 = K[r_1|t]$$
$$h_2 = K[r_2|t]$$
$$h_3 = K[r_3|t]$$

where $h_1$, $h_2$, and $h_3$ are the columns of $H$, and $r_1$, $r_2$, and $r_3$ are the columns of $R$. We can solve for $R$ and $t$ by solving the following equation:

$$\lambda K^{-1} h_1 = [r_1|t]$$
$$\lambda K^{-1} h_2 = [r_2|t]$$
$$\lambda K^{-1} h_3 = [r_3|t]$$

where $\lambda$ is a scaling factor.

### C. Radial Distortion Parameters

Finally, we need to estimate the radial distortion parameters, $k_1$ and $k_2$. Since this has not been optimized yet, we provide a guess of (0, 0).

## II. Non-linear Geometric Error Minimization

In order to minimize the camera distortion, we minimize the reprojection error using the equation

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||m_{ij} - \hat{m}(\mathbf{A}, \mathbf{k_1}, \mathbf{k_2}, \mathbf{R_i}, \mathbf{t_i}, \mathbf{M_j})||^{\mathbf{2}} \qquad (2)$$

After optimization, the distortion coefficients, $k_1$ and $k_2$, were found to be:

```
[-1.49682936e-22 -9.09494702e-13]
```

Fig. 3: Optimized Distortion Coefficients, $k_1$ and $k_2$

The full calibration matrix after optimizing the distortion coefficients is shown in Figure 4.

```
Camera Intrinsic Matrix K:
 [[ 2.06189191e+03 -2.85059170e+00  7.76002424e+02]
 [ 0.00000000e+00  2.04779869e+03  1.36323995e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Fig. 4: Optimized Camera Intrinsic Matrix, $K$

### A. Reprojected Corners

A few rectified images are shown with rectified and reprojected corners:

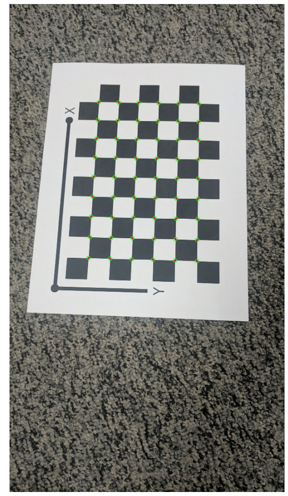

Fig. 5: Undistorted Image 1

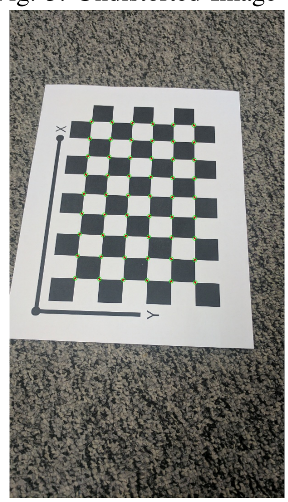

Fig. 6: Undistorted Image 2

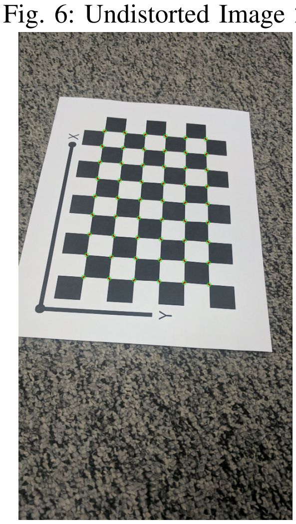

Fig. 7: Undistorted Image 3