RBE549 : Homework 1 - AutoCalib

Mihir Deshmukh
Worcester Polytechnic Institute
Worcester, MA
Email: mpdeshmukh@wpi.edu

***Abstract*—This homework involves calculating the intrinsic, extrinsic, and radial distortion parameters for a given camera using the camera calibration method presented by Zhengyou Zhang.**

I. Introduction

The process of estimating the parameters of the camera, like the focal length, principal point, and distortion coefficients, is called camera calibration. Photogrammetric calibration is performed by observing a calibration object whose 3D structure is known. The intrinsic camera matrix is shown in the equation below:

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

Here, (α, β) are the scale factors in image u and v axes, (u_0, v_0) are coordinates of the principal points, and γ describes the skewness between the two images axes.

Zhang's method assumes the chessboard plane to be at Z=0. Hence, the projection matrix becomes as shown in Fig.1. This is used to convert a world coordinate point to pixel coordinates. A is the intrinsic matrix, and the following matrix is the extrinsic parameter or the location and orientation of the camera with respect to the world frame. And the last is the chessboard corner point in the world frame. Note Z=0 as mentioned in the method.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Fig. 1. Projection matrix equation

II. Initial Parameter Estimation

To calculate the intrinsic parameters, we first try and get an initial estimate, which we will optimize further using nonlinear optimization. To get this estimate, it involves the following steps:

A. ChessBoard Corner Detection

First, using the 13 provided calibration images containing the images of a known size (21.5 mm) chessBoard, we use the *cv2.findChessboardCorners* to find these corners. Fig. 2 shows this output for Image 1. We have the list of corners for every image for which we know the world frame coordinates, which is the inner 9 * 6 grid on the chessboard (we are ignoring the outer corners).

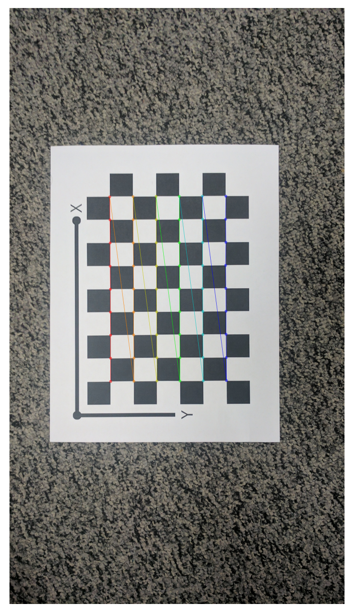

Fig. 2. Detected chessBoard corners

B. Estimating Intrinsic Parameters

Now that we have the corners in pixel coordinates and the world coordinates, we generate the homography H between the image corners and the world corners. Once, we have the homography H between the world points and the pixel coordinates, we can find A. Assume that:

$$\begin{aligned} B &= A^{-T}A^{-1} \\ &= \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \end{aligned}$$

The Matrix B can also be represented as a $6D$ Vector.

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

Let the i^{ith} column vector of H be $h_i = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix}^T$ Then we have.

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

where

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

This way, the constraints for a given homography can be rewritten as the following 2 homogenous equations in b. Where v is defined using all the H matrices obtained from each of the 13 calibration images.

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 01$$

Given that we have 13 images, we can calculate values for all the parameters. Once b is known, we can compute the camera intrinsic parameters matrix A using the following equations.

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - [B_{13}^2 + v0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\ \alpha &= \sqrt{\lambda/B_{11}} \\ \beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\ \gamma &= -B_{12}\alpha^2\beta/\lambda \\ u_0 &= \gamma v_0/\beta - B_{13}\alpha^2/\lambda \end{aligned}$$

By plugging the values in 1, we can compute initial estimate of the camera intrinsic matrix A.

$$A = \begin{bmatrix} 2052.789 & -0.3697 & 763.06 \\ 0.00 & 2036.635 & 1352.614 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

C. Estimating Extrinsic Parameters

We use the intrinsic parameters as well as the homography H obtained in the previous step to compute the extrinsic parameters. The extrinsic parameters give the position and orientation of the camera with respect to the world frame. Hence, we get 13 individual matrices for each of the images. They are calculated using the following equations:

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

Extrinsic Parameters Rt for Image 0:

$$\begin{bmatrix} 2.610e-02 & 9.945e-01 & 1.825e-02 & -6.036e+01 \\ -9.958e-01 & 2.546e-02 & 8.733e-02 & 9.552e+01 \\ 8.727e-02 & -2.056e-02 & 9.910e-01 & 6.164e+02 \end{bmatrix} \quad (2)$$

D. Radial Distortion

The initial estimate of the radial distortion is taken to be [0, 0], this will be found out using the next non-linear optimization step. The following equation is used to account fro distortion while calculating the reprojection error for optimization.

$$\begin{aligned} \breve{u} &= u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \breve{v} &= v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \,. \end{aligned}$$

Fig. 3. Equation to account for distortion

III. Non-Linear Geometric Error Minimization

Now, the intrinsic parameters as well the distorting coefficients are passed the scipy.optimize to reduce the re-projection error(the error between the pixel values gotten using Fig.**??** and the initial intrinsic & extrinsic parameters). We equation for the same is shown in Fig.4.

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

Fig. 4. Optimization equation

After optimization we obtain the below parameters.
New Intrinsic Parameters A:

$$\begin{bmatrix} 2052.782 & -0.36886 & 763.06 \\ 0.0 & 2036.622 & 1352.631 \\ 0.0 & 0.00 & 1.000 \end{bmatrix} \quad (3)$$

Optimized distortion parameters (k1,k2):
$[0.0140004, -0.101975]$

IV. Re-projection Results

After the non-linear minimization, we calculate the reprojection error. Fig.5 shows the reprojection error for each image. The mean reprojection error for the 13 images is:

$$Reprojectionerror : 0.7502158719728964 \quad (4)$$

The following figures show the reprojected corners on the rectified images Fig7-Fig14. All the other results can be seen in the files attached.

```
Reprojection Error for Image 0: 0.84080951704672
Reprojection Error for Image 1: 0.6719514456894702
Reprojection Error for Image 2: 0.64220369533045
Reprojection Error for Image 3: 0.8560799125231574
Reprojection Error for Image 4: 0.6447034716450264
Reprojection Error for Image 5: 0.7100754071752333
Reprojection Error for Image 6: 0.8461449124487898
Reprojection Error for Image 7: 0.9774302991796047
Reprojection Error for Image 8: 0.5874614865225203
Reprojection Error for Image 9: 0.9740221219133679
Reprojection Error for Image 10: 0.7237075497571075
Reprojection Error for Image 11: 0.761537209638511
Reprojection Error for Image 12: 0.5166793067776941
```

Fig. 5. Reprojection error for each image.

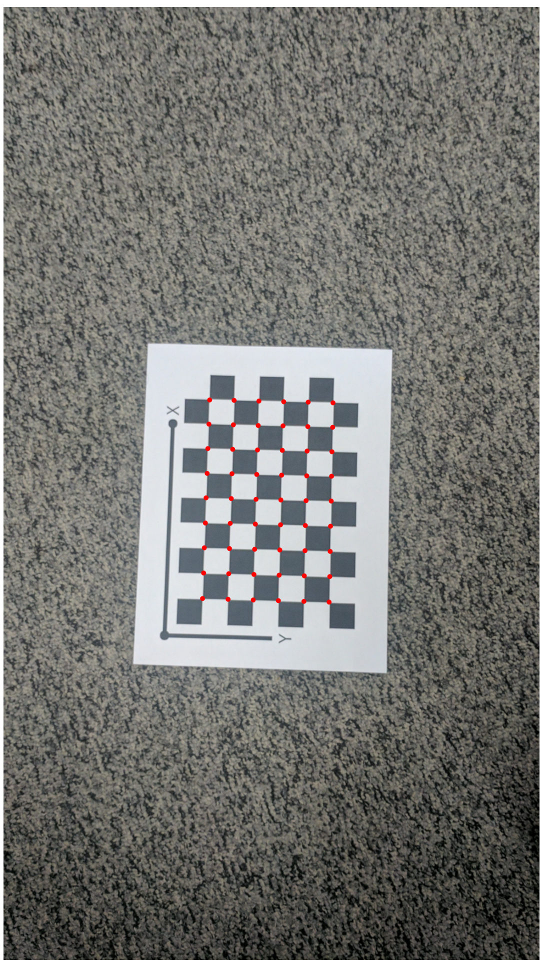

Fig. 6. Reprojected Image 0

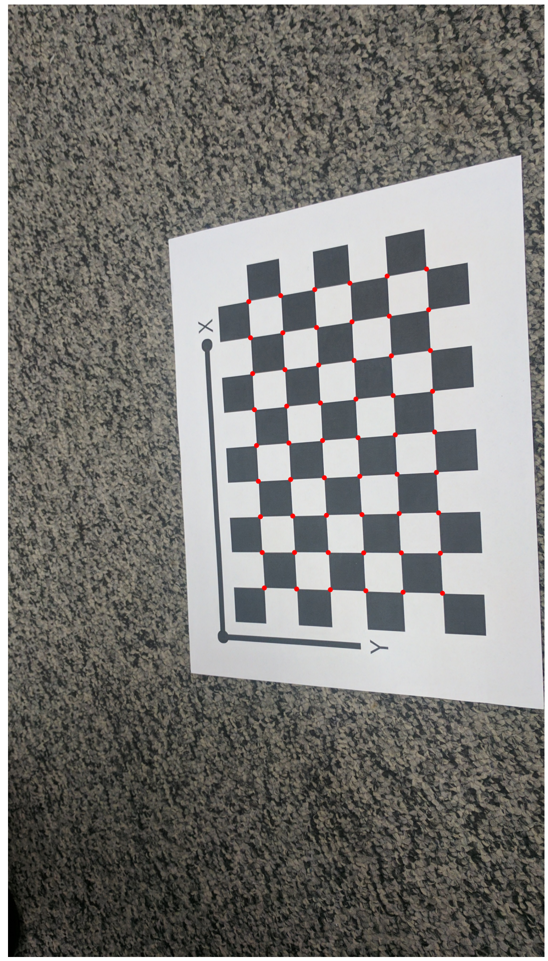

Fig. 7. Reprojected Image 1

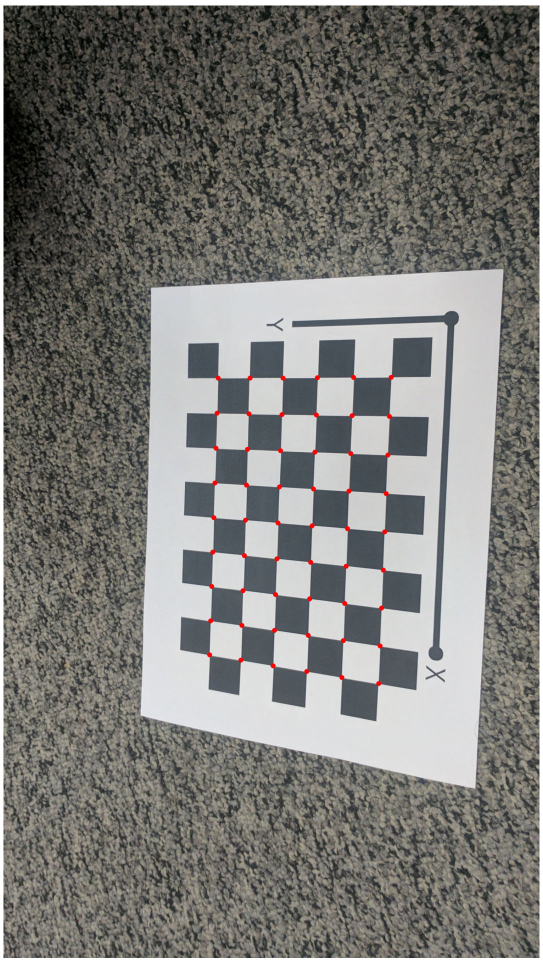

Fig. 8. Reprojected Image 2

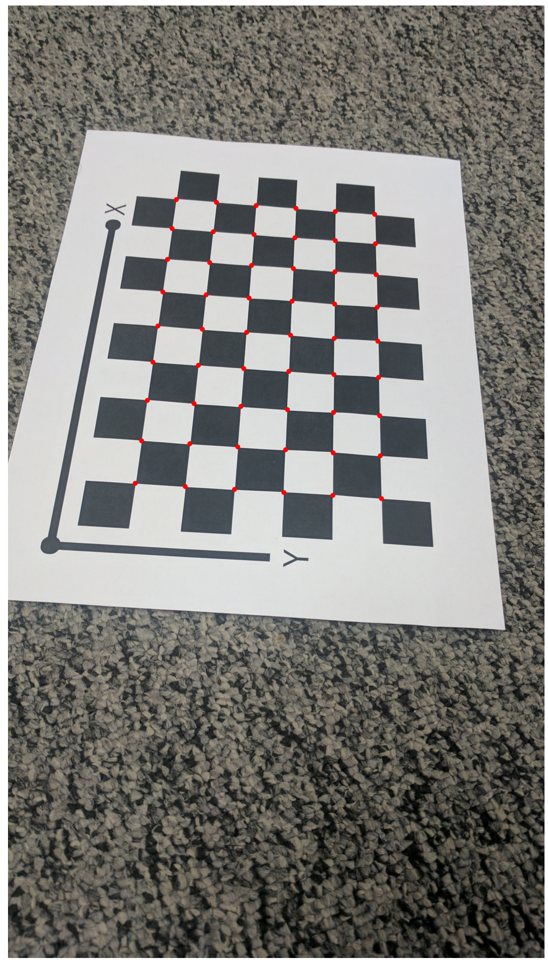

Fig. 9. Reprojected Image 3

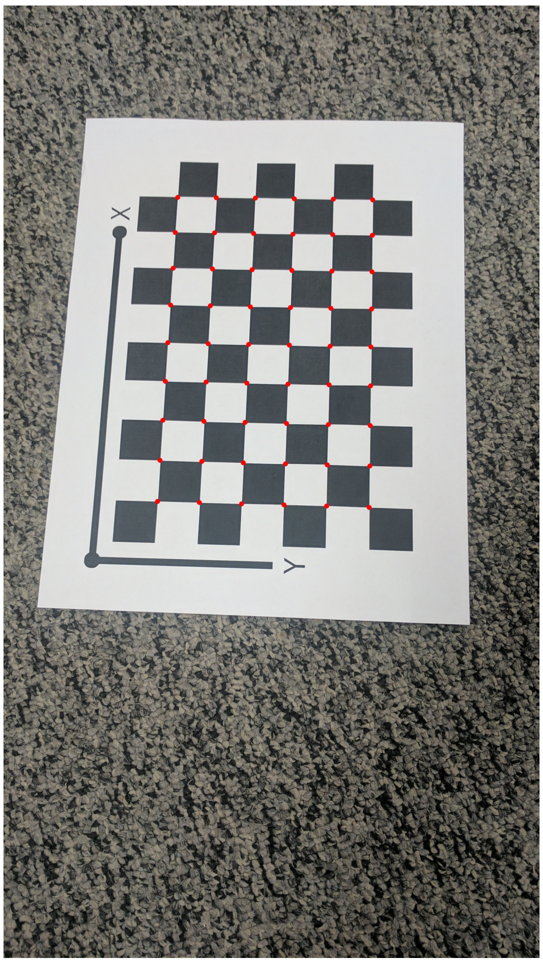

Fig. 10. Reprojected Image 4

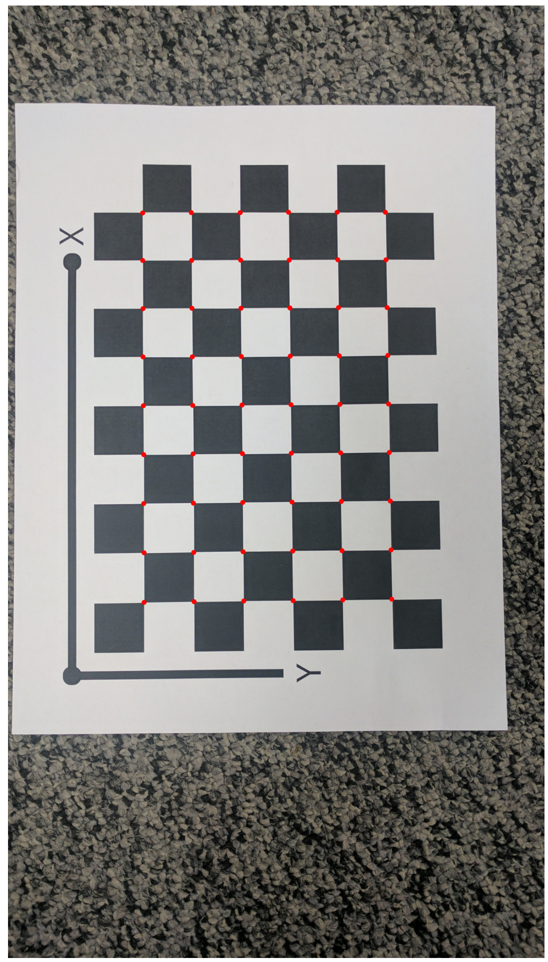

Fig. 11. Reprojected Image 5

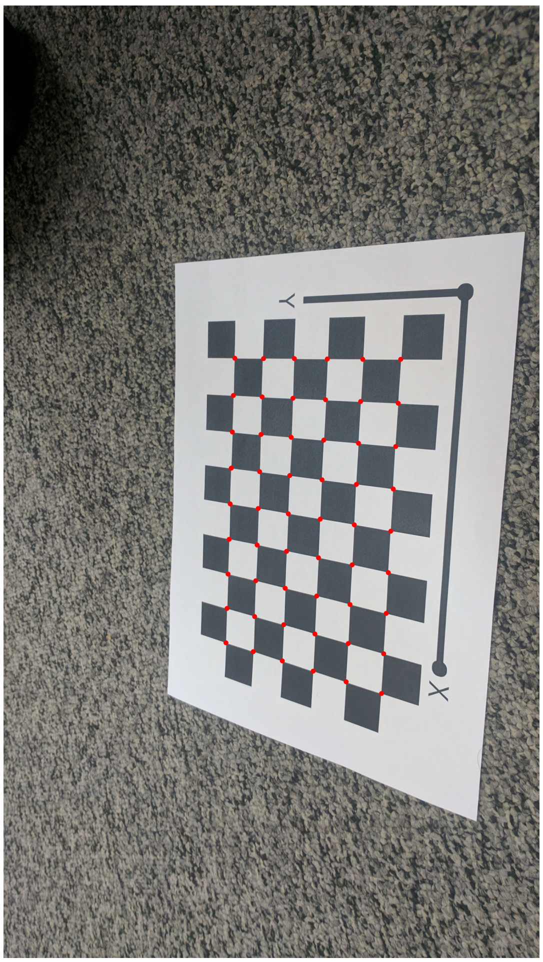

Fig. 12. Reprojected Image 6

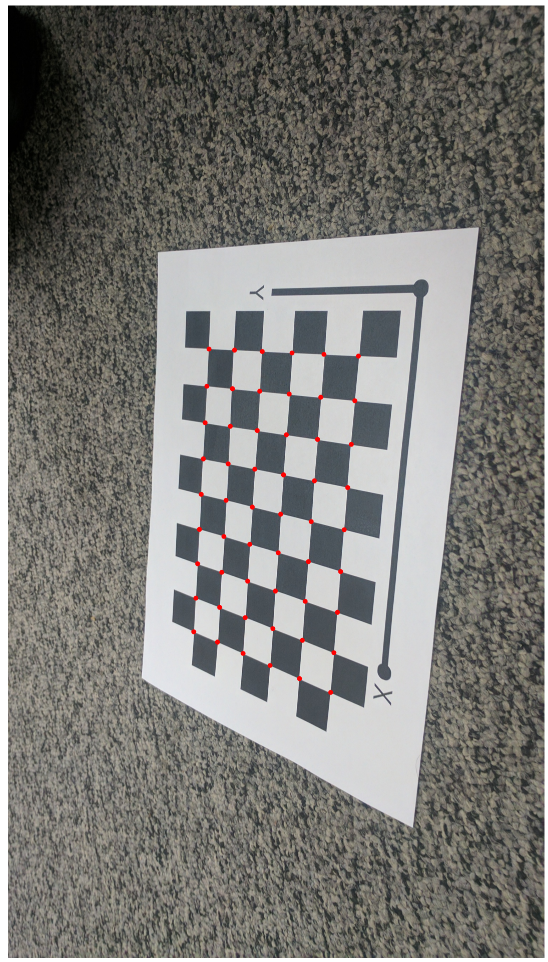

Fig. 13. Reprojected Image 7

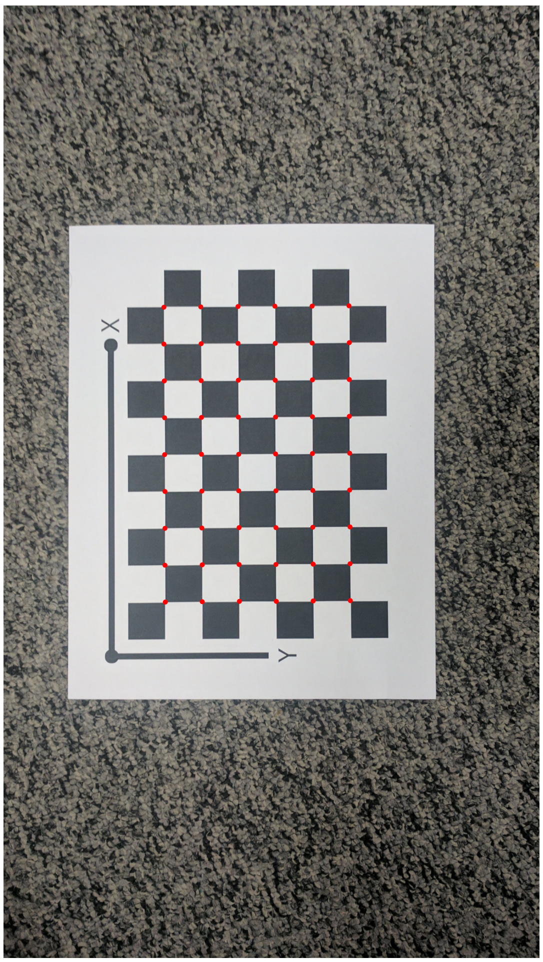

Fig. 14. Reprojected Image 8

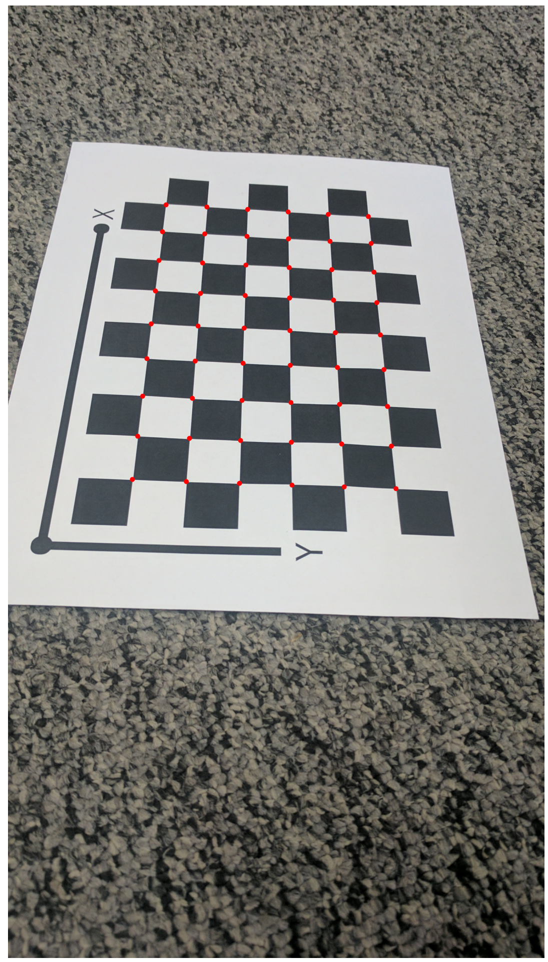

Fig. 15. Reprojected Image 9

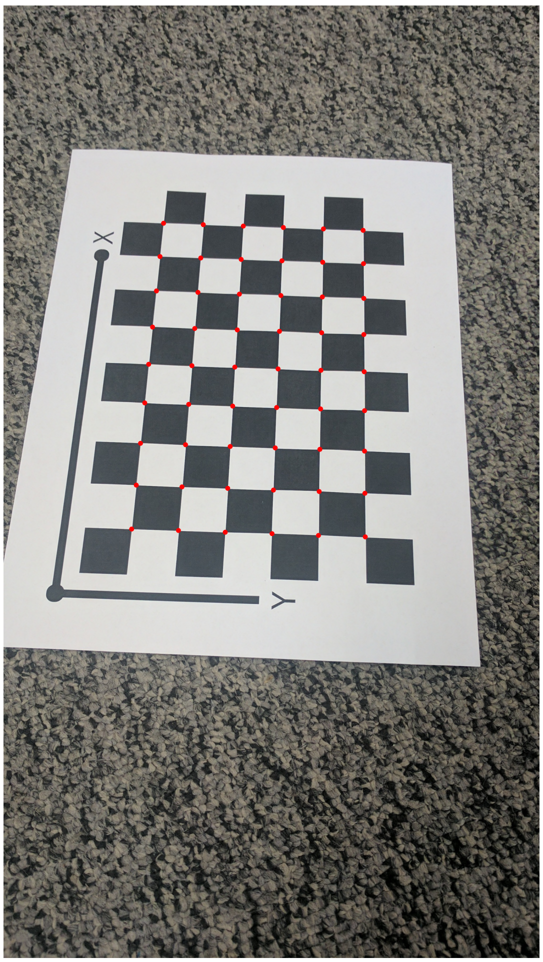

Fig. 16. Reprojected Image 10

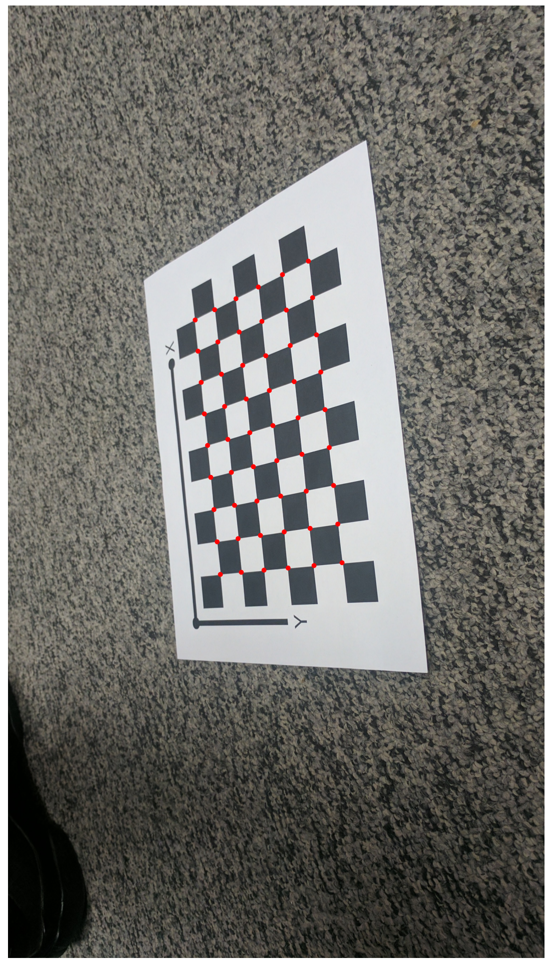

Fig. 17. Reprojected Image 11

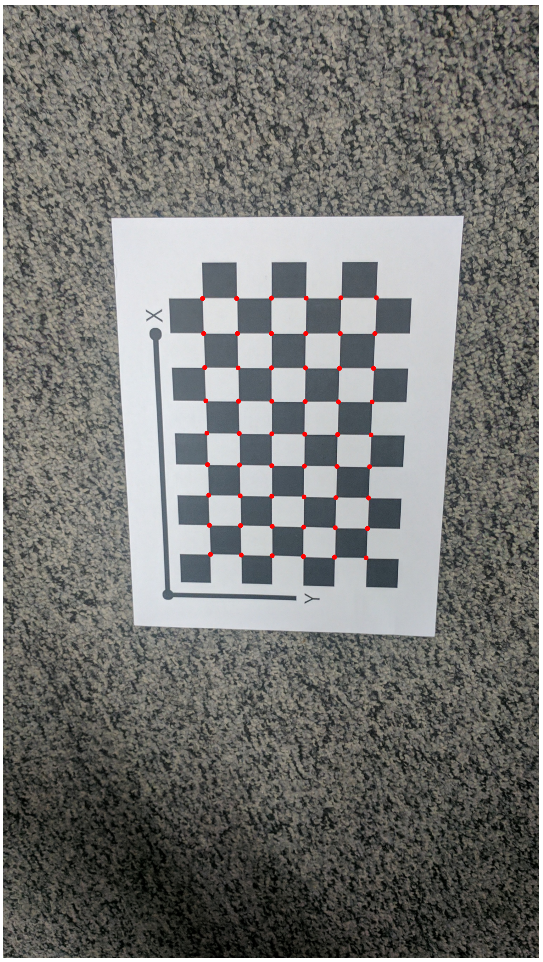

Fig. 18. Reprojected Image 12