Homework 1: AutoCalib

(Using 1 Late Day.)

Rohith Jayarajan (115458437)
University of Maryland
College Park, Maryland 20740
rohith23@umd.edu

***Abstract*—In this work, the technique of camera calibration as described in Zhengyou Zhang's 1998 paper titled "A Flexible New Technique for Camera Calibration" is studied and implemented. Using at least three images of a calibration target with known measurements, the camera intrinsics are estimated.**

I. Introduction

For any computer vision research where 3D geometry is involved, estimating the the camera intrinsic parameters is an inevitable and important task. Camera intrinsic parameters comprise of focal lengths fx, fy, principle point cx, cy, skew and distortion parameters. One of the best ways to perform camera calibration automatically was presented by Zhengyou Zhang of Microsoft in 1998 [1]. The camera calibration matrix is denoted by **K** and is of the form shown below

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

and the radial distortion parameters are $\mathbf{k1} and \mathbf{k2}$.

II. Data Preparation

The method in Zhangs paper uses a calibration target for camera instrinsic parameter estimation. In this work, the calibration target is a checkerboard of dimensions 9 inner rows and 6 inner columns. Each square in the checkerboard pattern has an edge length of 21.5mm. The print of the checkerboard was taken on an A4 paper using a Google Pixel XL phone camera with locked focus. The calibration target is shown in figure 15.

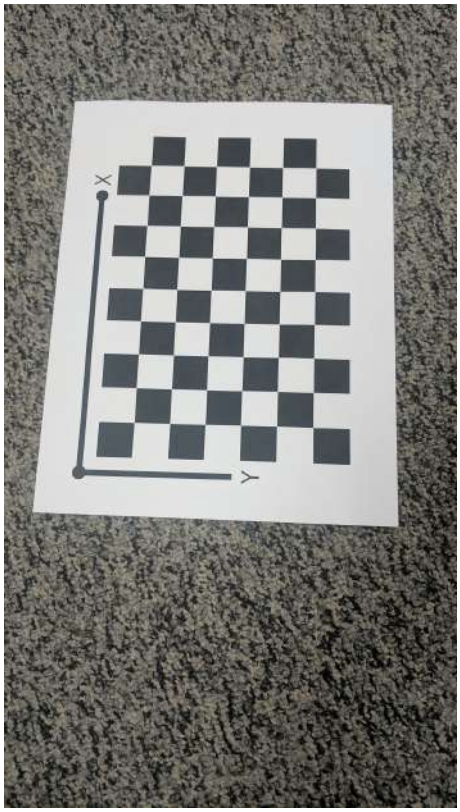

Fig. 1. Calibration Target.

III. Implementation

A. Acquiring Image and World Points

The image points (pixel coordinates) are denoted by x, world points (checkerboard points) by X, radial distortion parameters by k_s, camera calibration matrix by K,rotation matrix and the translation of the camera in the world frame by R and t respectively.

To detect the corners of the checkerboard pattern, the OpenCV function *cv2.findChessboardCorners()* was used and the corners were refined by the function *cv2.cornerSubPix()*. The corners were detected in the checkerboard pattern and an example is shown in figure 2

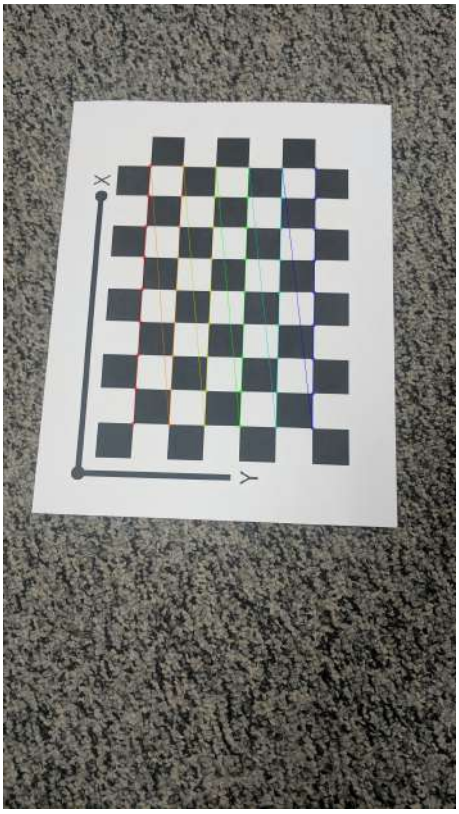

Fig. 2. Detected corners.

B. Solving for Approximate K or Camera Intrinsic Matrix

Without loss of generality, the model plane is assumed to be on $Z = 0$ axis. The relationship between the image coordinate and the 3D world coordinate simplifies to equation 2.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{2}$$

where s is a scale factor, r_i is the i^{th} rotation column vector and t is the translation vector.

For the closed form solution, a vector b which is shown below is used.

$$b = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix} \tag{3}$$

where B_{ij} is the element of matrix B which is described in Section 3.1 equation 5 in Zhang's paper.

The solution of b in $Vb = 0$ which is described in Section 3.1 equation 8 in Zhang's paper gives us the camera intrinsic matrix A or K.

C. Estimate Approximate R and t or Camera Extrinsics

Let the i_{th} column of the homography matrix **H** be h_i. Once the value of camera matrix K is computed, the extrinsic parameters for each image can be computed using the following equations given below

$$\begin{aligned} r_1 &= \lambda A^{-1} h_1, \\ r_2 &= \lambda A^{-1} h_2, \\ r_3 &= r_1 \times r_2, \\ t &= \lambda A^{-1} h_3 \end{aligned}$$

with $= 1/A^{-1}h_1$. The matrix $R = [r_1, r_2, r_3]^T$

D. Approximate Distortion k_c

A value of $k_c = [0, 0]^T$ is considered a good initial estimate.

E. Non-Linear Geometric Error Minimization

Once we have the initial estimates for K, R, t, k_c, we minimize the below geometric error in equation 4 using $scipy.optimize.least_squaressolver$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \|x_{ij} - x_{ij}(K, R, \hat{t}_i, Xj, k_c)\| \tag{4}$$

IV. Results

On successful implementation of the technique mentioned in Zhang's paper, the following results were obtained

A. Initial Estimate of B Matrix

$$\begin{bmatrix} -1.50143134e-07 \\ -8.28742618e-11 \\ -1.52412971e-07 \\ 1.14436171e-04 \\ 2.05856610e-04 \\ -9.99999972e-01 \end{bmatrix}^T \tag{5}$$

B. Initial Estimate of A Matrix

$$\begin{bmatrix} 2.05637858e+03 & -1.12657240e+00 & 7.61435224e+02 \\ 0.00000000e+00 & 2.04100896e+03 & 1.35023618e+03 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix} \tag{6}$$

C. Initial Estimate of k_c Matrix

$$\begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{7}$$

D. Refined Estimate of A Matrix

$$\begin{bmatrix} 2.05637631e+03 & -1.12822620e+00 & 7.61440990e+02 \\ 0.00000000e+00 & 2.04100604e+03 & 1.35026685e+03 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix} \tag{8}$$

E. Refined Estimate of k_c Matrix

$$\begin{bmatrix} 0.00565324 & -0.03628325 \end{bmatrix}^T \tag{9}$$

F. Mean Reprojection Error

The mean reprojection error after calibration was **0.693026198997**.

The reprojection of the points are shown in the figure 3. This gives an excellent visual representation of the successful calibration.

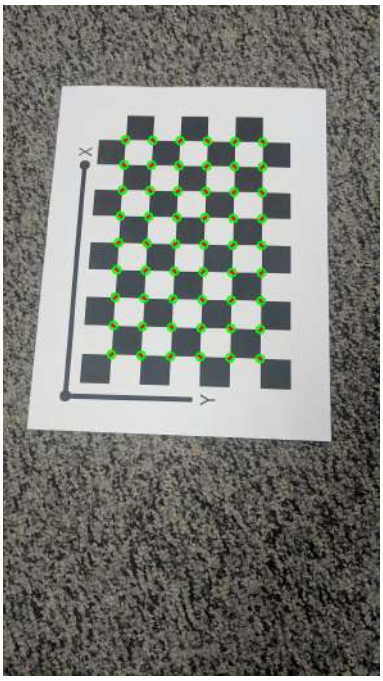

Fig. 3. Reprojection of points. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

V. Conclusion

The technique specified in Zhang's paper was successfully studied and implemented to calibrate a monocular camera with a reprojection error of 0.693026198997.

References

[1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.888718

APPENDIX A
APPENDIX

The reprojection 3D points in the other images in the given dataset are shown below.

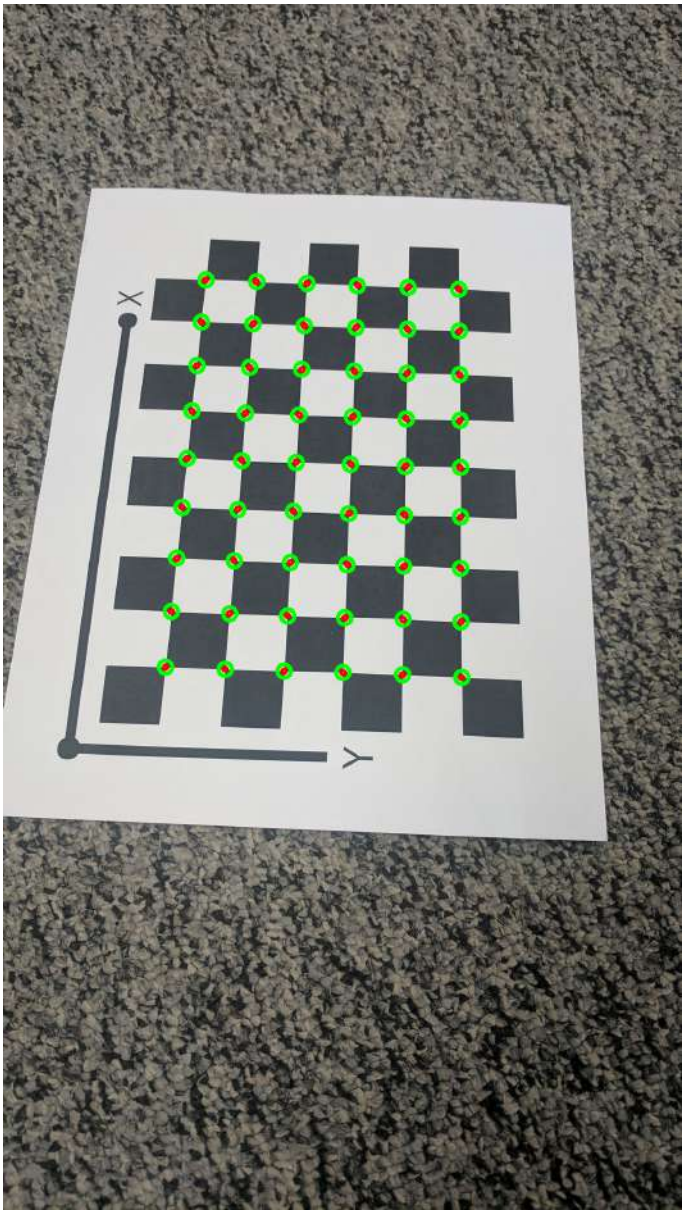

Fig. 4. Reprojection of points in image 1. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

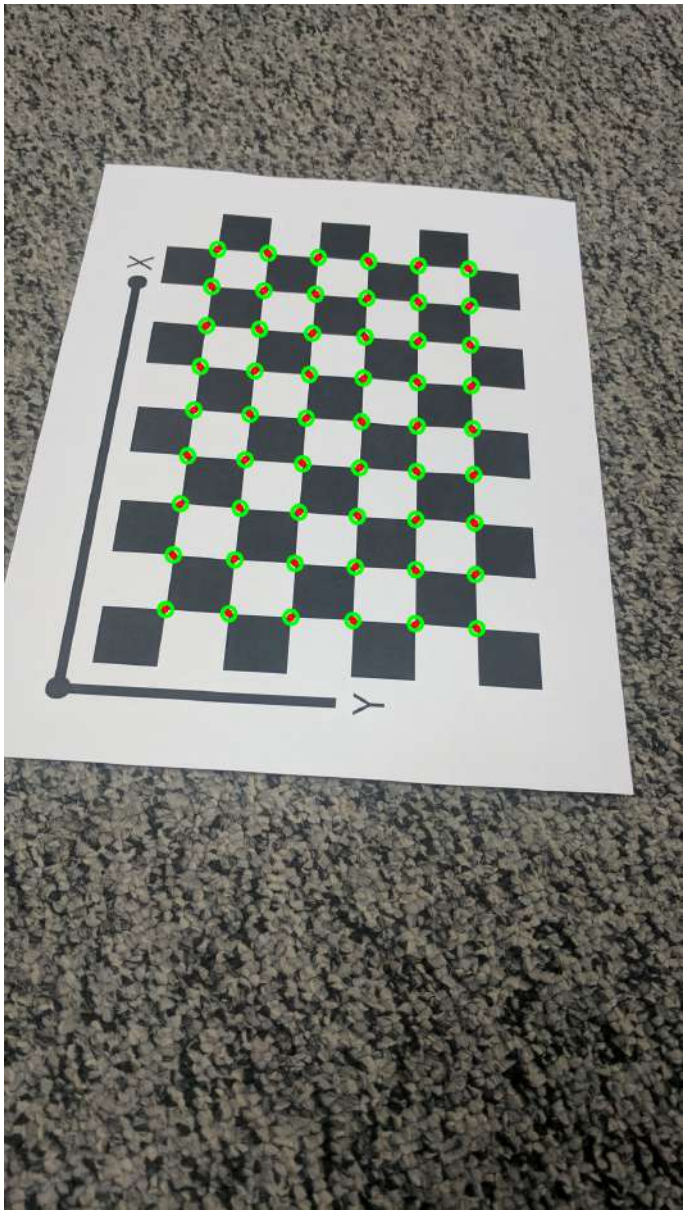

Fig. 5. Reprojection of points in image 2. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

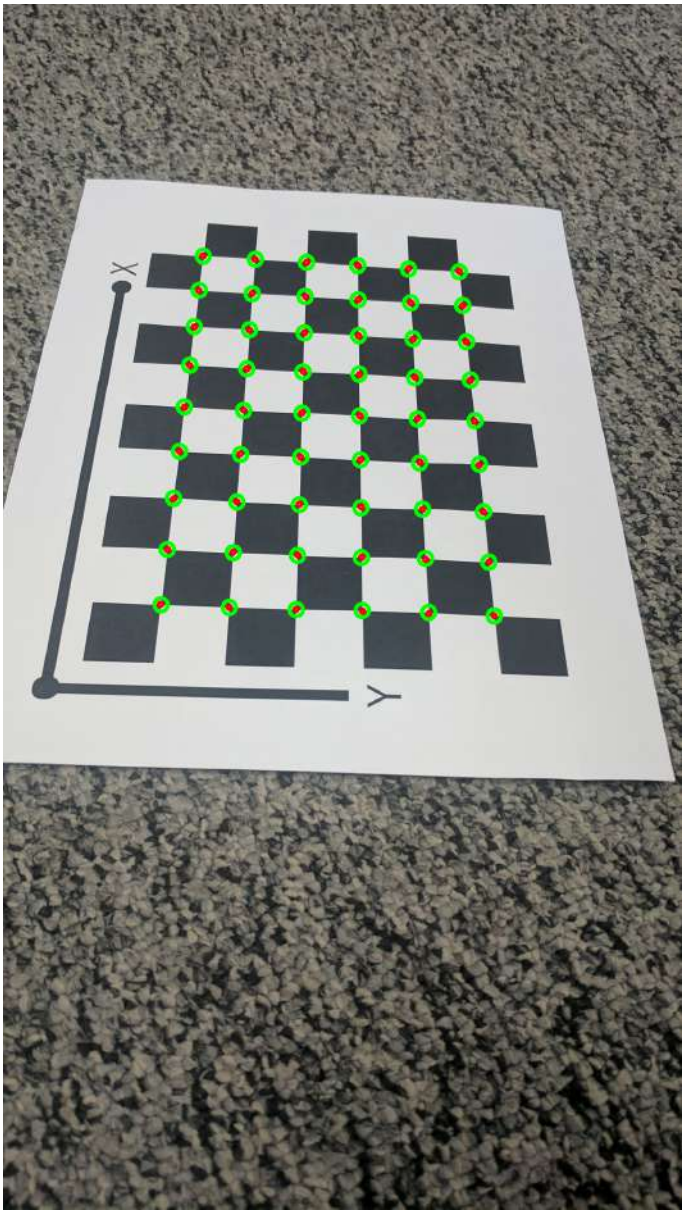

Fig. 6. Reprojection of points in image 3. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

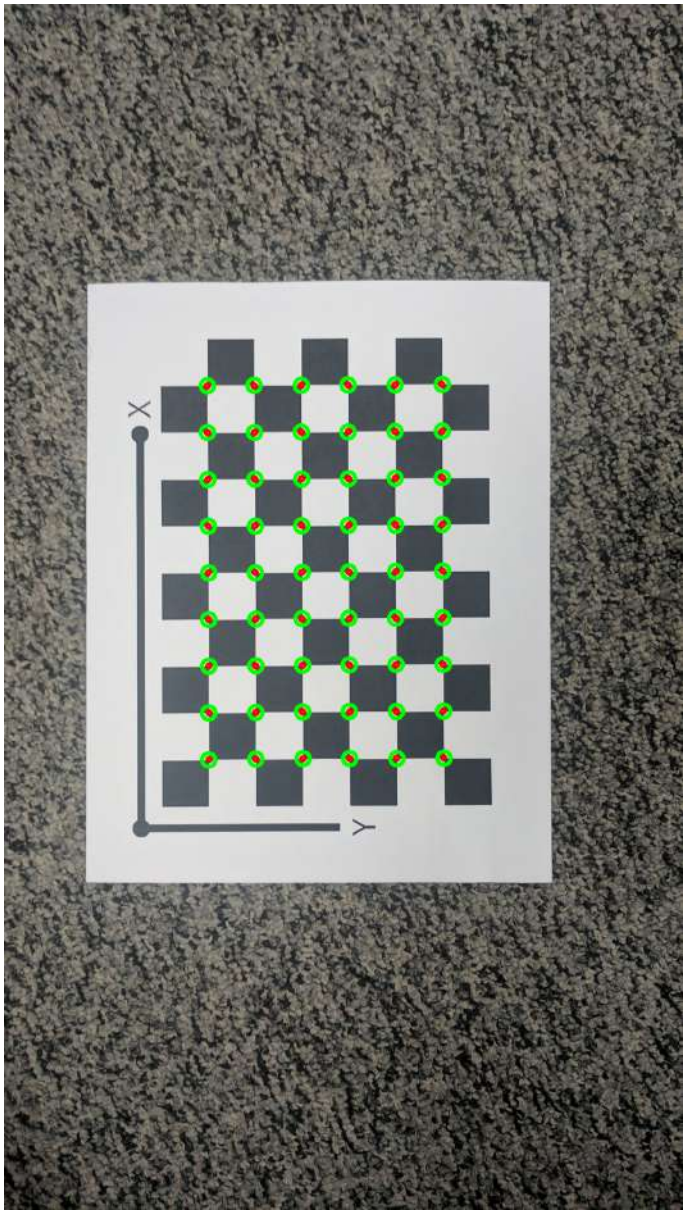

Fig. 7. Reprojection of points in image 4. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

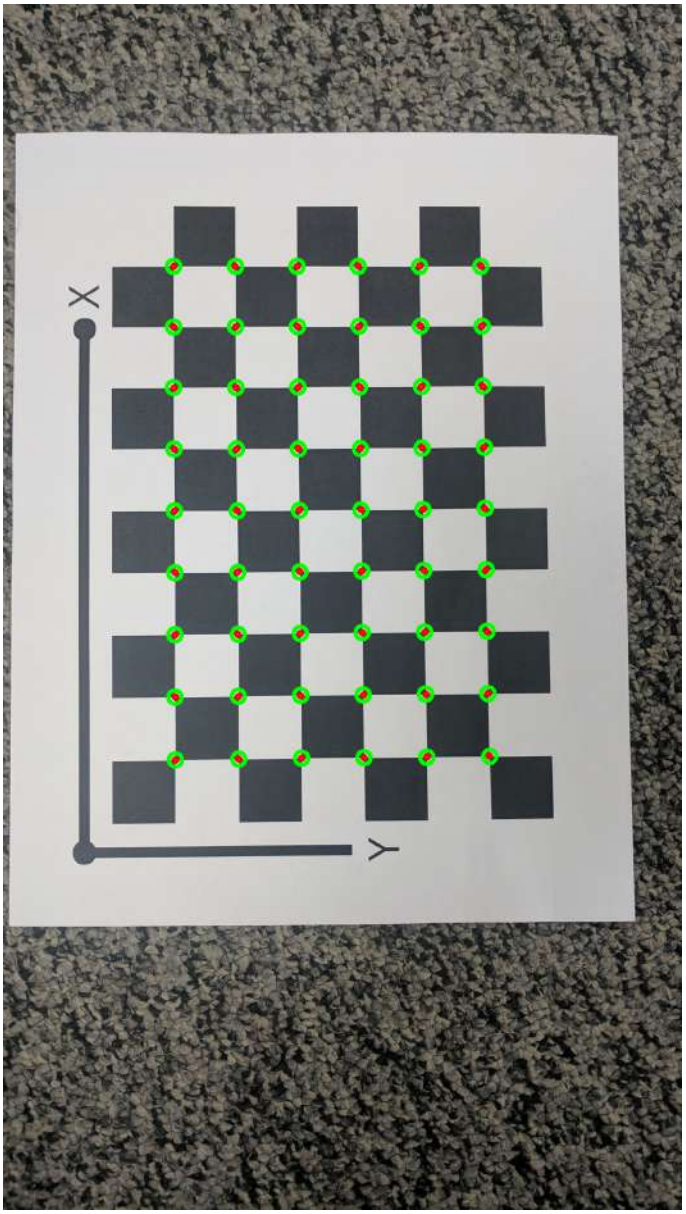

Fig. 8. Reprojection of points in image 5. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

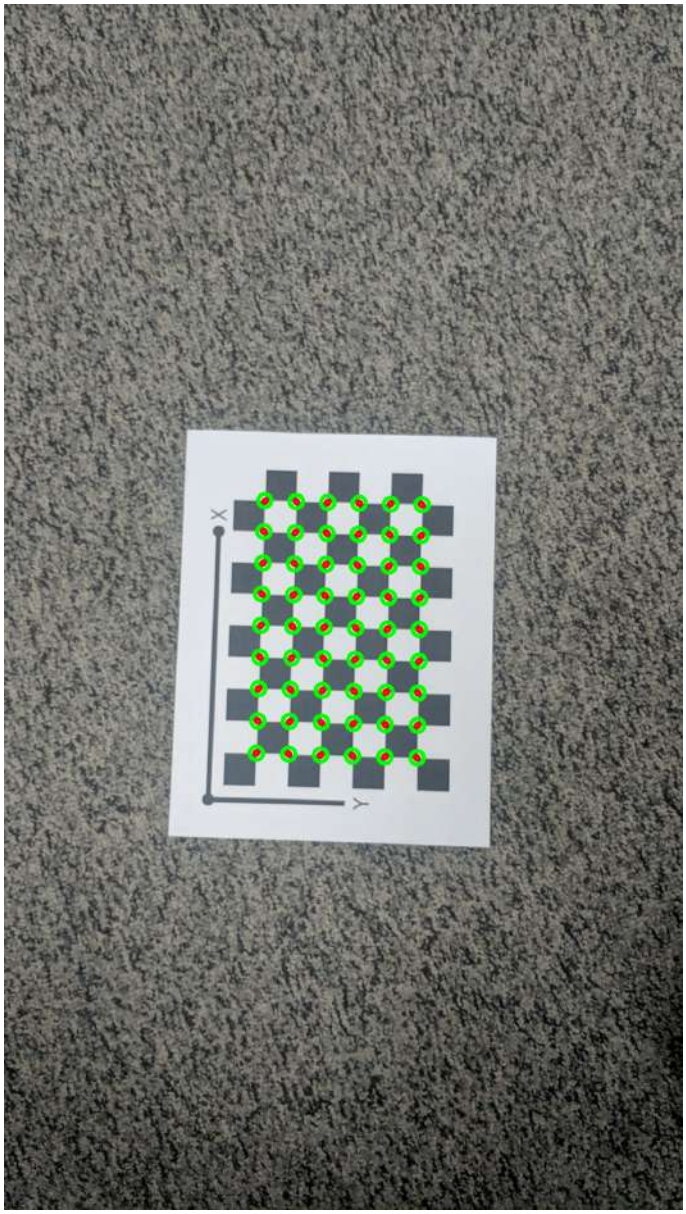

Fig. 9. Reprojection of points in image 6. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

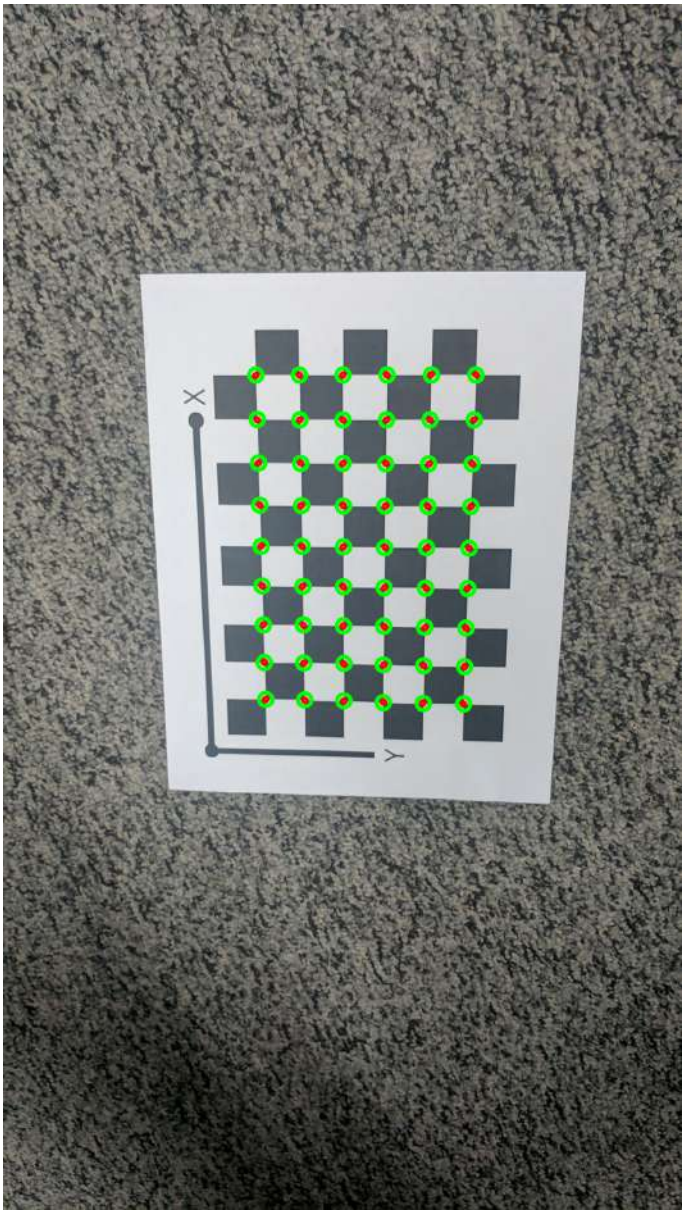

Fig. 10. Reprojection of points in image 7. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

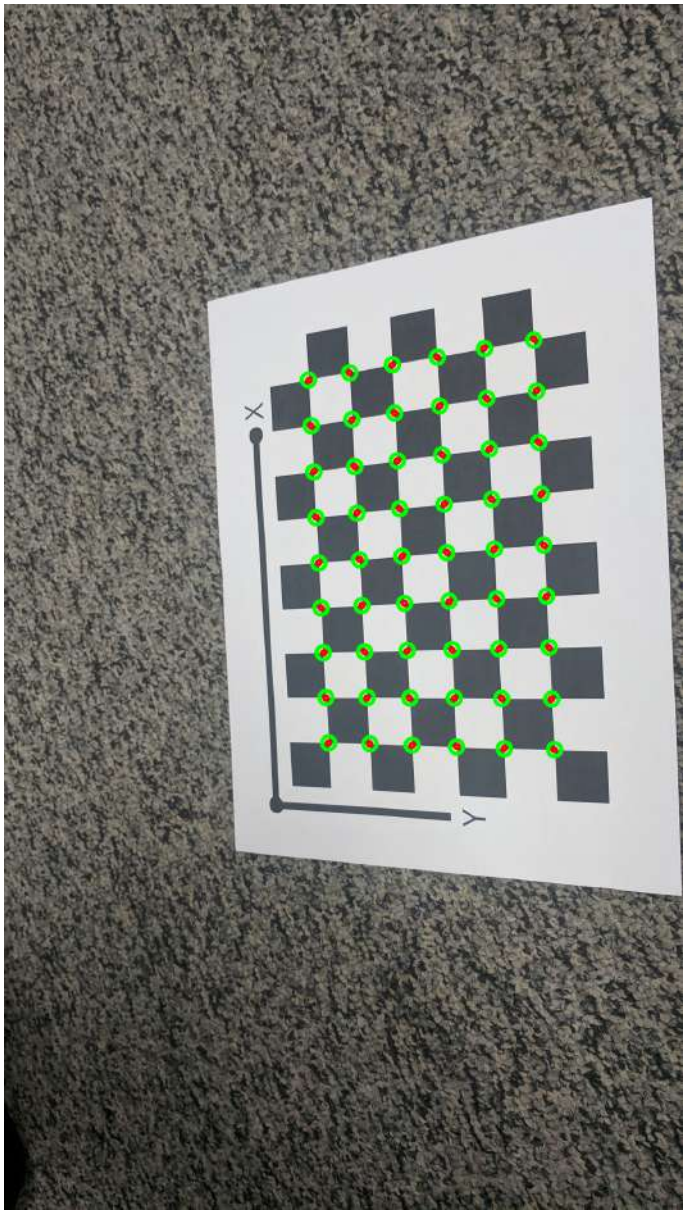

Fig. 11. Reprojection of points in image 8. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

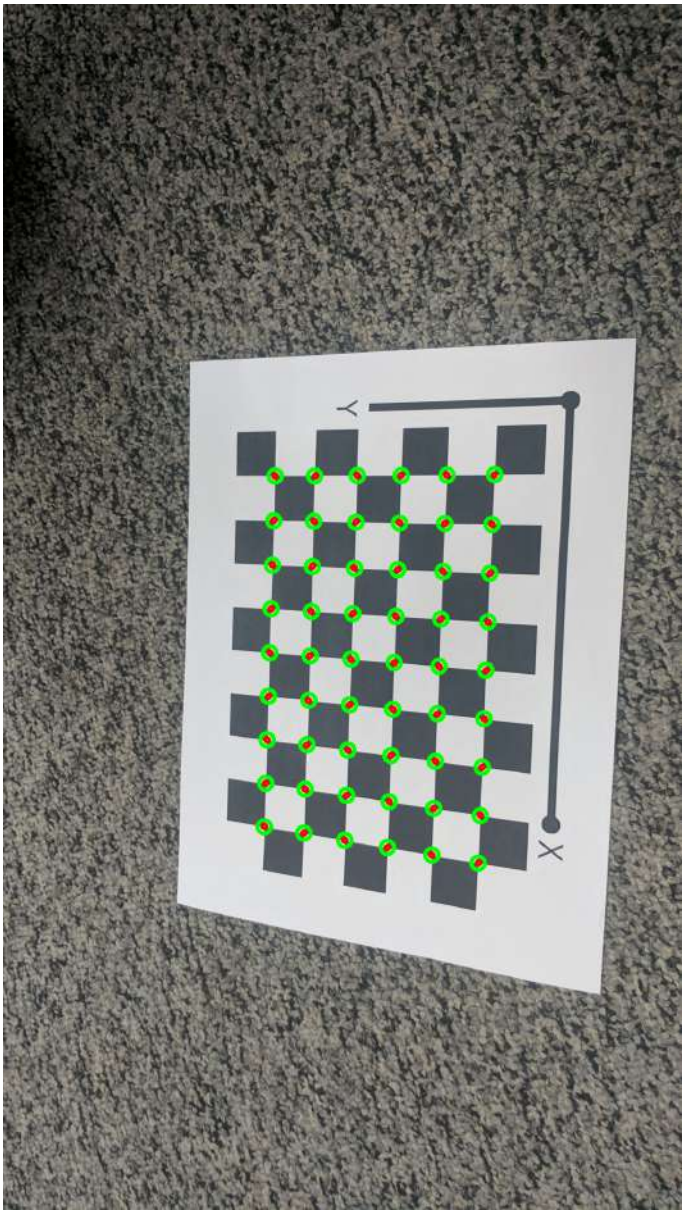

Fig. 12. Reprojection of points in image 9. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

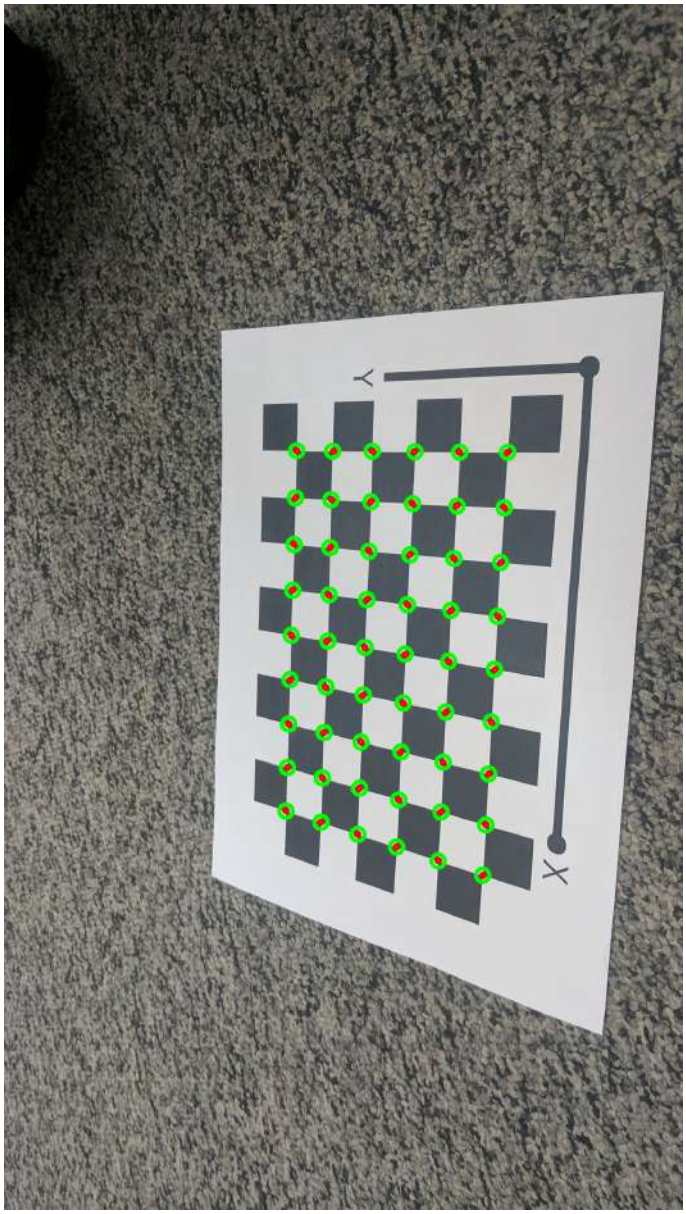

Fig. 13. Reprojection of points in image 10. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

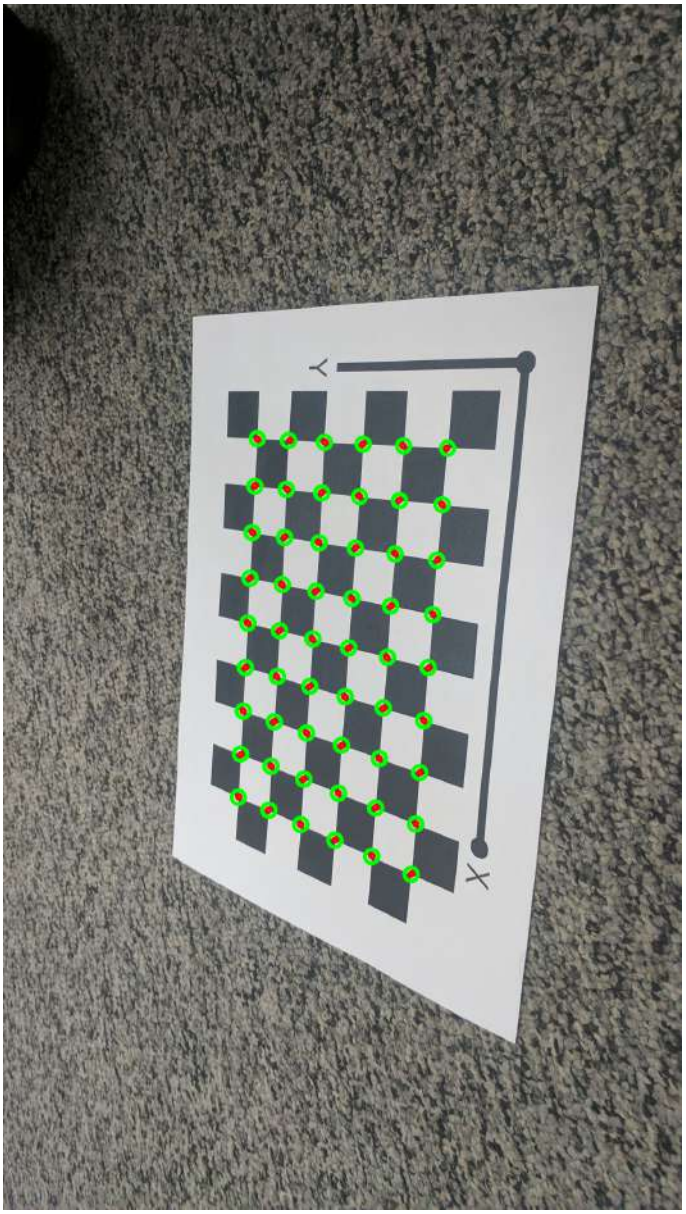

Fig. 14. Reprojection of points in image 11. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.

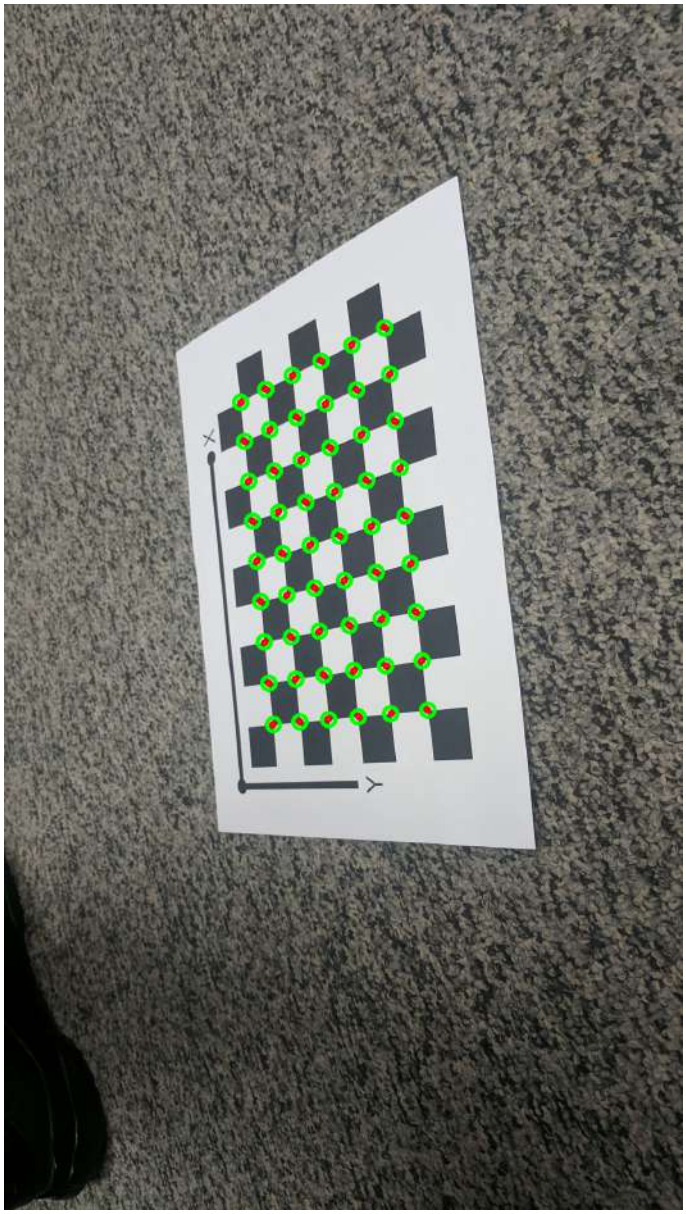

Fig. 15. Reprojection of points in image 12. Green Circle denotes the detected corner in the image. Red dot indicates the reprojected 3D point.