AutoCalib

John Kanu
PhD Student, Computer Science, UMD
Email: jdkanu@cs.umd.edu

USING 2 LATE DAYS

***Abstract*—In this paper I present a method for automatically calibrating a camera, based on Zhang's method.**

I. Initial Parameter Estimation

The objective is to determine the parameters K, R, t, k_s that minimizes reconstruction error. This is a non-linear, geometric error minimization. The least-squares problem cannot be solved analytically in closed form, or by linear least-squares. So I initialize the parameters using an estimation, which is then fed into an optimizer in the `scipy.optimize` package to return our result.

A. Camera Intrinsic Matrix

The initial estimate for the camera intrinsic matrix is determined using SVD to solve the equation

$$Vb = 0$$

, where variables V and b are defined in Zhang's method.

For each pair of model and camera image, I compute the 3-D and 2-D positions of camera points in each image, respectively. The 3-D positions for model points are assumed to have a value of 0 in the Z component. All points are normalized by mean and standard deviation, which in theory improves the performance of the algorithm. The equations are stacked in $Vb = 0$, and singular-value decomposition (SVD) is applied to solve for b. The parameters of K are then uniquely defined based on the values of b, as given in Appendix B of Zhang's paper.

B. Camera extrinsics

The extrinsic view parameters are computed in a few steps after the estimation of the intrinsic parameters. Given the calibration matrix K and the homography matrix H, a scale parameter λ is computed as

$$\lambda = \frac{1}{||A^{-1} \cdot h_0||} = \frac{1}{||A^{-1} \cdot h_1||}$$

and columns r_0, r_1, and r_2 of R are computed as

$$r_0 = \lambda \cdot A^{-1} \cdot h_0$$

$$r_1 = \lambda \cdot A^{-1} \cdot h_1$$

$$r_2 = r_0 \times r_1$$

$$t = \lambda \cdot A^{-1} \cdot h_2$$

C. Distortion parameter

The distortion parameter is initialized as $k_c = [0, 0]^T$ under the assumption that the camera has minimal distortion.

D. Non-linear Geometric Error Minimization

Using `scipy.optimize` I minimize the geometric error defined by

$$\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||u_{i,j} - P(A, k, R, t, X_j)||^2$$

where $u_{i,j}$ are the observed 2-D image points, X_j are the 3-D model points, and P is the projection defined by these parameters from the model space to the image space.

II. Results

A. Camera intrinsic matrix

$$K = \begin{bmatrix} 694.97446097 & -2.1214567 & 248.71275362 \\ 0 & 691.17535932 & 449.23440652 \\ 0 & 0 & 1 \end{bmatrix}$$

B. Reprojection Error

I compute reprojection error given image points $u_{i,j}$ and projected model points $p_{i,j}$ as

$$\text{Error} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |u_{i,j} - p_{i,j}|}{M \cdot N} = 0.539032299237$$

C. Rectified images

Fig. 1: IMG_20170209_042606.jpg

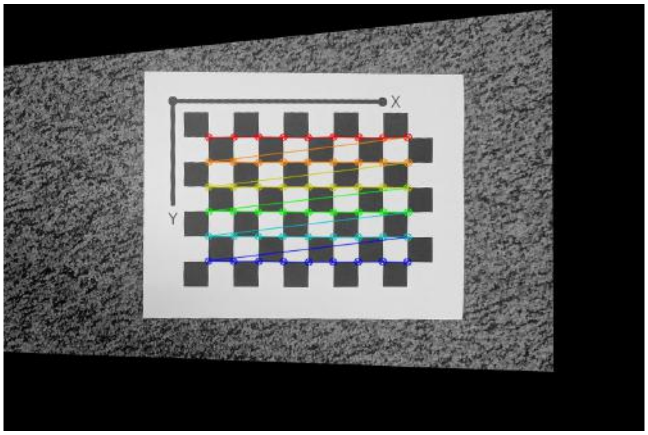

Fig. 2: IMG_20170209_042608.jpg

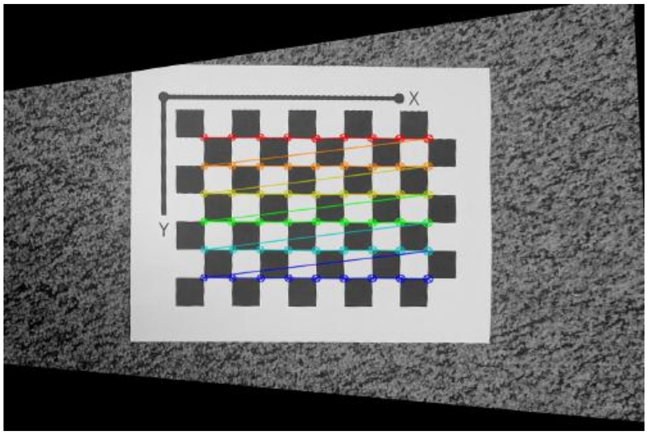

Fig. 3: IMG_20170209_042610.jpg

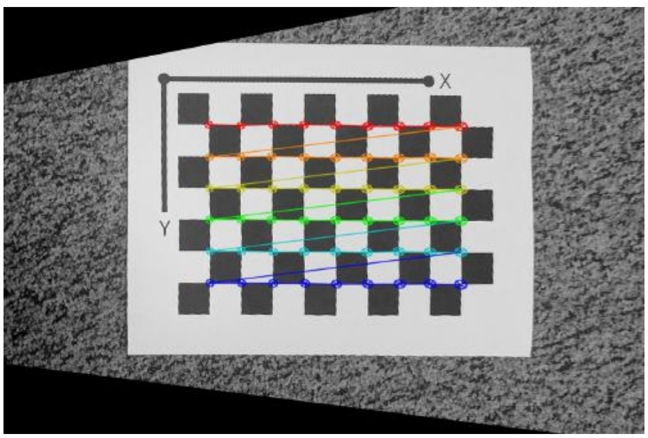

Fig. 4: IMG_20170209_042612.jpg

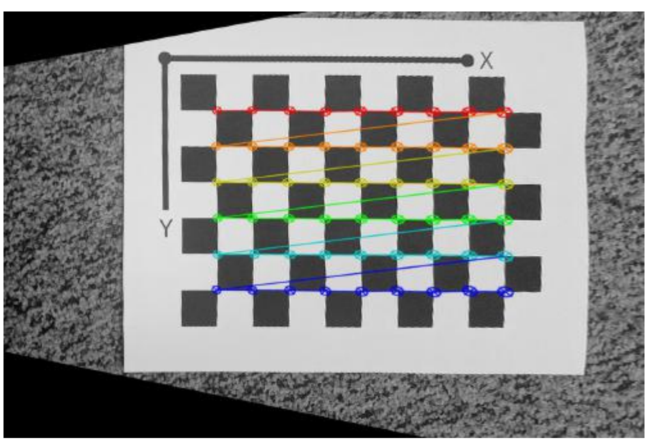

Fig. 5: IMG_20170209_042614.jpg

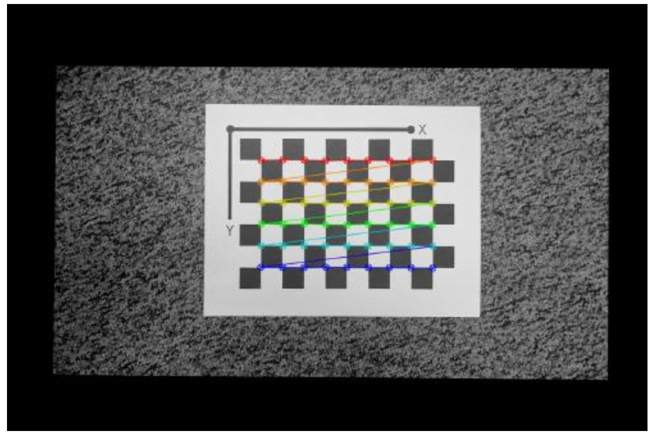

Fig. 6: IMG_20170209_042616.jpg

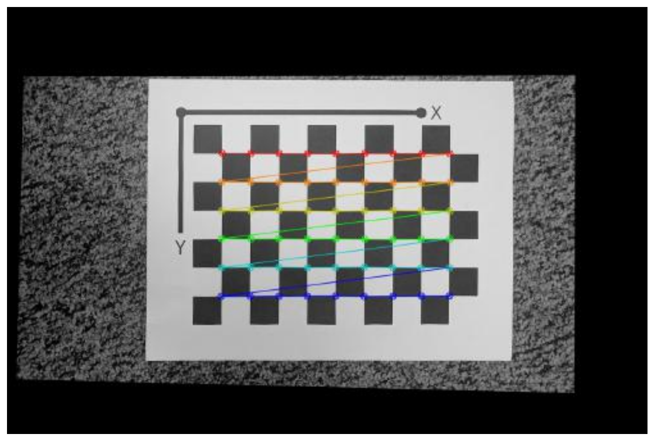

Fig. 7: IMG_20170209_042619.jpg

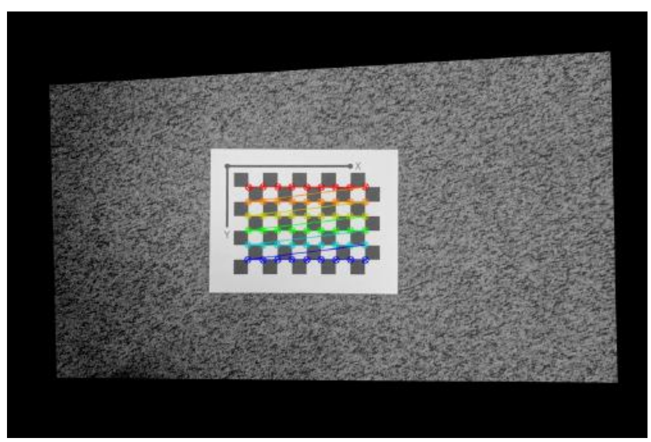

Fig. 8: IMG_20170209_042621.jpg

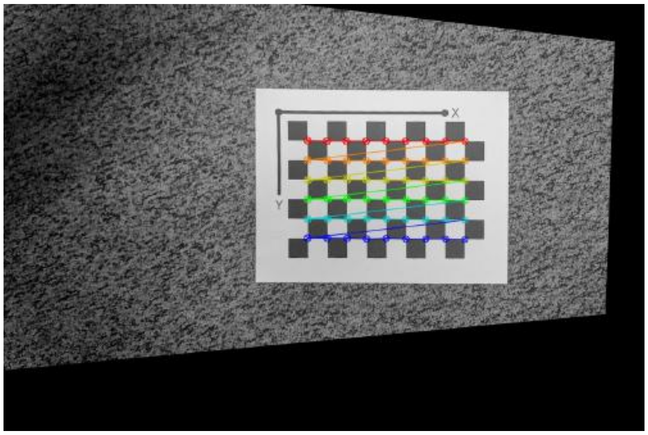

Fig. 9: IMG_20170209_042624.jpg

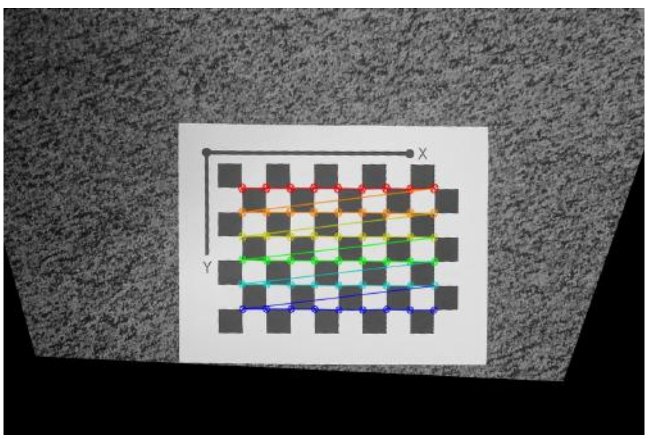

Fig. 10: IMG_20170209_042627.jpg

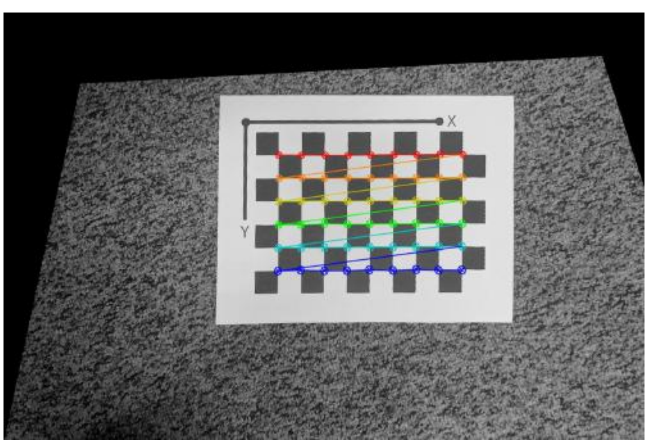

Fig. 11: IMG_20170209_042629.jpg

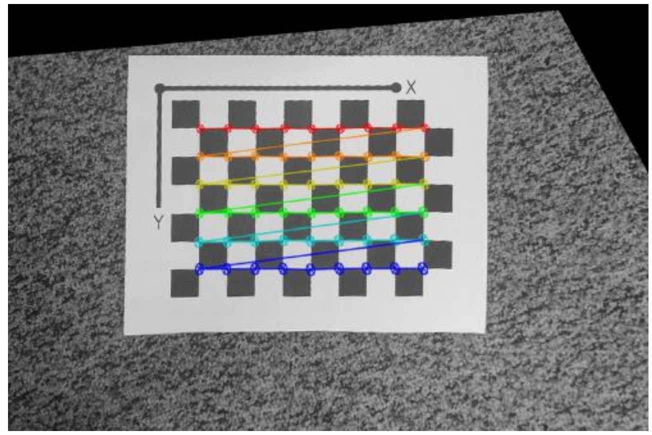

Fig. 12: IMG_20170209_042630.jpg

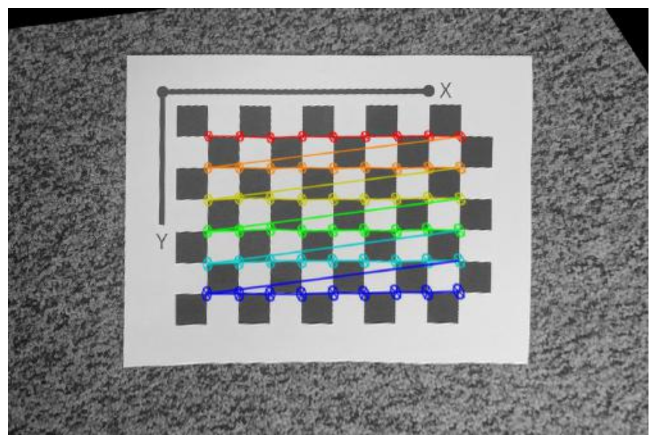

Fig. 13: IMG_20170209_042634.jpg