CS549 Homework 1: AutoCalib

Yaşar İdikut
yidikut@wpi.edu

***Abstract*—In this homework, I implement a version of Zhang's algorithm for automated camera calibration.**

I. Introduction

Camera calibration is important especially for 3D applications. This process essentially is about calculating the focal lengths, principle points, and distortion coefficients. In this homework, we estimate such parameters and mesause the performance of our implementation through re-projection error for the checkerboard corners.

We use a checkerboard image printed on an A4 paper where the size of each square is 21.5mm. In implementation, as long as the square edge lengths are consistent, for the re-projection metric, this need not be specified.

A. Solving for Approximate K or Camera Intrinsic Matrix

Solving for K relies on calculating the homography between the checkerboard corner locations on the image and the known relative locations of the checkerboard corners. To detect the corners of checkerboard, I apply some image filtering and masking and use the cv2.findChessboardCorners function. This function returns the corners in the image coordinate frame. I further run the cv2.cornerSubPix function to get sub-pixel accuracy for those corner locations. From here, I calculate the homography using the same function as implemented in Project 1.

Using the homography matrix, I calculate the V matrix to construct and equation like so,

$$\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2},\\ h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \ .$$

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \ .$$

$$\mathbf{V}\mathbf{b} = \mathbf{0} \ ,$$

Given above equation, it is trivial to solve for the b matrix using SVD(Singular Value Decomposition). The mapping from this b matrix to the values of the K matrix (Zhang's notation refers to this matrix as A) is pretty trivial as well. This K and b matrices are defined as follows.

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \ .$$

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} .$$

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\ \alpha &= \sqrt{\lambda/B_{11}} \\ \beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\ \gamma &= -B_{12}\alpha^2\beta/\lambda \\ u_0 &= \gamma v_0/\beta - B_{13}\alpha^2/\lambda \ . \end{aligned}$$

B. Estimate Approximate R and t or Camera Extrinsics

Likewise, given the homography and parameters of K, it is trivial to get the extrinsic parameters with the following set of equations.

$$\begin{aligned} \mathbf{r}_1 &= \lambda\mathbf{A}^{-1}\mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda\mathbf{A}^{-1}\mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda\mathbf{A}^{-1}\mathbf{h}_3 \end{aligned}$$

C. Approximate Distortion k and Non-linear Geometric Error Minimization

With intrinsics(K) and extrinsics(R—t) calculated, the last missing matrix is the distortion matrix (k). Since distortion is assumed to be minimal, our initial estimate is a simple 0 matrix. Given these three matrices, we can calculate the re-projection as follows.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} .$$

$$\breve{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$
$$\breve{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] .$$

As can be seen from the table comparing the re-projection errors, this works pretty well. However, we can "fine-tune" our K and k to perform better using some parameter optimization. In this case, the problem is formally represented as follows.

$$\text{argmin}_{f_x, f_y, c_x, c_y, k_1, k_2} \sum_{i=1}^{N} \sum_{j=1}^{M} ||x_{i,j} - \hat{x_{i,j}} \left(K, R_i, t_i, X_j, k \right) ||$$

I used the scipy.optimize.least_squares function and optimized the parameters of K and k to minimize re-projection error.

II. Results

In this section, I present the initial estimated and optimized calibration parameters and re-projection performance for all thirteen images given, however, I only show the first five pictures side-by-side. All of these images were taken from a Google Pixel XL phone with focus locked.

Here are the initial estimated K_{est} (intrinsic) and k_{est} (distortion) matrices.

$$K_{est} = \begin{bmatrix} 2054.52493 & -0.376426223 & 764.572589 \\ 0 & 2039.21767 & 1348.66744 \\ 0 & 0 & 1 \end{bmatrix}$$

$$k_{est} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Here are the optimized estimated K_{est} (intrinsic) and K_{est} (distortion) matrices.

$$K_{opt} = \begin{bmatrix} 2054.52493 & -0.376426223 & 764.572590 \\ 0 & 2039.21767 & 1348.66744 \\ 0 & 0 & 1 \end{bmatrix}$$

$$k_{opt} = \begin{bmatrix} -0.01259005 & 0.09191314 \end{bmatrix}$$

Below is a table comparin the re-projection errors for estimated and optimized calibration parameters.

Re-projection Error for initial estimated parameters (L1 Norm)	0.7575416096971362
Re-projection Error for initial optimized parameters (L1 Norm)	0.7436039624193288

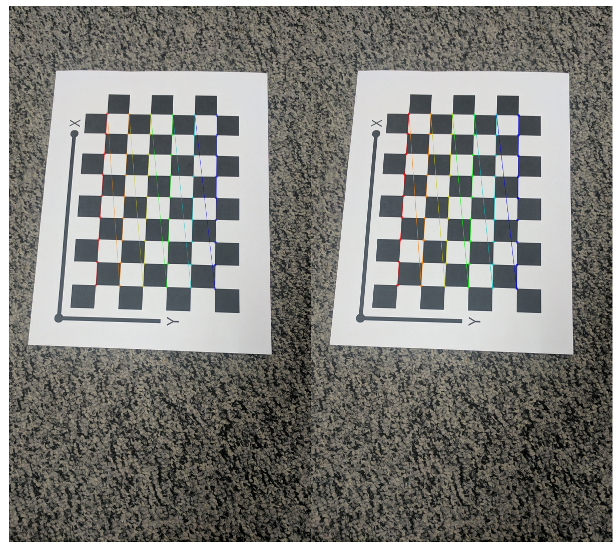

Fig. 1: Image 0 corners (left) and re-projected corners (right) visualized

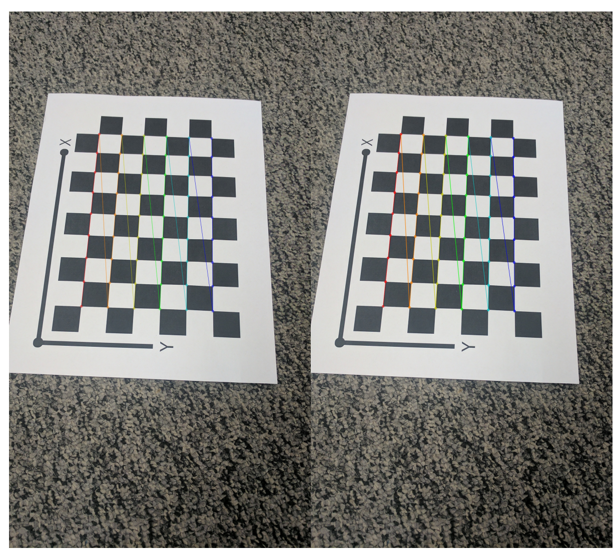

Fig. 2: Image 1 corners (left) and re-projected corners (right) visualized

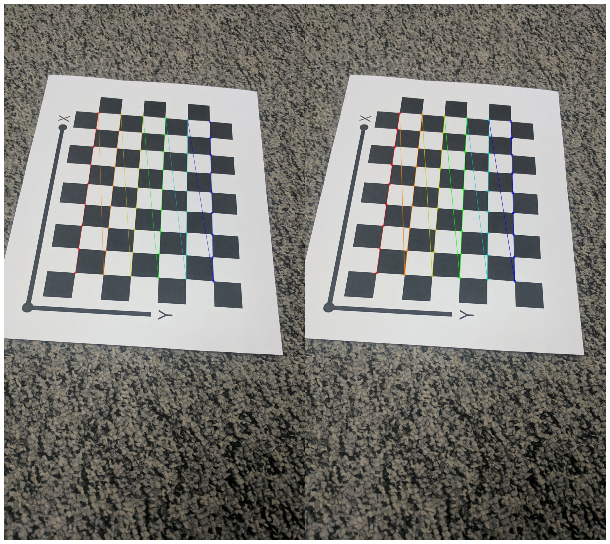

Fig. 3: Image 2 corners (left) and re-projected corners (right) visualized

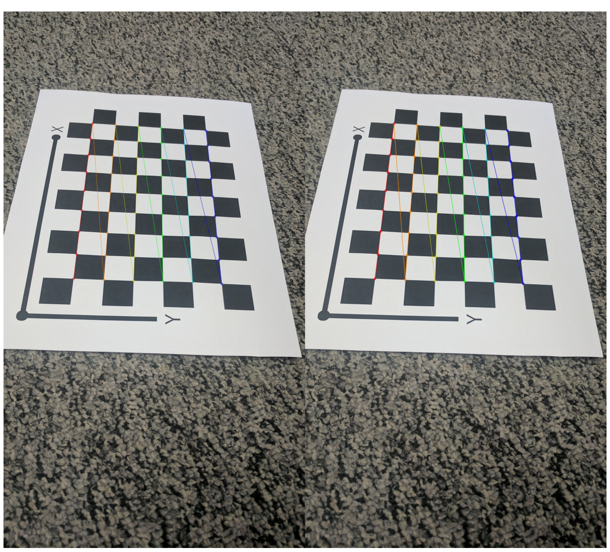

Fig. 4: Image 3 corners (left) and re-projected corners (right) visualized

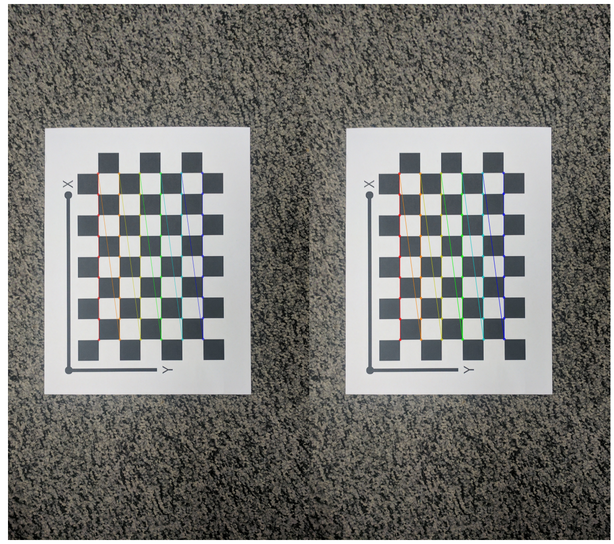

Fig. 5: Image 4 corners (left) and re-projected corners (right) visualized

III. Discussion and Conclusion

As can be seen from the side-by-side comparisons and low re-projection error, the implementation of this automated calibration algorithm performs well. The initial estimated parameters work well, and the further optimization conducted influenced the re-projection error only slightly. This is probably the camera used already had almost no distortion.

References

[1] Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.