# CMSC733: Homework 1: AutoCalib

Varun Asthana
vasthana@umd.edu

## I. Initial Parameter Estimation

To calibrate a camera, maximum likelihood estimation is used. For this an initial set of parameters is required for the variables to be optimized. Thus first step is to find the initial parameters- camera intrinsic matrix $(K)$, rotation and translation (RT) matrix for all images (to transform into camera coordinate system), and distortion coefficients $(K_d)$.

### A. Camera Intrinsic Matrix $(K)$

For each given image of the chess board of grid size (10 X 7), I found the corners of the inner grid (9 X 6) by using the *cv2.findChessboardCorners* function. Of these detected corners, only the four extremes were considered to find the homography (H) to transform world points to image points. Detected corners are shown in Fig 1.

Once all the H matrices were calculated, camera intrinsic matrix $(K)$ is calculated by following the procedure in the section 3.1 of [1]. In the process to solve eq(9) of [1], I used the *numpy.linalg.svd* function to find the solution for null space vector. Once the null space vector ('b' as defined in [1]) was known, then all the 5 unknown parameters of the camera intrinsic matrix was calculated. The obtained initial value for the matrix $K$-

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

$$K = \begin{bmatrix} 2034.75 & 0.49 & 772.70 \\ 0 & 2017.90 & 1366.90 \\ 0 & 0 & 1 \end{bmatrix}$$

### B. RT Matrix

From previous section, we now have the initial value of the K matrix. This matrix can now be used to estimate the rotation matrix and translation to bring world points into camera coordinate system. Using the equations given in the section 3.1 of [1], $r_1$, $r_2$, $r_3$ *and* $t$ vectors were found for each given calibration image.

### C. Distortion Coefficients $(K_d)$

We start by assigning $k_1$ and $k_2$ both as 0.

$$K_d = [k_1, k_2]$$

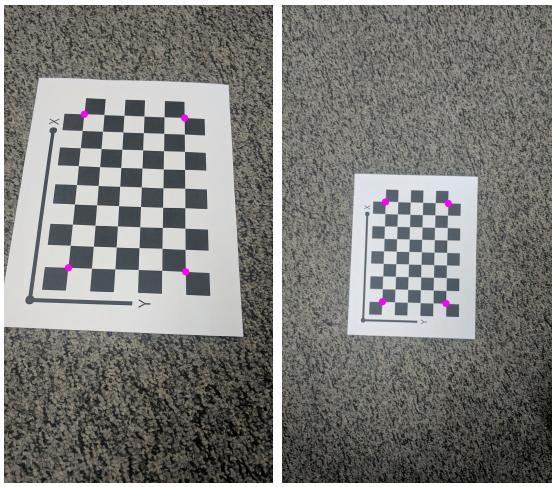


Fig. 1: Detected 4 corners for homography calculation

## II. Non Linear Optimization for $K$ and $K_d$

Non-linear Geometric Error Minimization of re-projection error is done by utilizing the *least_squares* function of *scipy.optimize*.

A residual function was defined which returns the value that is to be minimized by optimizing the first argument of this function. Once this was implemented with the *method='lm'*, the results obtained for the optimized camera intrinsic matrix $(K')$ and optimized distortion coefficients $(K'_d)$ were-

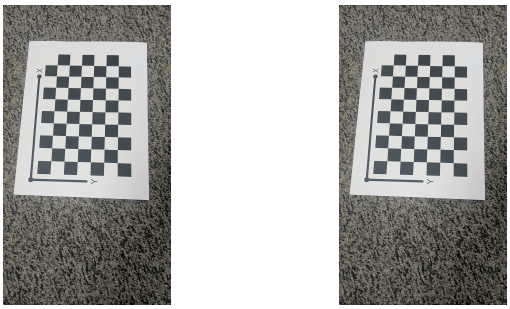
$$K' = \begin{bmatrix} 2.034746e+3 & -2.437024e-10 & 7.727043e+2 \\ 0 & 2.017909e+3 & 1.360909e+3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K'_d = \begin{bmatrix} 0.003739 & -0.004961 \end{bmatrix}$$

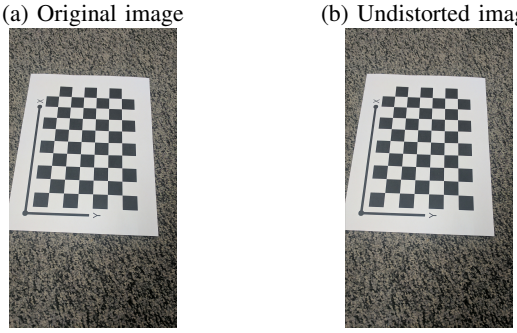
Using these new parameters, new RT matrix was calculated for all images to get the re-projection error. The mean error obtained was 3.594.

Also these parameters were used to get the new undistorted images using the *cv2.undistort()* function. The outputs are shown in Fig 2. From the original images, it was visible that they did not have any (or any noticeable) radial distortion, hence the output images are very much alike the original

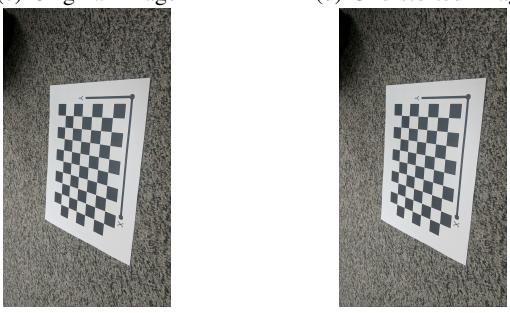
images. This is also confirmed by the near to zero values of distortion coefficients.



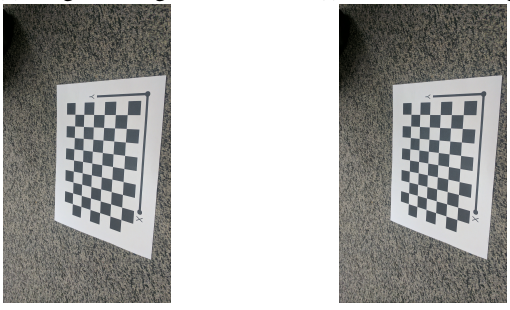
(a) Original image (b) Undistorted image



(c) Original image (d) Undistorted image



(e) Original image (f) Undistorted image



(g) Original image (h) Undistorted image

Fig. 2: Sample outputs for original and undistorted images

## III. Conclusion

Auto-calibration task was successfully completed in the required manner. The results were also as per the expectation i.e. near to zero distortion after optimization.

## References

[1] https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf