CMSC733: HW 1 - Auto Calib

Gnyana Teja Samudrala
Email: sgteja@terpmail.umd.edu

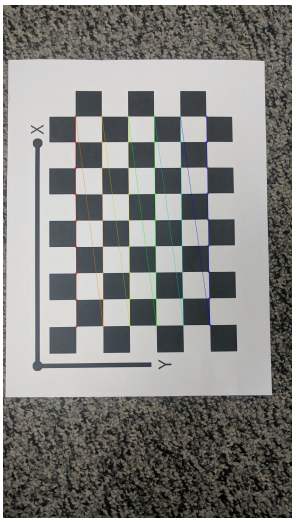

Fig. 1. Corners detected using findChessboardCorners on 1.

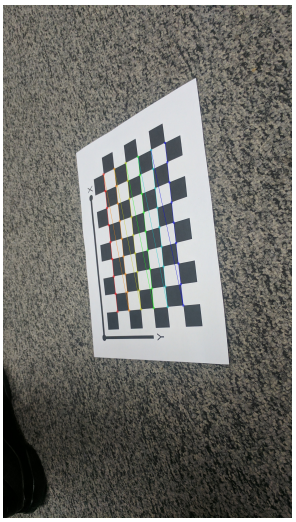

Fig. 2. Corners detected using findChessboardCorners on 2.

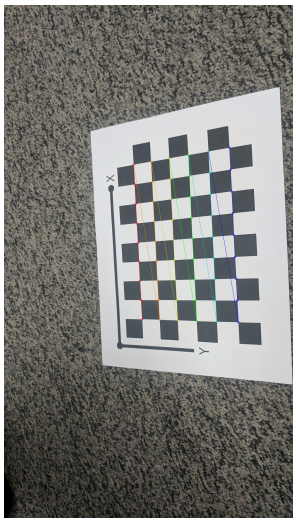

Fig. 3. Corners detected using findChessboardCorners on 3.

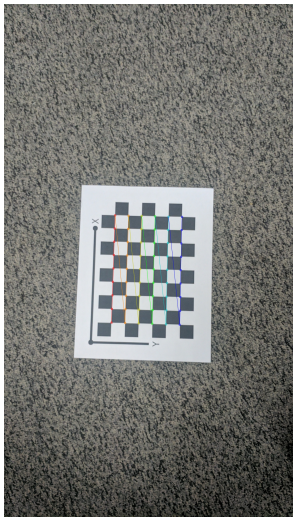

Fig. 4. Corners detected using findChessboardCorners on 4.

I. Initial Estimation

The corners from the given images is retrived and then the homography is estimated using the eigen vectors method as given in the Zhang's paper [1] [**?**]. The corners in the shape of $9X6$ are considered for the estimation. So there will be a total of 54 points for each image. So the size of the matrix L in the equation ($Lx = 0$) whose solution will yield the homography matrix of the image will be $108X9$. The corner points detected are shown in the images 1- 13

A. Intrinsic parameters

Once the homography matrix for every image is calculated, then we find the initial estimate for the intrinsic paramters. This is done by using the closed form solution in section 3.1 of the Zhangs paper. We estimate a matrix V from which we retrieve the extrinsic and intrinsic parameters. As we have more than 3 images we can get the unique solution and a good estimate for gamma. The intrinsic parameter matrix obtained in this step is shown in equation. 1

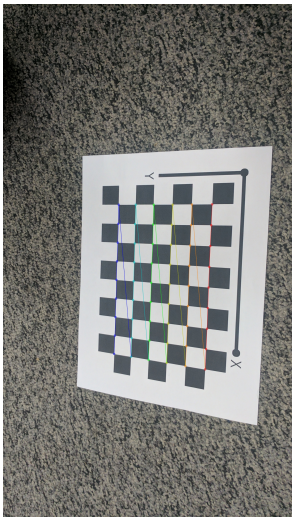

Fig. 5. Corners detected using findChessboardCorners on 5.

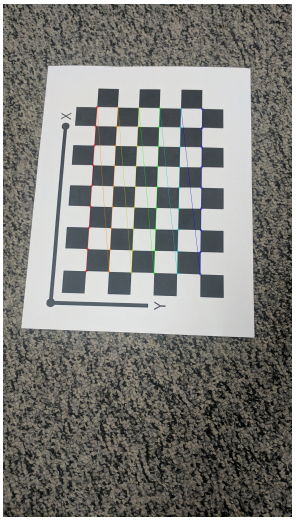

Fig. 6. Corners detected using findChessboardCorners on 6.

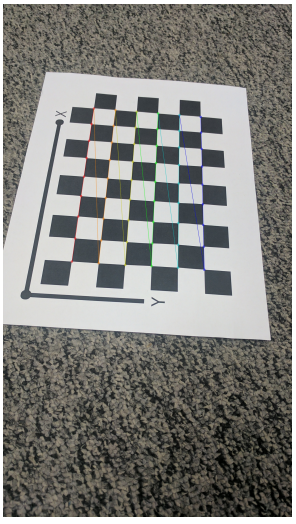

Fig. 7. Corners detected using findChessboardCorners on 7.

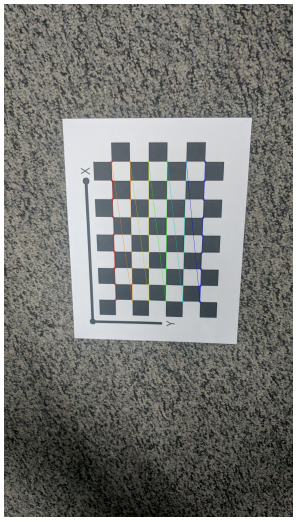

Fig. 8. Corners detected using findChessboardCorners on 8.

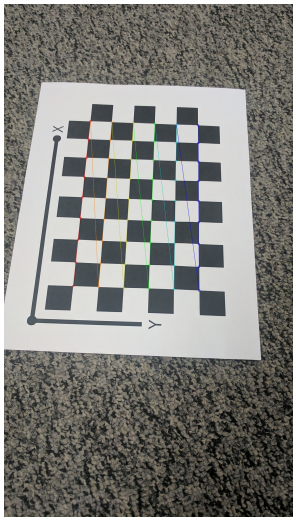

Fig. 9. Corners detected using findChessboardCorners on 9.

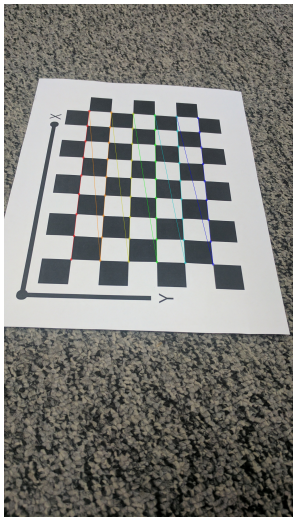

Fig. 10. Corners detected using findChessboardCorners on 10.

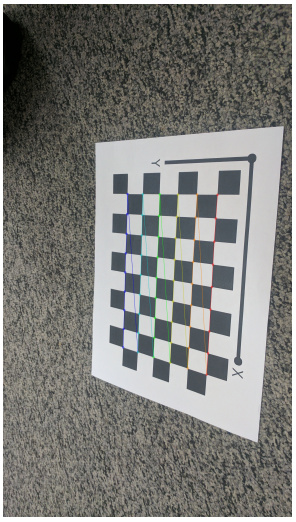

Fig. 11. Corners detected using findChessboardCorners on 11.

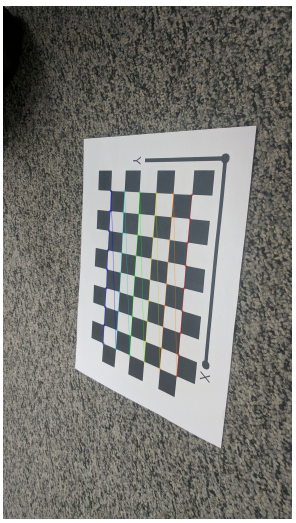

Fig. 12. Corners detected using findChessboardCorners on 12.

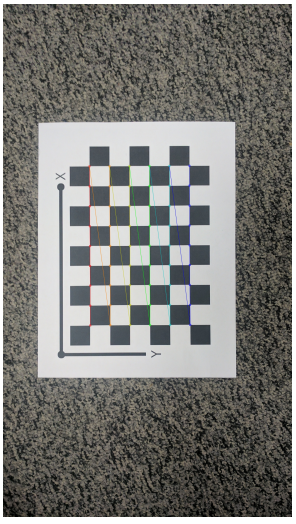

Fig. 13. Corners detected using findChessboardCorners on 13.

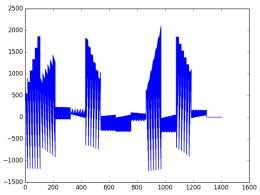

Fig. 14. The plot of the function before optimisation.

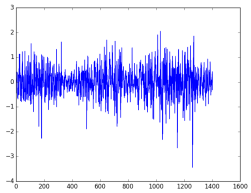

Fig. 15. The plot of the function after optimisation.

$$A_{init} = \begin{bmatrix} 2240.75 & -117.19 & 128.93 \\ 0 & 2269.6 & 1268.78 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

B. Extrinsic parameters

Also the extrinsic parameters which are 3 rotation and a translation vector are calculated using the camera matrix and the homography matrix. We get a rotation matrix $(3X3)$ and translation vector $(3X1)$ for each image.

C. Distortion coefficients

The distortion coefficients k_1 and k_2 are assumed to be zero for the inital estimate.

II. Non-linear Estimation

All the parameters estimated in the previous sections are given to the non linear estimator and then optimised results are obtained. The scipy optimise library is used, in which least squares Trust Region Reflective algorithm is used. This algorithm is selected to apply the boundary conditions given. This idea of the code is taken from the example given in the scipy library [2] [**?**]. The conditions were imposed to so that the values in the rotation matrix lie in the range of -1,1 and the translation in z-axis is positive. The plots of the initial error and optimised values are seen in the figures. 14 and 15

The world points are projected back onto the images using the optimized values, these images can be seen from 16 - 13. The reprojection error before optimisation is 415.744 and after optimisation 0.4305. The optimised camera matrix is as in equation. 2

$$A_{final} = \begin{bmatrix} 2218.203 & -137.19 & 238.21 \\ 0 & 2250.2 & 1213.56 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

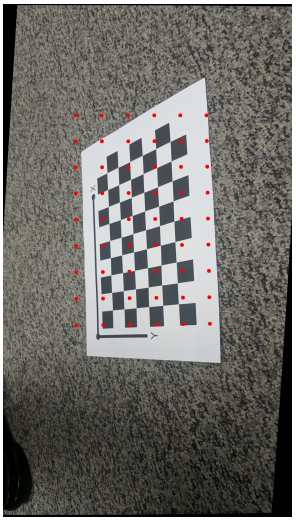

Fig. 16. Corners reprojected on 1.

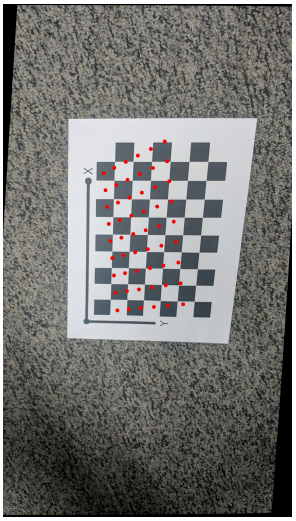

Fig. 17. Corners reprojected on 2.

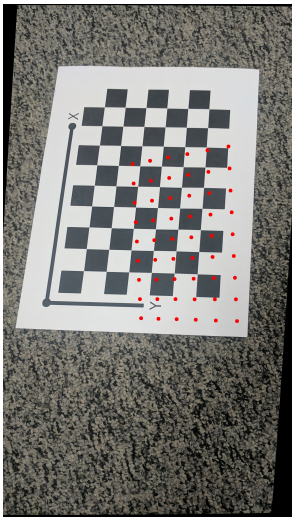

Fig. 18. Corners reprojected on 3.

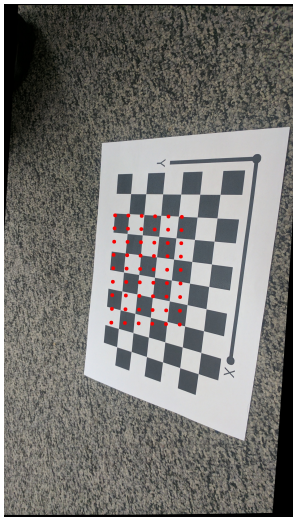

Fig. 19. Corners reprojected on 4.

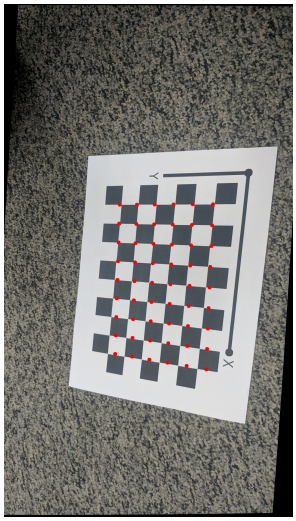

Fig. 20. Corners reprojected on 5.

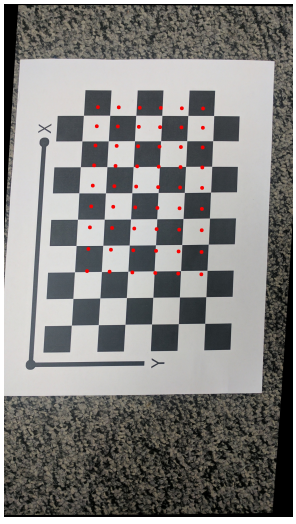

Fig. 21. Corners reprojected on 6.

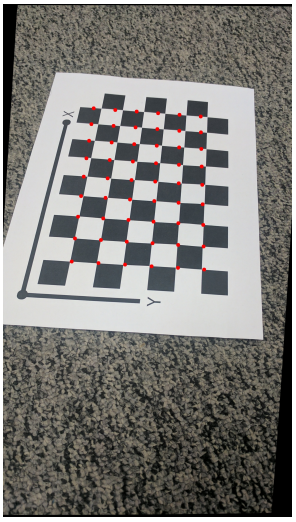

Fig. 22. Corners reprojected on 7.

Fig. 23. Corners reprojected on 8.

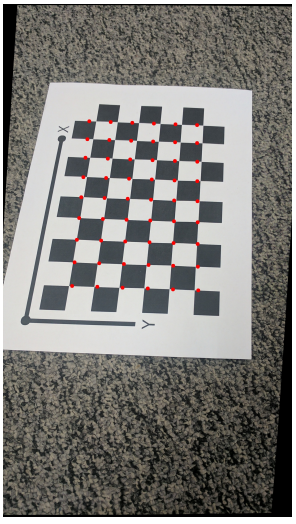

Fig. 24. Corners reprojected on 9.

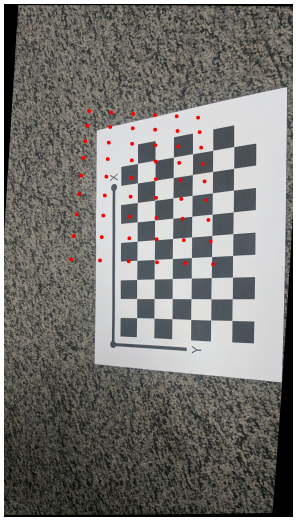

Fig. 25. Corners reprojected on 10.

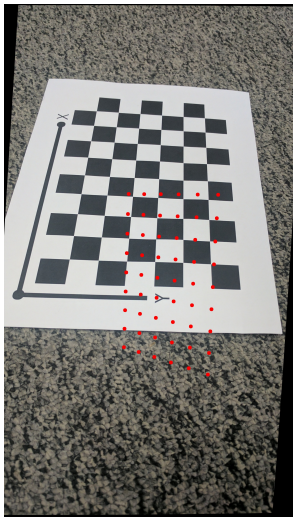

Fig. 26. Corners reprojected on 11.

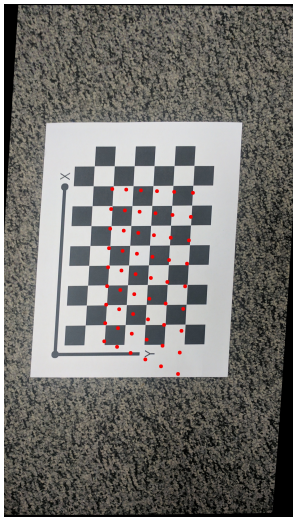

Fig. 27. Corners reprojected on 12.

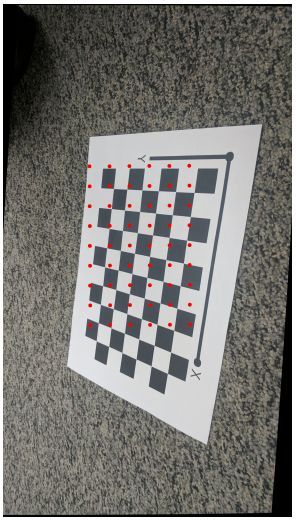

Fig. 28. Corners reprojected on 13.

REFERENCES

[1] Z. Zhang, "A flexible new technique for camera calibration," 1998. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf

[2] N. Mayrov, "Large-scale bundle adjustment in scipy." [Online]. Available: https://scipy-cookbook.readthedocs.io/items/bundle$_adjustment.html$