Auto Calibration

Gokul Hari
Directory ID: hgokul@umd.edu

I. Auto Calibration

Computing point correspondences

To implement Zhang's approach to perform camera calibration, I need to obtain the real world- image point correspondences of a set of checkerboard images. Neglecting the outer squares in the checker board, I have 9×6 points spaced at 21.5mm. Thus I can generate a (x,y) mesh grid of world coordinates for each corner point numbered from 0 to 5 row-wise and 0 to 8 numbered column-wise, totalling upto 54 points (n_p). These world points are denoted as M. Now, that I computed the world points, I computed the image coordinates of the checkerboard using opencv's inbuilt function - cv2.findChessboardCorners, and ensured that the order of all the n_p point correspondences between m and M are matched across all n_i images in the dataset provided.

Solving for approximate intrinsics

The relationship between the point correspondences m and M is given as ,

$$m = K \begin{bmatrix} R & t \end{bmatrix} M \tag{1}$$

where K is the intrinsic camera matrix given by,

$$K = \begin{bmatrix} \alpha & \gamma & u0 \\ 0 & \beta & v0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2}$$

I followed the steps mentioned in section 3.1 [1] to compute the V_i matrix using the homography H_i estimated between the point correspondences m_i and M_i, where i denotes the images from 0-n_i. I stacked the n_i (13, in our case) V_i matrices to form the V matrix of shape $2n_i \times 6$, and solved the homogenous equation V.b = 0, where b = $[B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]$. The elements are formulated from,

$$B = K^{-T}K = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \tag{3}$$

where B is a symmetric matrix. The K matrix can be computed from B in two ways,

- By computing the α, β, γ, u0, v0 values by substituting them in the equations shown in appendix B of [1]
- By performing cholesky decomposition of B to obtain the K^{-1} and performing inverse of it yields K.

However, the estimated intrinsic matrix K is an initial unoptimized estimate and let it be denoted as K_{init}

Solving for approximate Extrinsics

To compute the extrinsic parameters, the rotation and translation of the camera, I followed the steps suggested in [1] section 3.1 to compute the rotation vectors r1, r2, r3, and the translation vector t, to form RT_{init_i} using K_{init}^{-1} and H_i for every i-th image in the data set of n_i images.

Thus, I had computed an initial estimate of the intrinsic camera parameters K_{init} and the Extrinsic parameters RT_{init_i} for all n_i images. Using these parameters, I can reproject the world coordinates (M_i) to the image plane, represented by 'reprojected' image coordinates $\hat{m}_i$

Projecting world coordinates M_i

The estimated intrinsic parameters K_{init}, the extrinsic parameters RT_{init_i} can be used to project the world coordinates (M_i) to camera coordinates $\hat{m_i}$, for a given image i in all of n_i images. Steps to perform the projection is as follows,

- First, convert the world coordinates $M_{(x,y)}$ to homogenous coordinates of $M_{(x,y,0,1)}$ of shape 1×4
- Compute X' = $RT_{3\times4}.M_{(x,y,0,1)}$, where X' = [x',y',z] and $RT_{3\times4}$ is the extrinsic matrix
- Compute ideal normalized image coordinates X = [x,y,1] as x = $\frac{x'}{z}$, and y = $\frac{y'}{z}$
- Compute radius of distortion r = $(x^2 + y^2)^{\frac{1}{2}}$
- Compute ideal pixel image coordinates U' = $K_{3\times3}.X$, where U' = [u',v',w]
- Compute U = [u,v,1] as u = $\frac{u'}{w}$, and v = $\frac{v'}{w}$
- Compute image coordinates $\hat{m} = [\hat{u}, \hat{v}]$, $\hat{u}$ and $\hat{v}$ given by equations 4, 5

$$\hat{u} = (u - u0)(k_1r^2 + k_2r^4) \tag{4}$$

$$\hat{v} = (v - v0)(k_1r^2 + k_2r^4) \tag{5}$$

where k1 and k2 are the radial distortion coefficients. For the initially estimated parameters K_{init} and RT_{init_i}, I assumed kC = $(k1, k2)^T$ = $(0, 0)^T$

The coordinates $\hat{m}$ are image coordinates reprojected using our camera calibration parameters. To know the accuracy of our intrinsic and extrinsic parameters, we can compute the reprojection error ρ_{error} between m_i and $\hat{m}_i$.

Reprojection Error

The reprojection error is computed as,

$$\rho_{error} = \frac{1}{n_i}\frac{1}{n_p}\sum_{i=1}^{n_i}\sum_{j=1}^{n_p} \|m_{ij} - \hat{m_{ij}}\| \tag{6}$$

```
Initially estimated K matrix :
 [[ 2.05610659e+03 -1.01710000e+00  7.61655250e+02]
 [ 0.00000000e+00  2.04050404e+03  1.35130849e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]

Distortion Coordinates before optimization: [0 0]
Projection error before optimization : 0.69824


optimized K matrix :
 [[ 2.05610347e+03 -1.01766000e+00  7.61661340e+02]
 [ 0.00000000e+00  2.04049476e+03  1.35132280e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]

Distortion Coordinates after optimization: [ 0.0139  -0.10302]
Projection error after optimization : 0.68136
```

Fig. 1. Results

where m_{ij} is the actual image coordinates and $\hat{m}_{ij}$ is the reprojected image coordinates for every i-th image and j-th corner point in image i . The mean reprojection error for the un-optimized parameters K_{init} in 1, kC = $[0, 0]^T$ and corresponding extrinsics for n_i images , was found to be **0.69824**

Non linear Geometric Error minimization

To remove lens distortion, we need to get an optimized estimate of the intrinsics, extrinsic parameters and distortion coefficients. This optimization was done with scipy.optimize function with the lm (Levenberg-Marquardat) method . The parameters $\alpha, \beta, \gamma, u0, v0, k1, k2$ were to be optimized. For this optimization function, I wrote a loss function that returns an error vector e_ρ = $[\rho_1, \rho_2, \rho_3 ... \rho_{n_i}]$

$$\rho_i = \sum_{j=1}^{n_p} \|m_{ij} - \hat{m}_{ij}\| \quad (7)$$

where, ρ_i is the sum of L2 norm of difference between all the actual image coordinates and reprojected image coordinates for a given image i. It is necessary that the length of the error vector e_ρ must be greater than the number of parameters to be optimized, therefore, this loss function requires $n_i > 7$ (since there are 7 parameters to optimize) In our case, n_i = 13 since 13 images are present.

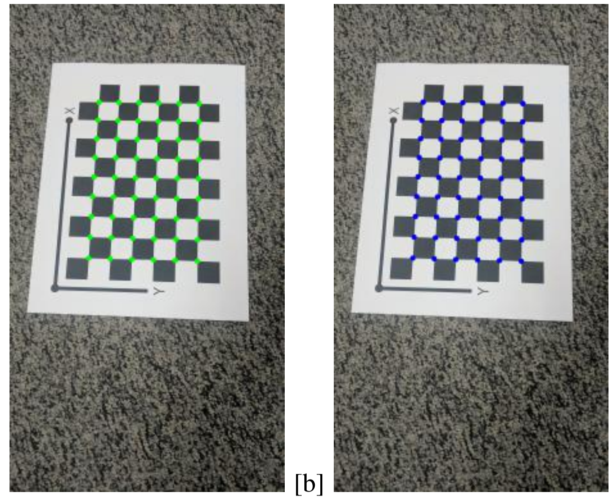
The optimized distortion coefficients are given as kC = $[0.0139, -0.10302]^T$. The optimized intrinsic parameter, K matrix is given in 1 The reprojection error after optimization is found to be **0.68136**

II. Results

After performing the calibration, and having obtained the optimized extrinsic parameters RT_i, the rectified checkerboard images with reprojected points is shown in figure 2. The images before and after the calibration are shown below

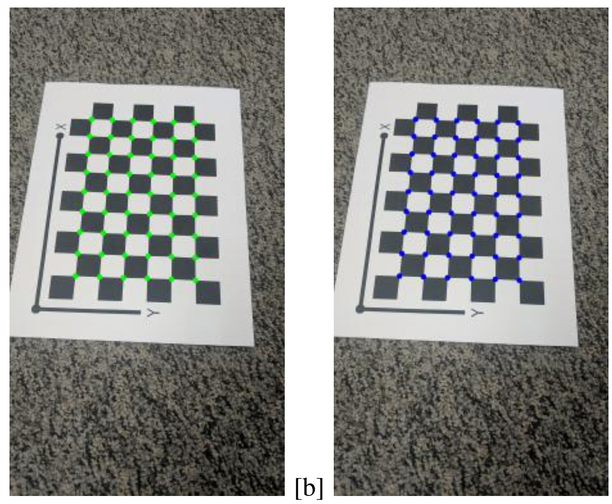
References

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



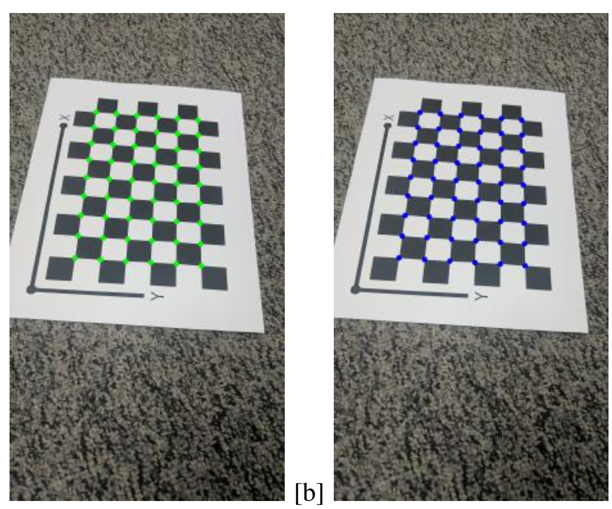
[a] [b]

Fig. 2. a) detected points in raw image 1 b) reprojected points in rectified image 1



[a] [b]

Fig. 3. a) detected points in raw image 2 b) reprojected points in rectified image 2



[a] [b]

Fig. 4. a) detected points in raw image 3 b) reprojected points in rectified image 3

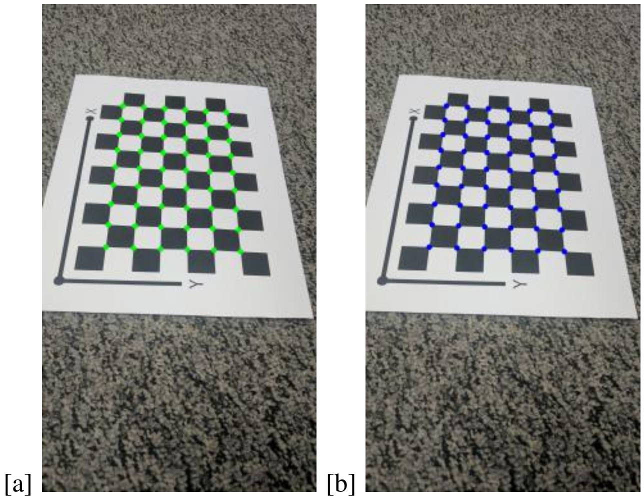


Fig. 5. a) detected points in raw image 4 b) reprojected points in rectified image 4

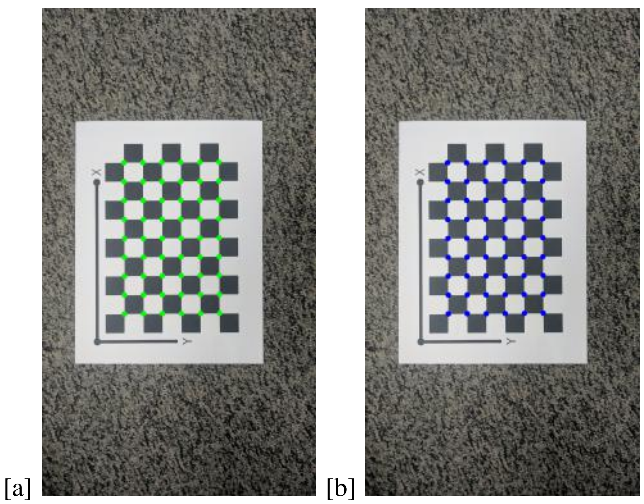


Fig. 6. a) detected points in raw image 5 b) reprojected points in rectified image 5

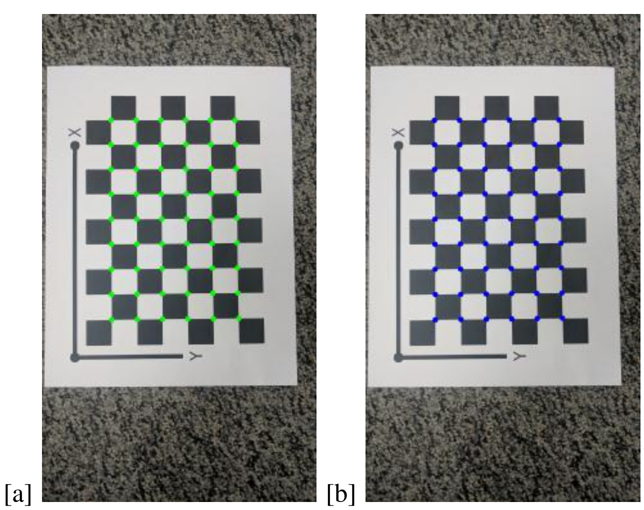


Fig. 7. a) detected points in raw image 6 b) reprojected points in rectified image 6

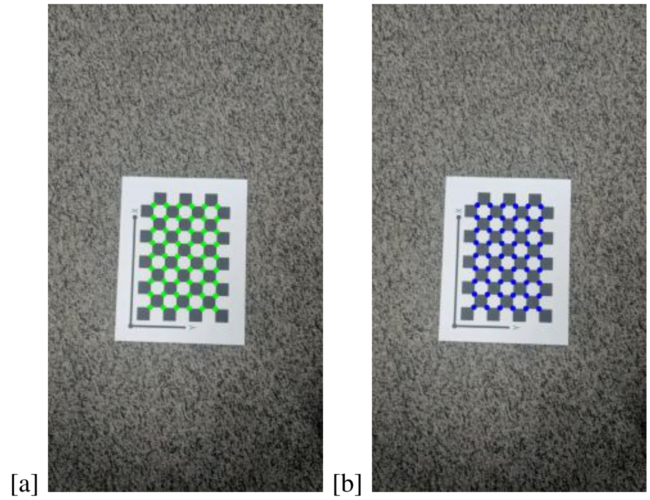


Fig. 8. a) detected points in raw image 7 b) reprojected points in rectified image 7

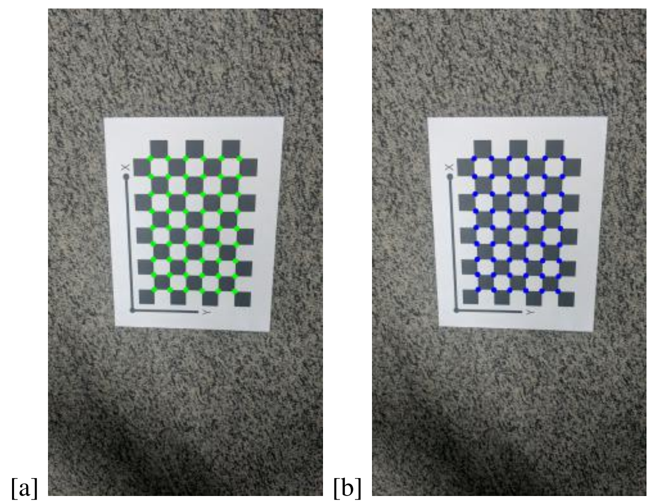


Fig. 9. a) detected points in raw image 8 b) reprojected points in rectified image 8

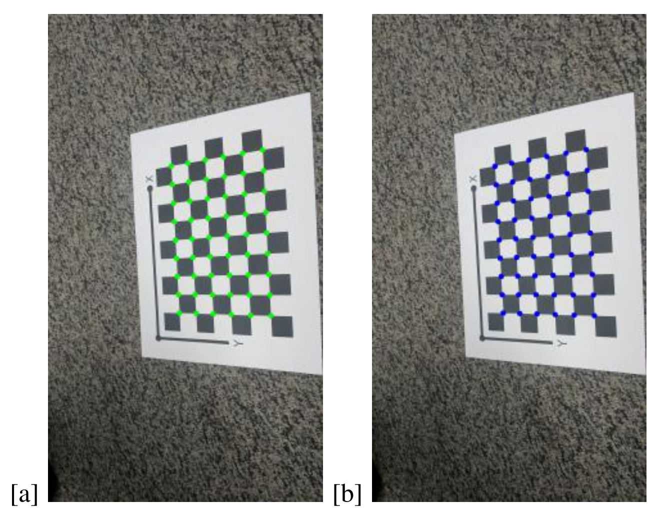
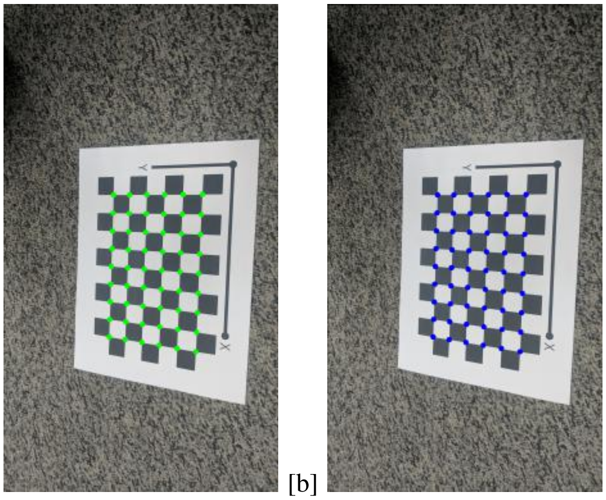
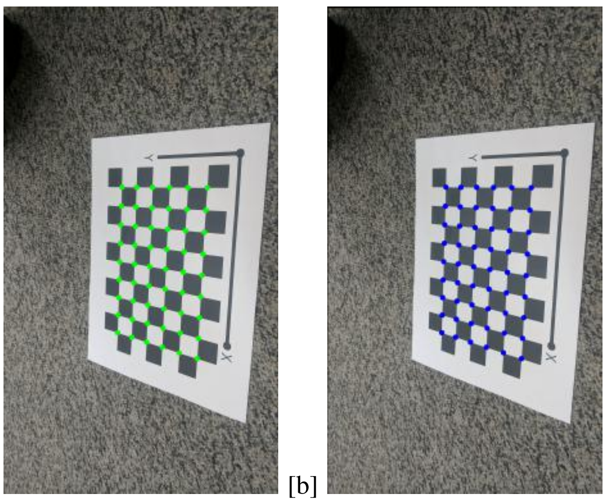


Fig. 10. a) detected points in raw image 9 b) reprojected points in rectified image 9



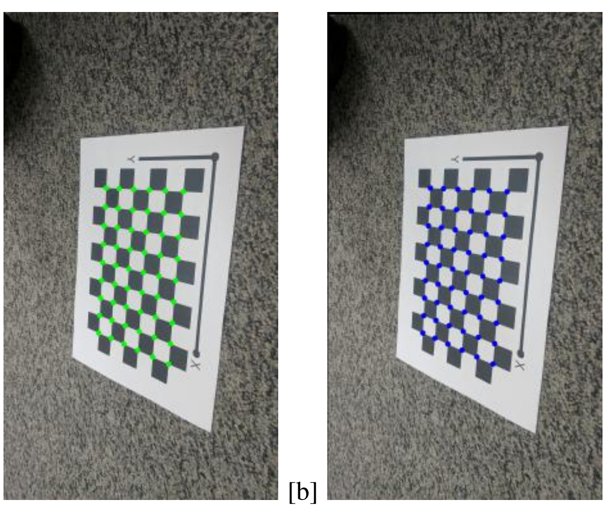
[a] [b]

Fig. 11. a) detected points in raw image 10 b) reprojected points in rectified image 10



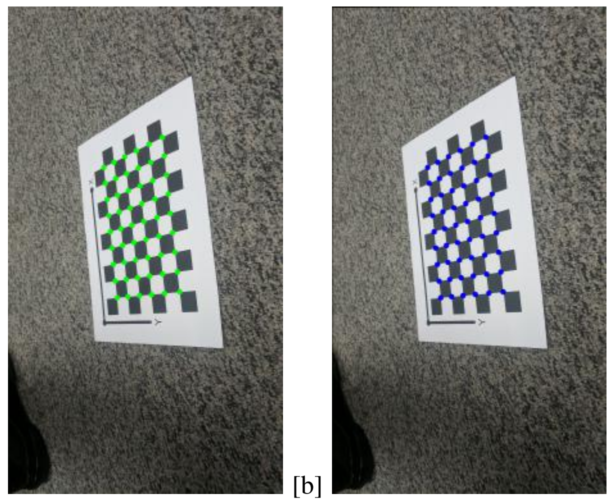
[a] [b]

Fig. 12. a) detected points in raw image 11 b) reprojected points in rectified image 11



[a] [b]

Fig. 13. a) detected points in raw image 12 b) reprojected points in rectified image 12



[a] [b]

Fig. 14. a) detected points in raw image 13 b) reprojected points in rectified image 13