RBE549: HW1 - AutoCalib

Shreyas Devdatta Khobragade
MS in Robotics Engineering
Worcester Polytechnic Institute
skhobragade@wpi.edu

***Abstract*—Camera calibration is a critical step in 3D computer vision, enabling accurate geometric understanding of the scene. In this project, Zhengyou Zhang's robust and flexible camera calibration technique is implemented and tested on a dataset of checkerboard images captured at varying orientations. The method involves two stages: an initial closed-form estimation of intrinsic and extrinsic parameters using homographies, followed by non-linear optimization of intrinsic parameters (focal lengths, principal point) and distortion coefficients (radial distortion k_1, k_2) using the Levenberg-Marquardt algorithm. The calibration process yields accurate intrinsic parameters, extrinsic parameters (rotation R and translation t), and a low reprojection error, allowing for effective undistortion and rectification of images. This report details the step-by-step implementation and results of the calibration procedure.**

I. Introduction

Camera calibration is a basic computer vision procedure, especially for 3D geometry tasks where precise knowledge of the intrinsic and extrinsic parameters of the camera is crucial. While extrinsic parameters, like translation and rotation, define the camera's position and orientation in relation to the scene, intrinsic parameters, like principal point and focal lengths, describe the camera's internal characteristics. In order to model lens distortions that impact image quality, distortion coefficients are also utilized.

In this report, we implement the camera calibration technique proposed by Zhengyou Zhang in his seminal paper *"A Flexible New Technique for Camera Calibration."* Zhang's method is widely regarded for its robustness and flexibility, as it only requires a planar calibration pattern (e.g., a checkerboard) imaged from multiple orientations. The technique involves two key stages: an initial closed-form solution to estimate intrinsic and extrinsic parameters using homographies and a subsequent non-linear optimization step to refine these parameters alongside distortion coefficients.

This project's goal is to estimate the camera's extrinsic (rotation R and translation t), intrinsic (focal lengths f_x, f_y, principal point c_x, c_y), and radial distortion coefficients (k_1, k_2). A dataset of chessboard photos taken in various orientations is used to accomplish this. Reprojection error, which measures how well the estimated parameters project 3D points onto the 2D image plane, is used to assess the calibration results. Image rectification and undistortion are made possible by the final calibrated parameters, which are essential for computer vision applications.

II. Mathematical Background

A camera captures three-dimensional (3D) world points and maps them onto a two-dimensional (2D) image plane using a mathematical transformation. This process is modeled by the pinhole camera model, which establishes a direct relationship between 3D world coordinates and their corresponding 2D projections. In this model, a 2D image point is represented as $x = (x, y)^T$, and its homogeneous counterpart is expressed as $\bar{x} = (x, y, 1)^T$. Likewise, a 3D world point is denoted by $X = (X, Y, Z)^T$ and its homogeneous representation by $\bar{X} = (X, Y, Z, 1)^T$.

The transformation from 3D world coordinates to 2D image coordinates follows the equation:

$$s\bar{x} = K \begin{bmatrix} R & t \end{bmatrix} \bar{X}$$

where s is an arbitrary scale factor, K represents the intrinsic camera matrix, and R and t define the extrinsic parameters, which describe the rotation and translation of the camera with respect to the world coordinate frame. The intrinsic matrix K encodes internal camera properties such as focal length and the location of the principal point.

For practical applications such as camera calibration, it is often assumed that the reference plane of the model lies at $Z = 0$ in the world coordinate system. Under this assumption, the transformation simplifies to:

$$s\bar{x} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

where r_1 and r_2 correspond to the first two columns of the rotation matrix R. This formulation leads to the homography equation, which defines the relationship between the model plane and its projected image:

$$H = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

The homography can be estimated directly from an image of a known planar pattern by establishing correspondences between a set of 2D image points and their associated 3D model points. The fundamental objective of camera calibration is to compute the intrinsic matrix K using homographies obtained from multiple images of a planar pattern observed at different orientations.

III. Camera Intrinsics Estimation

To estimate the intrinsic camera matrix K, we begin by detecting checkerboard corners in a series of calibration images. The function `cv2.findChessboardCorners` from OpenCV is used to identify these corners, given a known pattern size. The Fig1 shows the output of corner detection. The world coordinates of the checkerboard are generated assuming a flat plane where $Z = 0$, using a predefined square size. This allows us to establish correspondences between the real-world (model) points and the detected image points. The homography matrix H is then computed using `cv2.findHomography`, which establishes a projective transformation between the model plane and the image plane. The homography captures how a set of 3D points maps to 2D points under perspective projection, providing a direct relationship between them.

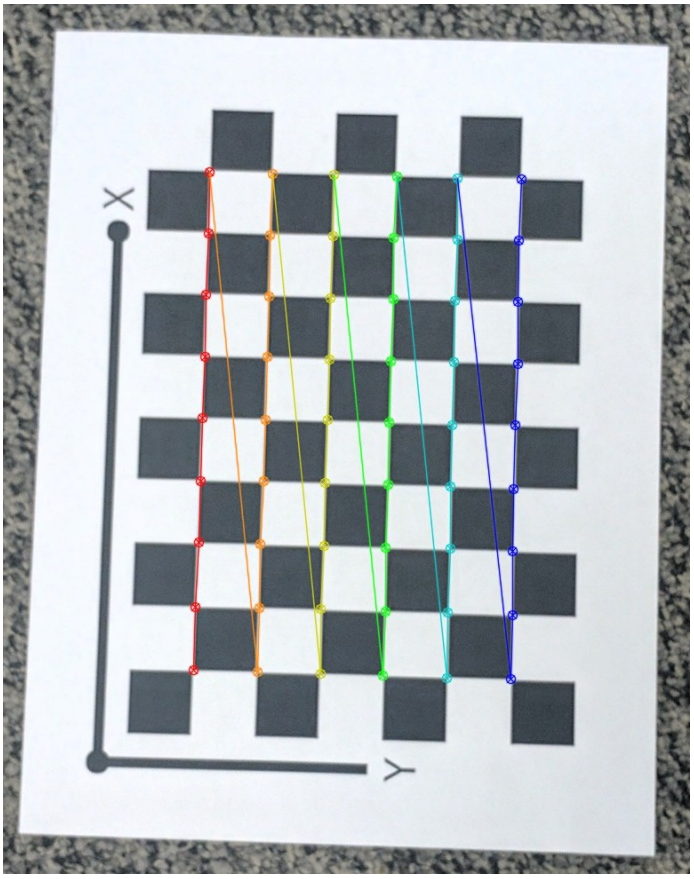

Fig. 1. Output of Corner Detection.

Once the homographies are obtained, we compute the intrinsic matrix K by solving a linear system derived from the orthogonality constraints of the rotation matrix. This involves forming a $\mathbf{V}$ matrix, constructed using the elements of the homographies. Using Zhang's method, constraints on intrinsic parameters are derived from each homography matrix $\mathbf{H}$. These constraints are represented as vectors v_{ij}:

$$v_{ij} = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix},$$

where h_{ij} are elements of $\mathbf{H}$. Two constraints are added for each homography and this can be rewritten in the form:

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

These constraints form a linear system represented by the matrix $\mathbf{V}$:

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

where $\mathbf{b}$ contains elements of the symmetric matrix $\mathbf{B}$ that encodes intrinsic parameters.

The equations used are:

$$h_i^T B h_j = v_{ij}^T b$$

where $B = K^{-\top}K^{-1}$ is a symmetric matrix represented by a six-element vector b.

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

We solve for b using Singular Value Decomposition (SVD), obtaining the intrinsic parameters as:

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2}$$

$$\lambda = B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}}$$

$$\alpha = \sqrt{\frac{\lambda}{B_{11}}}, \quad \beta = \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}}$$

$$\gamma = -\frac{B_{12}\alpha^2\beta}{\lambda}, \quad u_0 = \frac{\gamma v_0}{\beta} - \frac{B_{13}\alpha^2}{\lambda}$$

From these parameters, the intrinsic matrix is reconstructed as:

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

This approach provides an initial approximation of the camera's intrinsic parameters, which can later be refined using nonlinear optimization techniques such as Levenberg-Marquardt to minimize the overall reprojection error across multiple images.

IV. Camera Extrinsics Estimation

The extrinsic parameters of a camera, consisting of the rotation matrix R and translation vector t, define its position and orientation in the world coordinate system. Given a homography matrix H, these parameters are estimated using the intrinsic matrix K. The homography equation is:

$$H = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

where r_1 and r_2 are the first two columns of R, and t is the translation vector. To extract these components, we compute:

$$r_1 = \lambda K^{-1} h_1, \quad r_2 = \lambda K^{-1} h_2$$

where h_1 and h_2 are the first two columns of H, and the scale factor is:

$$\lambda = \frac{1}{\|K^{-1} h_1\|}$$

To ensure R is a valid rotation matrix, the third column is computed as $r_3 = r_1 \times r_2$. The translation vector is $t = \lambda K^{-1} h_3$, where h_3 is the third column of H. The final extrinsic matrix is:

$$\mathbf{E} = \begin{bmatrix} R & t \end{bmatrix}$$

This process is repeated for all homographies, providing an estimate of the camera's external pose for each image.

V. Non-Linear Geometric Error Minimization

After obtaining the initial estimates of the intrinsic matrix K, the extrinsic parameters R and t, and the distortion coefficients k, we refine these parameters by minimizing the geometric reprojection error. This error is defined as the difference between the observed image points $x_{i,j}$ and the projected points $\hat{x}_{i,j}$, computed using the current estimates of K, R, t, and k. The objective function for optimization is given by:

$$\underset{f_x, f_y, c_x, c_y, k_1, k_2}{\arg\min} \sum_{i=1}^{N} \sum_{j=1}^{M} \|x_{i,j} - \hat{x}_{i,j}(K, R_i, t_i, X_j, k)\|$$

where $x_{i,j}$ and $\hat{x}_{i,j}$ are the inhomogeneous image coordinates of the observed and projected points, respectively. The function `scipy.optimize.least_squares` is used to iteratively minimize this error using the Levenberg-Marquardt algorithm.

The distortion model applies radial distortion, modifying the ideal projection using:

$$x_{\text{distorted}} = x_{\text{projected}} \cdot \left(1 + k_1 r^2 + k_2 r^4\right)$$

where

$$r^2 = x_{\text{projected}}^2 + y_{\text{projected}}^2$$

is the squared radial distance. The optimization process refines the camera parameters by adjusting K, R, t, and k to minimize reprojection errors. Once optimized, the updated intrinsic matrix and distortion coefficients are obtained, ensuring a more accurate camera calibration for improved real-world measurements.

VI. Results

The described method was tested on a dataset consisting of 13 images. This section presents the resulting camera intrinsic parameters along with the associated errors.

The camera calibration matrix K before and after optimization is shown below:

$$K_{\text{before}} = \begin{bmatrix} 2053.33645 & -0.46737 & 762.771046 \\ 0 & 2037.21492 & 1352.42128 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_{\text{after}} = \begin{bmatrix} 2045.83874 & -1.64026 & 759.425669 \\ 0 & 2037.53965 & 1345.26896 \\ 0 & 0 & 1 \end{bmatrix}$$

The values of the distortion coefficients after optimization are as follows:

$$k_1 = 0.17194771293647218, \quad k_2 = -0.736270606471421$$

The reprojection errors before and after optimization for each image are listed in Table I. The mean reprojection error over all images is also provided.

TABLE I
Reprojection Errors

Image	Error (Initial)	Error (Optimized)
1	1.346	0.337
2	1.916	0.407
3	2.440	0.503
4	3.492	0.562
5	0.824	0.272
6	1.318	0.378
7	0.853	0.499
8	1.213	0.304
9	1.086	0.330
10	1.064	0.344
11	1.723	0.431
12	1.610	0.487
13	2.960	0.483
Mean Error	**1.680**	**0.411**

After correcting the errors caused by radial distortion, the image is undistorted using the `cv.undistort()` function. Subsequently, the 3D points are re-projected onto the calibration images. This process produces an undistorted image with accurately projected corner points. The results can be seen in images provided in Appendix section.

References

[1] Zhengyou Zhang, *A Flexible New Technique for Camera Calibration*, Technical Report MSR-TR-98-71, Microsoft Research, Microsoft Corporation, Redmond, WA, December 2, 1998 (updated August 13, 2008). Available at: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf.

APPENDIX

This section contains the images after rectification with points reprojected corners and originally detected corners. The red dots represent the reprojected corners and the green dots represent originally detected corners.

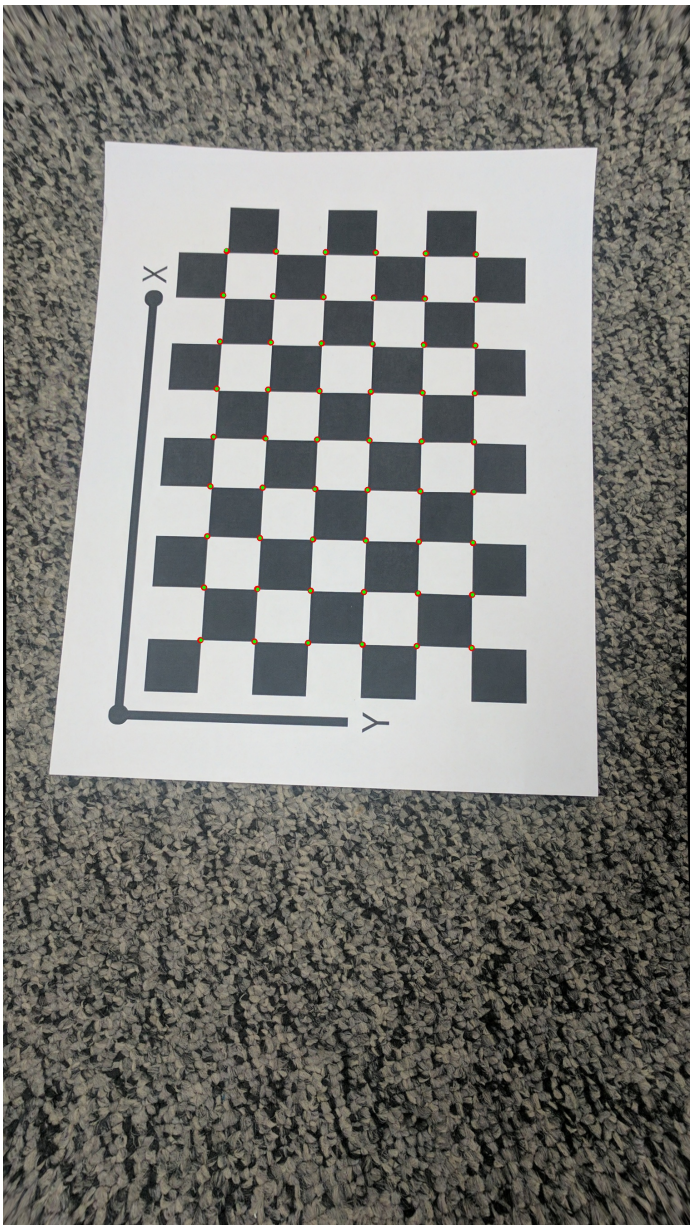

Fig. 2. Output of Image 1 after Rectification.

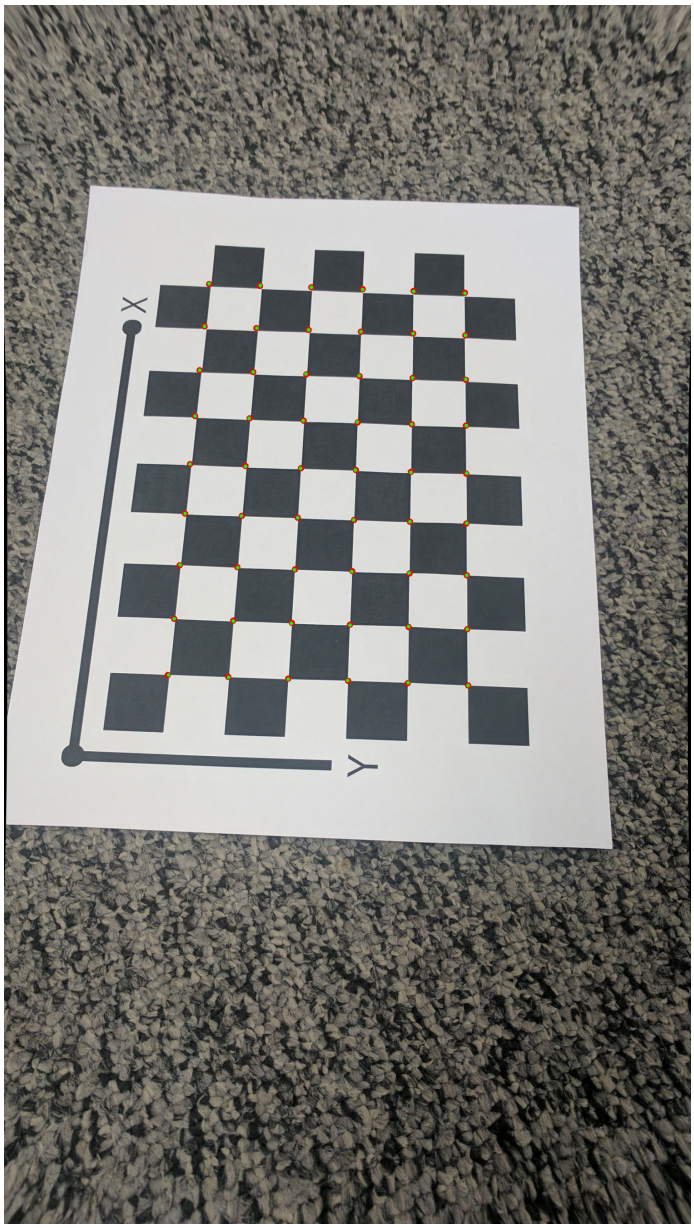

Fig. 3. Output of Image 2 after Rectification.

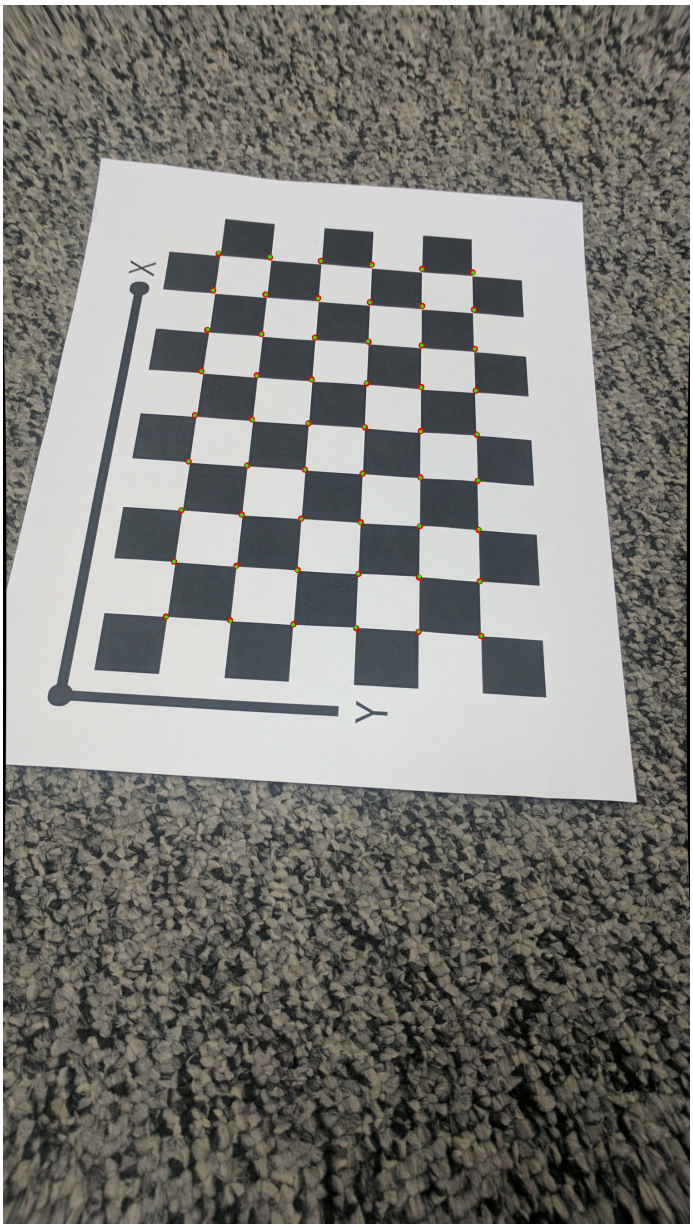

Fig. 4. Output of Image 3 after Rectification.

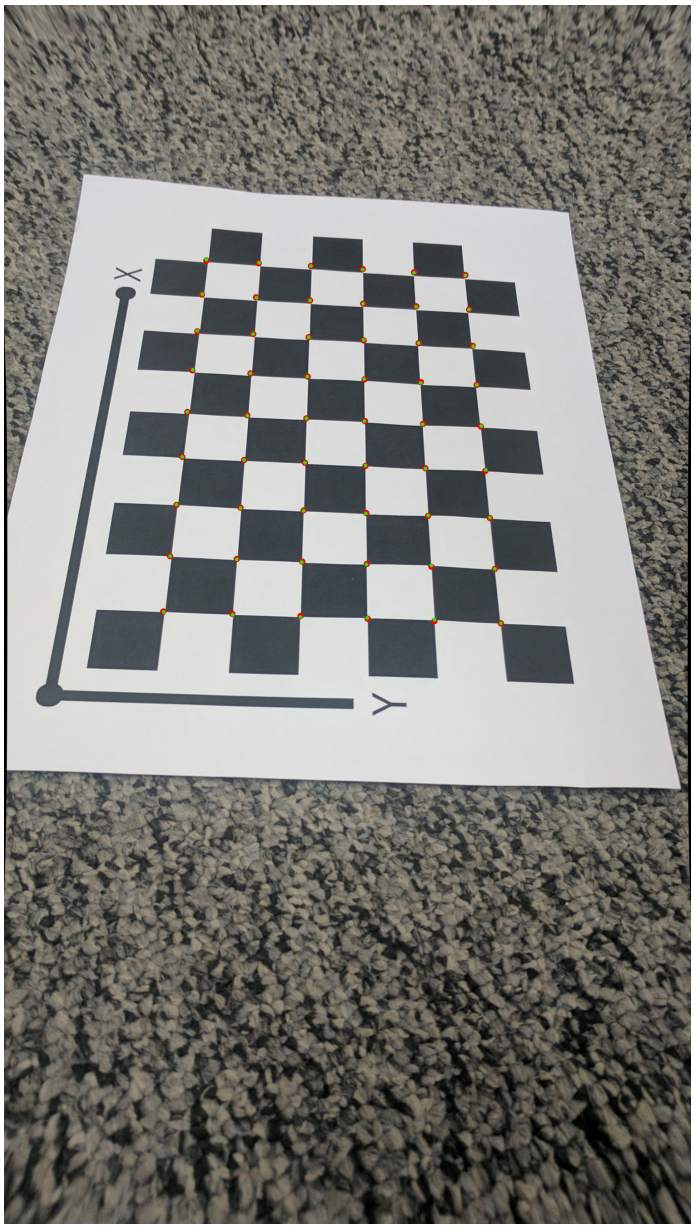

Fig. 5. Output of Image 4 after Rectification.

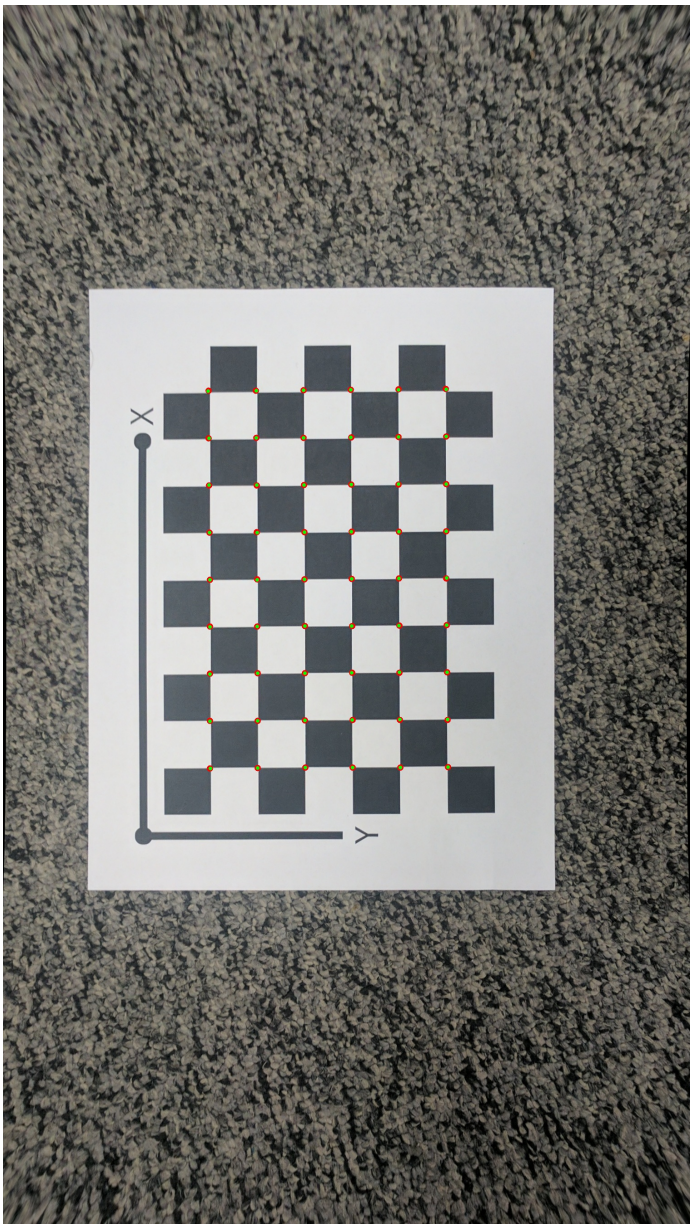

Fig. 6. Output of Image 5 after Rectification.

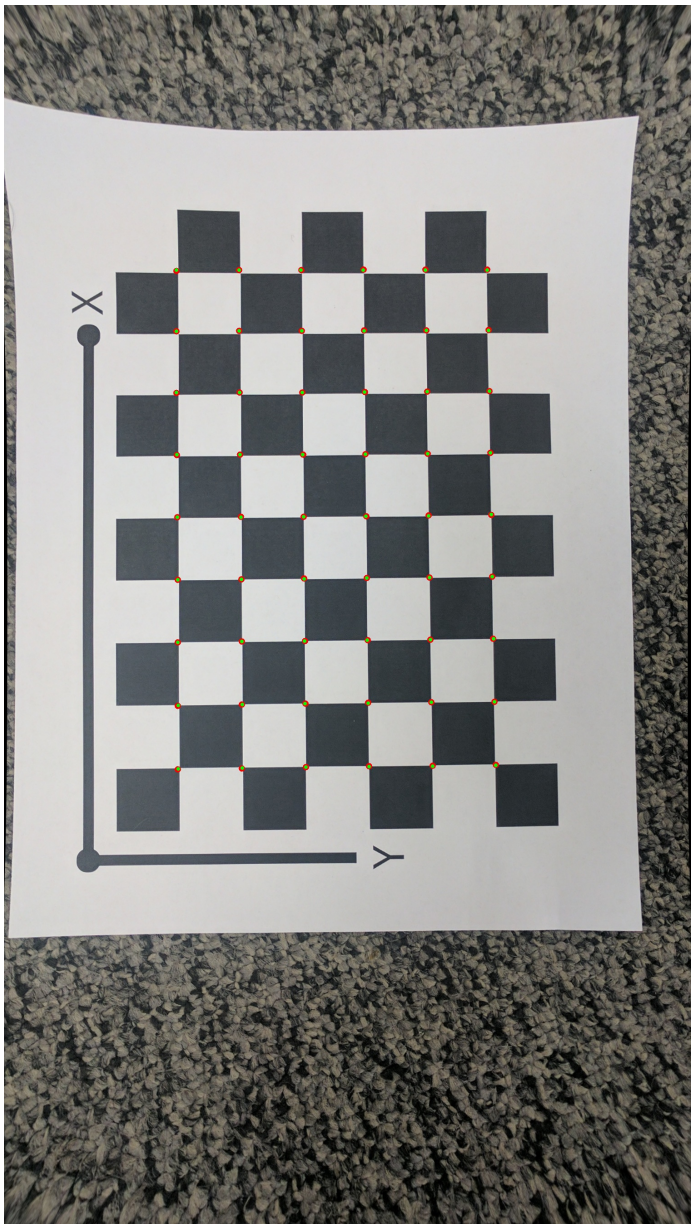

Fig. 7. Output of Image 6 after Rectification.

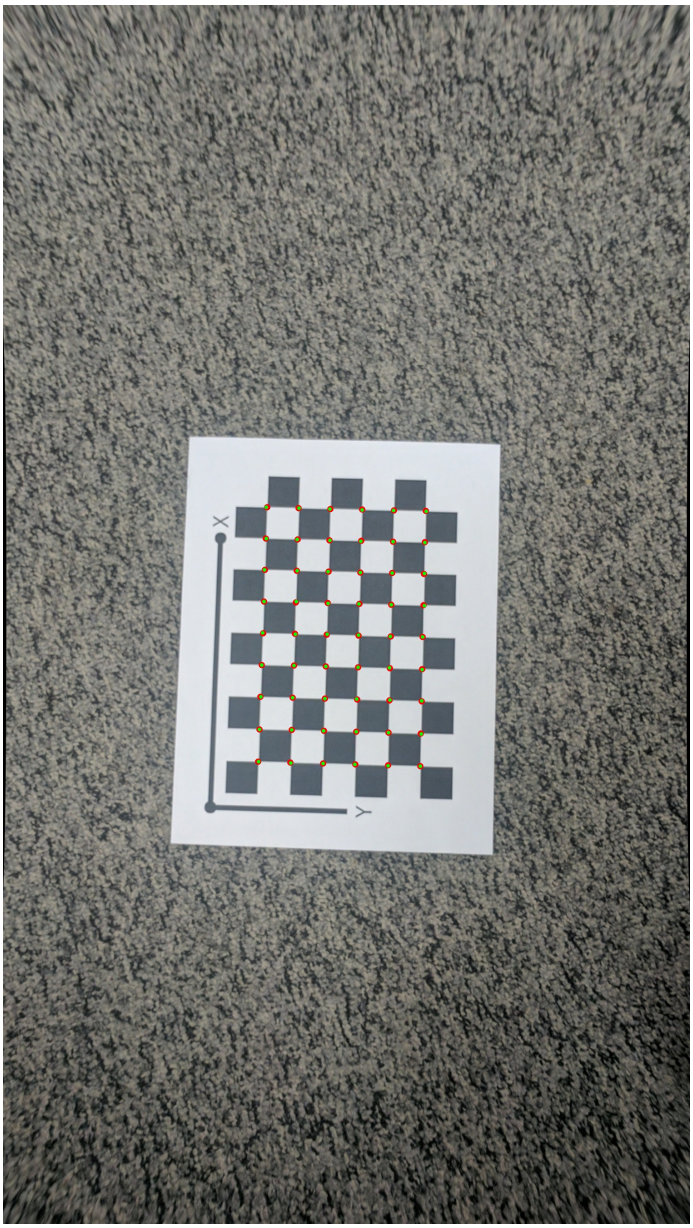

Fig. 8. Output of Image 7 after Rectification.

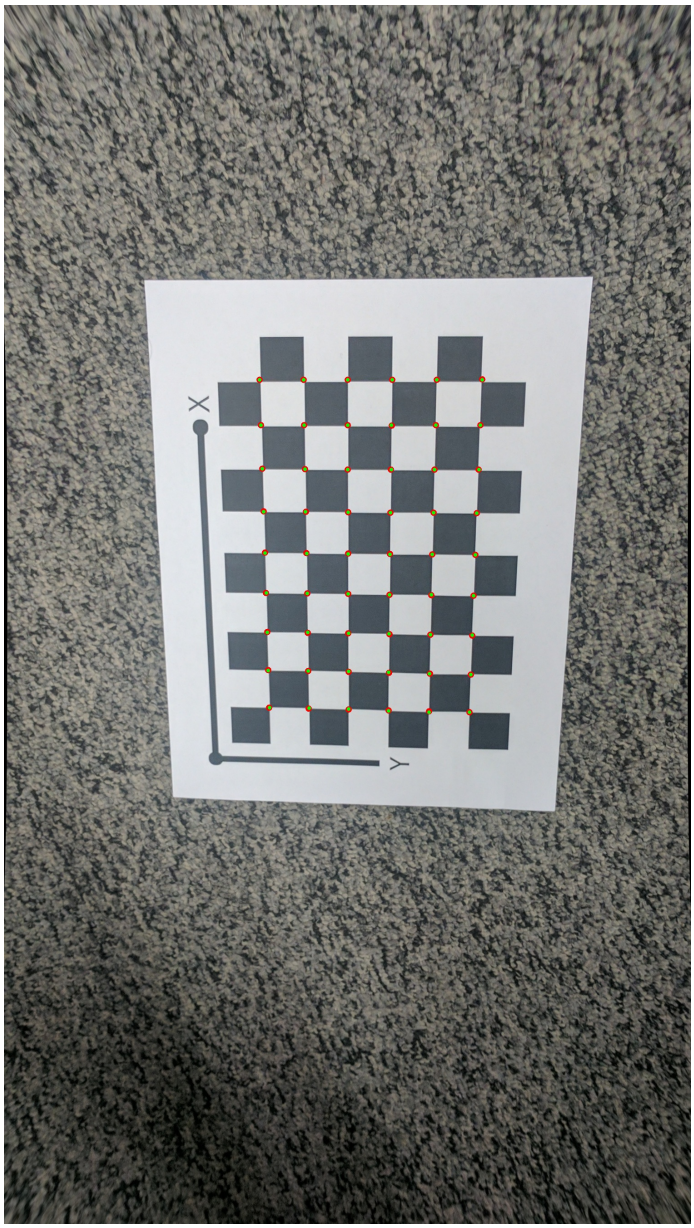

Fig. 9. Output of Image 8 after Rectification.

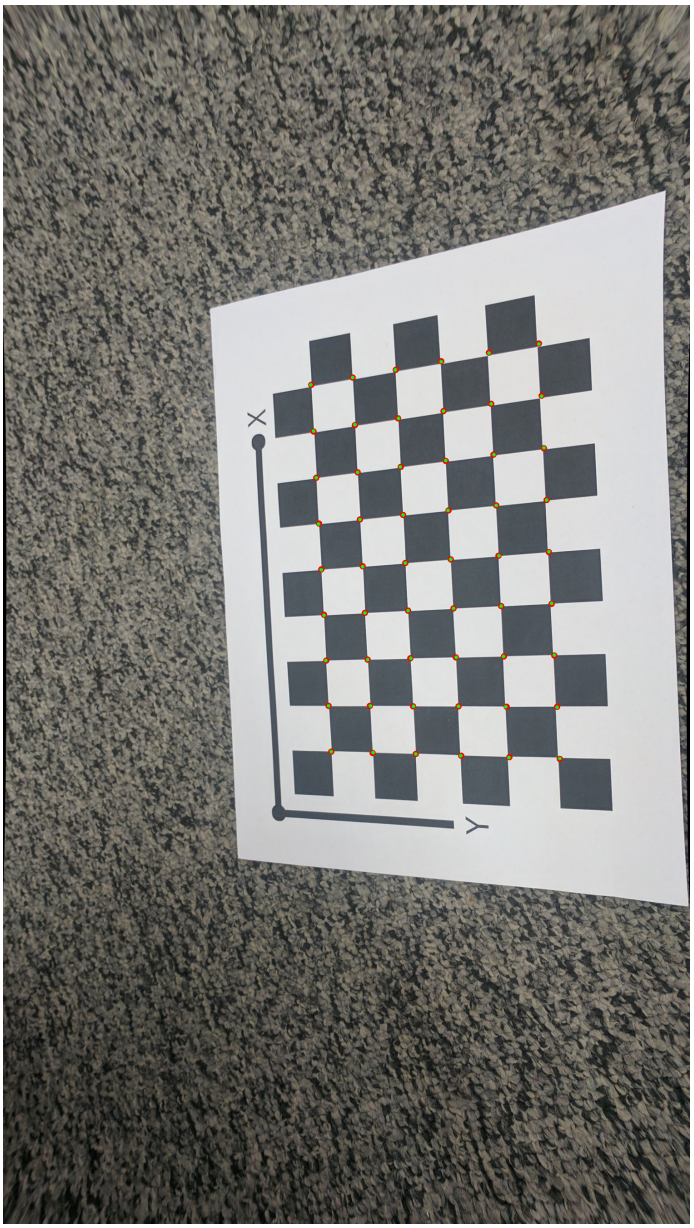

Fig. 10. Output of Image 9 after Rectification.

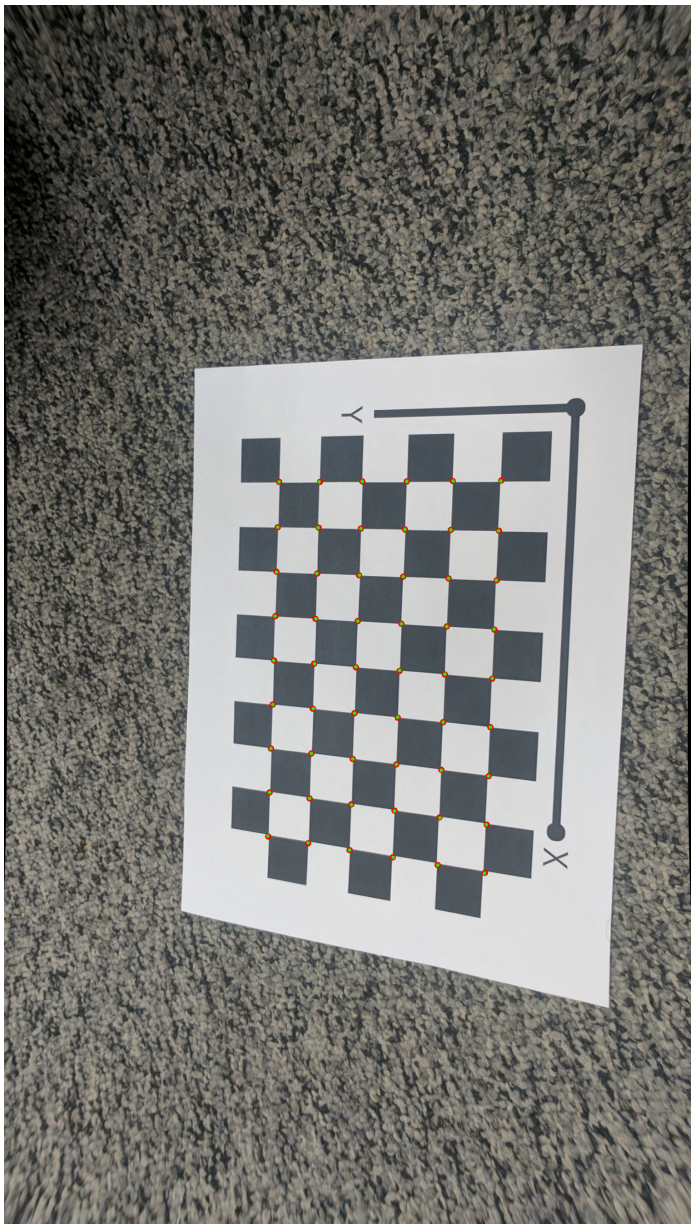

Fig. 11. Output of Image 10 after Rectification.

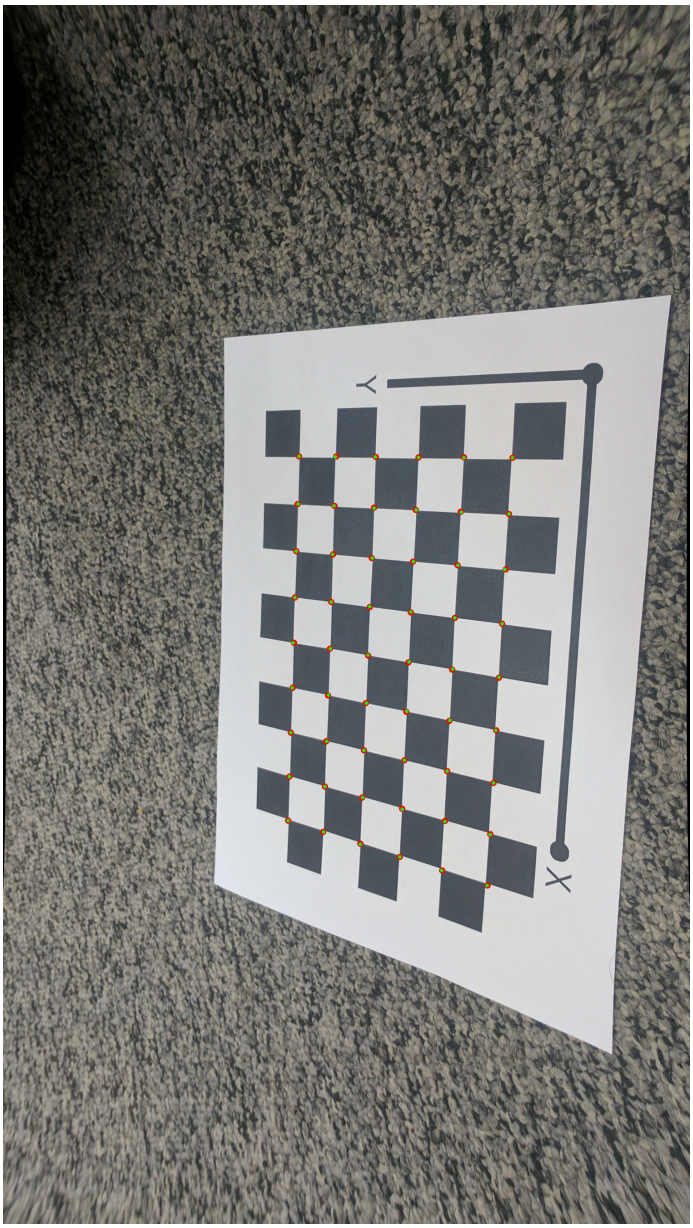

Fig. 12. Output of Image 11 after Rectification.

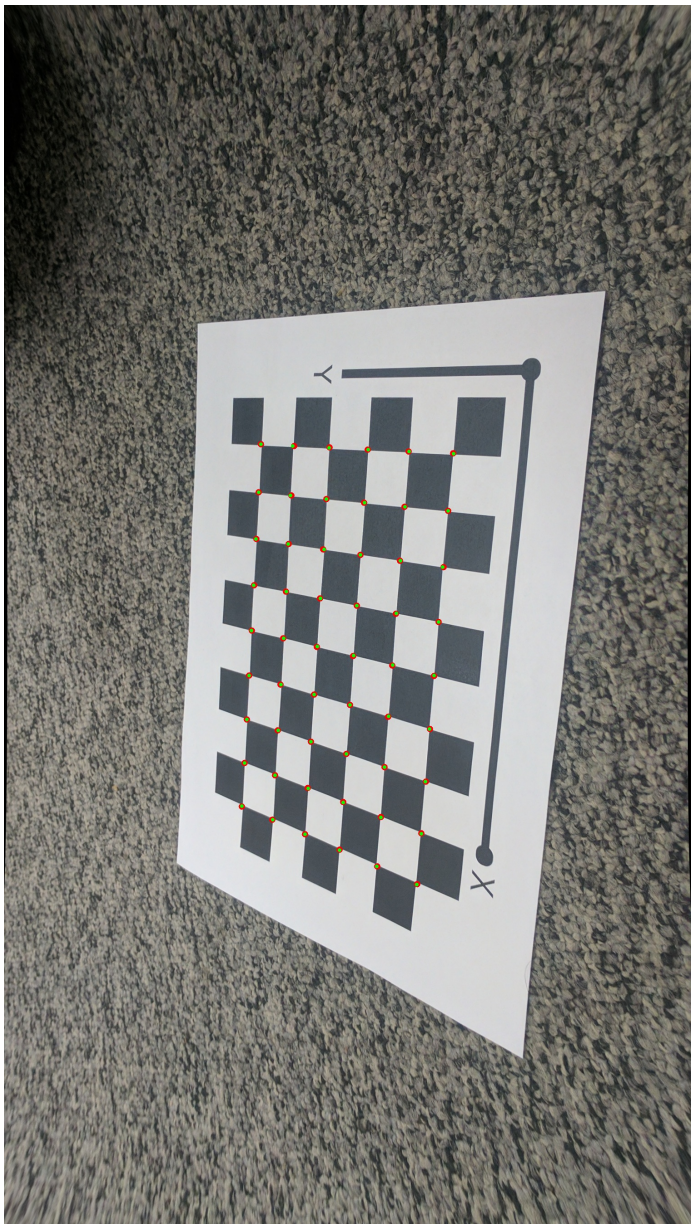

Fig. 13. Output of Image 12 after Rectification.

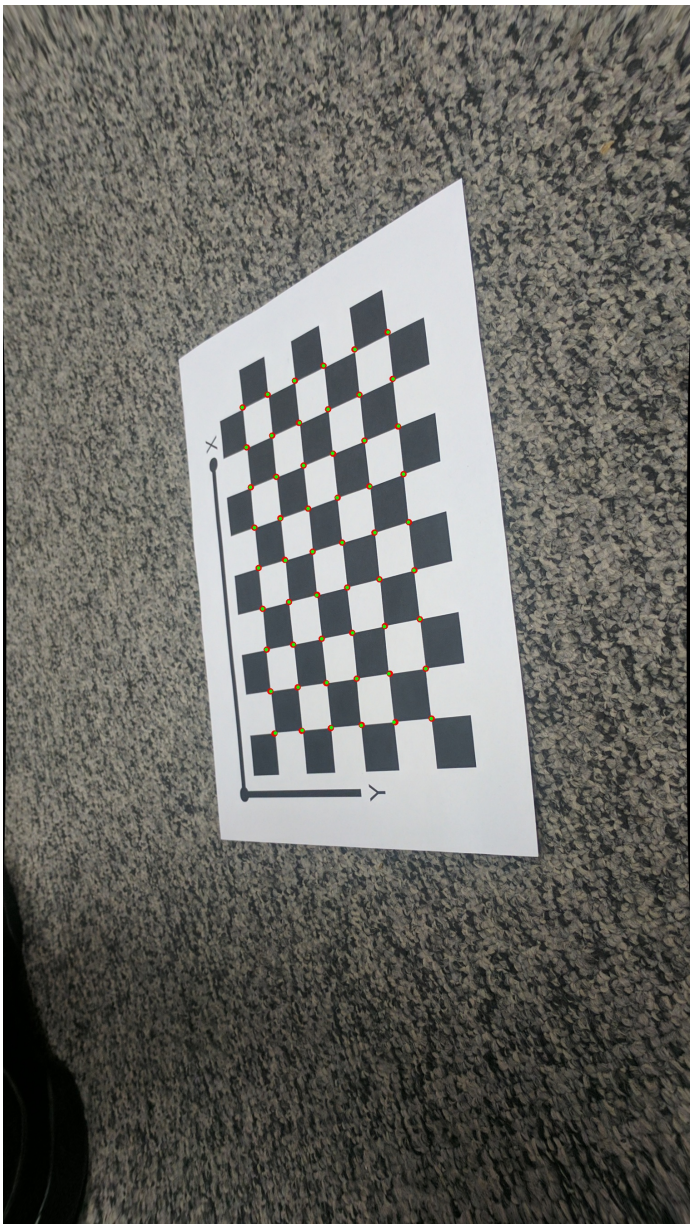

Fig. 14. Output of Image 13 after Rectification.