University of Maryland

College Park

PROJECT 3

ENPM673 - PERCEPTION FOR AUTONOMOUS ROBOTS

Camera Calibration

Author:

Aaqib Barodawala (119348710)

Instructor and TAs:

Dr. Samer Charifa

Arunava Basu, Madhu NC,

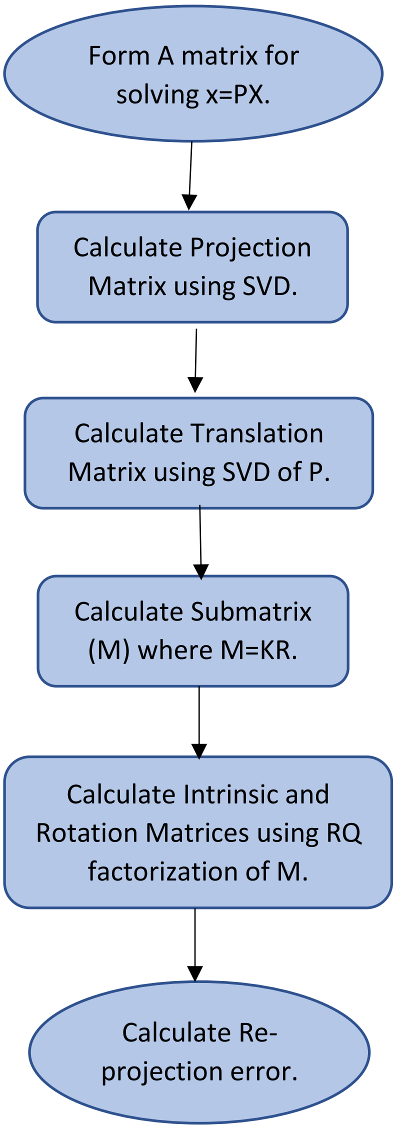
Ritvik Oruganti

PROBLEM 1:
Perform camera calibration and find the intrinsic matrix K. Find the minimum number of matching points to solve this mathematically. Provide a pipeline/block diagram to calibrate the camera with the given image. Write down the mathematical formation to find the intrinsic matrix K. Find the P matrix and decompose it to find translation, rotation and intrinsic matrices using Gram–Schmidt process and compute the reprojection error for each point.

SOLUTION:

1. Six matching points are required to solve this mathematically. Since intrinsic matrix K has 5 unknowns, Rotation matrix has 3 unknowns, Translation matrix has 3 unknowns, the total unknowns become 11 unknowns and therefore, we require 11 equations to get Projection matrix P. Hence, we need 6 matching points to solve this problem.

2. Block Diagram/Pipeline:



3. The solution begins with finding the Projection matrix (P) using the equation, $x = PX$. Since equation, $x_i = PX_i$, involves homogeneous coordinates where x_i are image points, X_i are world points and P is the projection matrix, x_i and PX_i must be proportional to each other. This means their cross product will equate to zero:

$$x_i \times PX_i = 0$$

Let p_1^T, p_2^T and p_3^T be the three row vectors of P. Therefore:

$$PX_i = \begin{bmatrix} p_1^T X_i \\ p_2^T X_i \\ p_3^T X_i \end{bmatrix}$$

$$x_i \times PX_i = \begin{bmatrix} v_i p_3^T X_i - w_i p_2^T X_i \\ w_i p_1^T X_i - u_i p_3^T X_i \\ u_i p_2^T X_i - v_i p_1^T X_i \end{bmatrix} = 0$$

Where u, v, w are the x, y, z image points.
x coordinate of the pixel = u/w
y coordinate of the pixel = v/w
$X = [X_i \quad Y_i \quad Z_i \quad 1]$

We can rewrite the above equation in the form $Ap = 0$, where A is of the form:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & -w.X_i & -w.Y_i & -w.Z_i & -w.1 & v.X_i & v.Y_i & v.Z_i & v.1 \\ w.X_i & w.Y_i & w.Z_i & w.1 & 0 & 0 & 0 & 0 & -u.X_i & -u.Y_i & -u.Z_i & -u.1 \\ -v.X_i & -v.Y_i & -v.Z_i & -v.1 & u.X_i & u.Y_i & u.Z_i & u.1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

After forming the A matrix, projection matrix P can be calculated by performing SVD of A. When using SVD, we get U, S, V where the last column of V is the solution of p. While using NumPy, it returns V^T after SVD. Hence, the last row of V^T will be the solution.

Since P matrix obtained is a 12×1 matrix, it is reshaped into a 3×4 matrix. After getting the P matrix, it is decomposed to Rotation (R), Translation (C) and Intrinsic (K) matrices.

The Translation matrix (C) is the null vector of P and can be found by performing SVD of P. Similarly, after getting U, S, V, the last column of V is the solution of p. After obtaining the C matrix, we can find the submatrix M by using the following equation:

$$M = P\left[I_3 \mid -\tilde{C}\right]^{-1}$$

Since $M = KR$, this non-singular matrix can be decomposed into upper-triangular Intrinsic matrix (K) and orthogonal Rotation matrix (R) using RQ factorization (Gram-Schmidt Process).
In Gram-Schmidt process, we use the following equations to project the row vectors of M into orthogonal space.

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{a}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\ \mathbf{u}_2 &= \mathbf{a}_2 - \mathrm{proj}_{\mathbf{u}_1}\,\mathbf{a}_2, & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\ \mathbf{u}_3 &= \mathbf{a}_3 - \mathrm{proj}_{\mathbf{u}_1}\,\mathbf{a}_3 - \mathrm{proj}_{\mathbf{u}_2}\,\mathbf{a}_3, & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \end{aligned}$$

Where, $\mathrm{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle}\mathbf{u}$

a_1, a_2, a_3 are the rows of matrix M.

Rotation matrix (R) can be calculated as follows:

$$R = [e_1 \quad e_2 \quad e_3]$$

Intrinsic matrix (K) can be calculated as follows:

$$K = MR^{-1}$$

To calculate the Re-projection error, we find new image points using the calculated Projection matrix. We use the formula:

$$x_{new} = PX$$

Calculating the Euclidean distance between the new image points and the corresponding given image points gives us the re-projection error for each point.

PROBLEMS FACED:

Performing NumPy operations correctly posed the biggest problem. Printing the matrices as well as using the debugger helped solved this issue.

RESULTS:

P:

```
[[ 2.87300079e+01 -1.88433521e+00 -6.98538406e+01  7.56880667e+02]
 [-2.01205702e+01  6.57695114e+01 -2.20906247e+01  2.13268011e+02]
 [-2.76739932e-02 -2.74983107e-03 -3.11534883e-02  1.00000000e+00]]
```

C:

```
[[16.01106826]
 [ 7.43927187]
 [17.21967888]
 [ 1.        ]]
```

CT:

```
[[  1.           0.           0.         -16.01106826]
 [  0.           1.           0.          -7.43927187]
 [  0.           0.           1.         -17.21967888]]
```

M:

```
[[ 2.87300079e+01 -1.88433521e+00 -6.98538406e+01]
 [-2.01205702e+01  6.57695114e+01 -2.20906247e+01]
 [-2.76739932e-02 -2.74983107e-03 -3.11534883e-02]]
```

R:

```
[[ 0.74737193  0.00548748 -0.66438324]
 [ 0.04784159 -0.99781461  0.04557607]
 [-0.6626812  -0.06584743 -0.74600116]]
```

K:

```
[[ 6.78713819e+01  7.10428035e-02  3.31962888e+01]
 [ 0.00000000e+00 -6.75951837e+01  2.54824021e+01]
```

[0.00000000e+00 0.00000000e+00 4.17606430e-02]]

Re-projection error for point (0, 0, 0) : 0.2934

Re-projection error for point (0, 3, 0) : 1.1361

Re-projection error for point (0, 7, 0) : 0.9218

Re-projection error for point (0, 11, 0) : 0.1249

Re-projection error for point (7, 1, 0) : 0.1219

Re-projection error for point (0, 11, 7) : 0.5552

Re-projection error for point (7, 9, 0) : 0.1023

Re-projection error for point (0, 1, 7) : 1.558

PROBLEM 2:

Perform camera calibration using OpenCV given multiple images of a checkerboard printed on a paper.

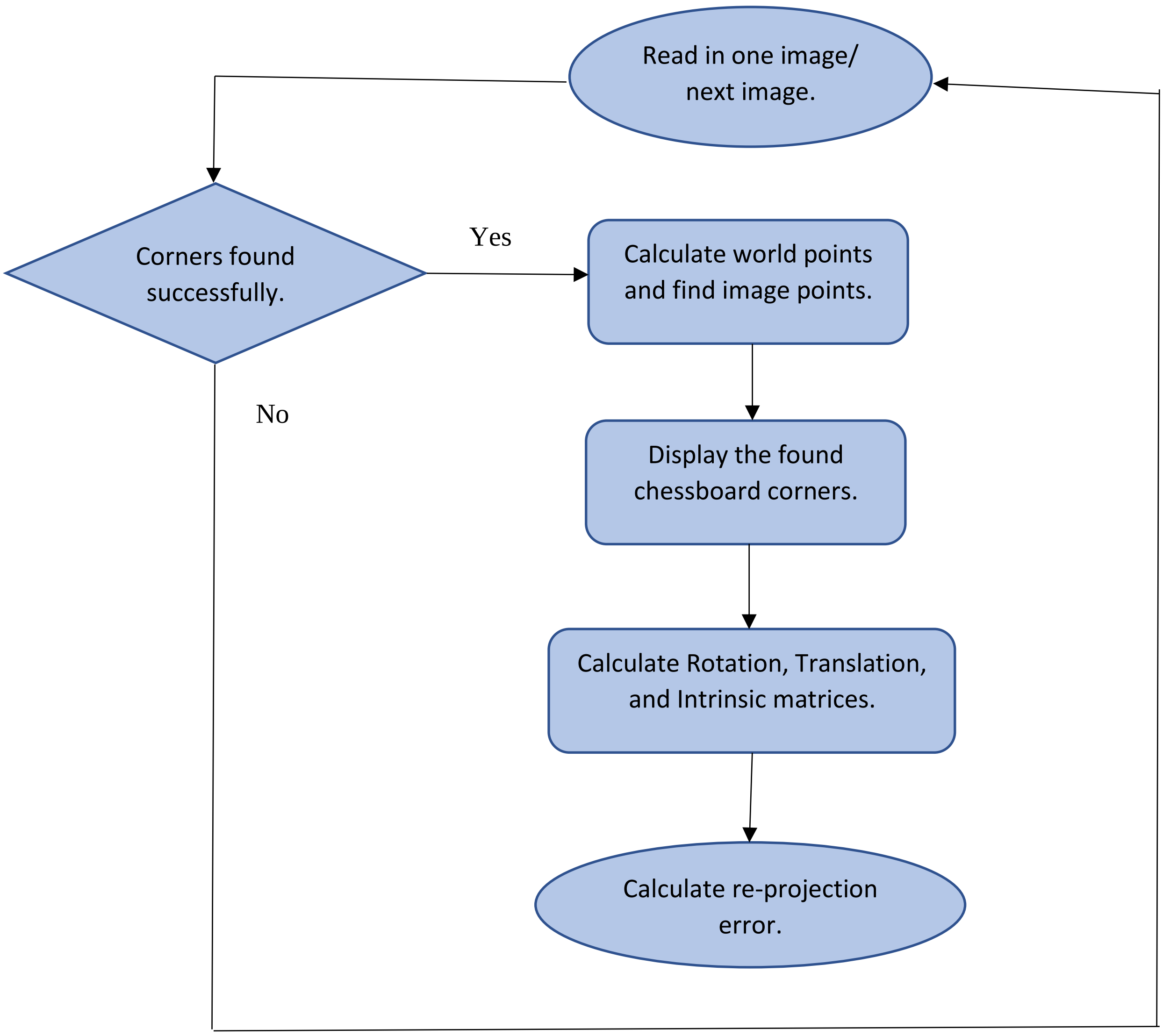
SOLUTION:

Steps involved to calibrate camera using OpenCV:

1. Calculate the world points of the checkerboard corners in 3D space and store them in a list where the points are arranged in a column-wise manner.
2. Find the image points of the checkerboard corners using OpenCV's "findChessBoardCorners()" function.
3. Calculate Rotation, Translation and Intrinsic matrices using OpenCV's "calibrateCamera()" function.
4. Find the new image points using OpenCV's "projectPoints()" function and calculate the re-projection error. The error is calculated by finding the Euclidean distances between each point and mean of the errors of all the points is calculated.

Disclaimer – This solution was adapted from OpenCV documentation.

Block Diagram /Pipeline:



IMPROVING ACCURACY:

To improve the accuracy of computed intrinsic matrix (K), take more images of the checkerboard from different angles and different distances. Also, remove distortions and exclude images with high re-projection errors and recalibrate. Furthermore, use a checkerboard with more squares and make sure the images are clear (optimal lighting) with no blurring.

PROBLEMS FACED:

Major problem faced was understanding the underlying functions used. After research and practising the said functions, the process was understood, and the problem was resolved.

RESULTS:

Intrinsic Matrix (K) - image 1:

[[3.28564301e+03 0.00000000e+00 6.83139844e+02]

 [0.00000000e+00 3.44445038e+03 1.44351255e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 1: 0.051

Re-projection error for image 1: 0.051

Intrinsic Matrix (K) - image 2:

[[1.82591695e+03 0.00000000e+00 7.67290944e+02]

 [0.00000000e+00 1.83080808e+03 1.45281177e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 2: 0.078

Re-projection error for image 2: 0.0785

Intrinsic Matrix (K) - image 3:

[[1.89815954e+03 0.00000000e+00 7.73122880e+02]

 [0.00000000e+00 1.89995367e+03 1.42536223e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 3: 0.0898

Re-projection error for image 3: 0.0961

Intrinsic Matrix (K) - image 4:

[[1.96224260e+03 0.00000000e+00 7.78563797e+02]

 [0.00000000e+00 1.96133817e+03 1.39781841e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 4: 0.1014

Re-projection error for image 4: 0.1206

Intrinsic Matrix (K) - image 5:

[[2.00277908e+03 0.00000000e+00 7.79677978e+02]

 [0.00000000e+00 1.99488002e+03 1.36835214e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 5: 0.0932

Re-projection error for image 5: 0.0876

Intrinsic Matrix (K) - image 6:

[[1.99954367e+03 0.00000000e+00 7.71267963e+02]

 [0.00000000e+00 1.99154878e+03 1.36795251e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 6: 0.0901

Re-projection error for image 6: 0.0491

Intrinsic Matrix (K) - image 7:

[[2.01172966e+03 0.00000000e+00 7.70515208e+02]

 [0.00000000e+00 2.00384765e+03 1.36219428e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 7: 0.0824

Re-projection error for image 7: 0.0272

Intrinsic Matrix (K) - image 8:

[[2.03430194e+03 0.00000000e+00 7.68079118e+02]

[0.00000000e+00 2.02551574e+03 1.35104071e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 8: 0.0788

Re-projection error for image 8: 0.0432

Intrinsic Matrix (K) - image 9:

[[2.04203700e+03 0.00000000e+00 7.68029826e+02]
 [0.00000000e+00 2.03353746e+03 1.35268945e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 9: 0.078

Re-projection error for image 9: 0.062

Intrinsic Matrix (K) - image 10:

[[2.04179380e+03 0.00000000e+00 7.67198899e+02]
 [0.00000000e+00 2.03343849e+03 1.35244407e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 10: 0.0783

Re-projection error for image 10: 0.0542

Intrinsic Matrix (K) - image 11:

[[2.04435120e+03 0.00000000e+00 7.65583359e+02]
 [0.00000000e+00 2.03625106e+03 1.35358574e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 11: 0.0814

Re-projection error for image 11: 0.0855

Intrinsic Matrix (K) - image 12:

[[2.04263857e+03 0.00000000e+00 7.64266106e+02]
 [0.00000000e+00 2.03470449e+03 1.35561706e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 12: 0.085

Total re-projection error for image 12: 0.1006

Intrinsic Matrix (K) - image 13:

[[2.04272943e+03 0.00000000e+00 7.64360022e+02]

 [0.00000000e+00 2.03501640e+03 1.35902591e+03]

 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Total re-projection error till image 13: 0.0876

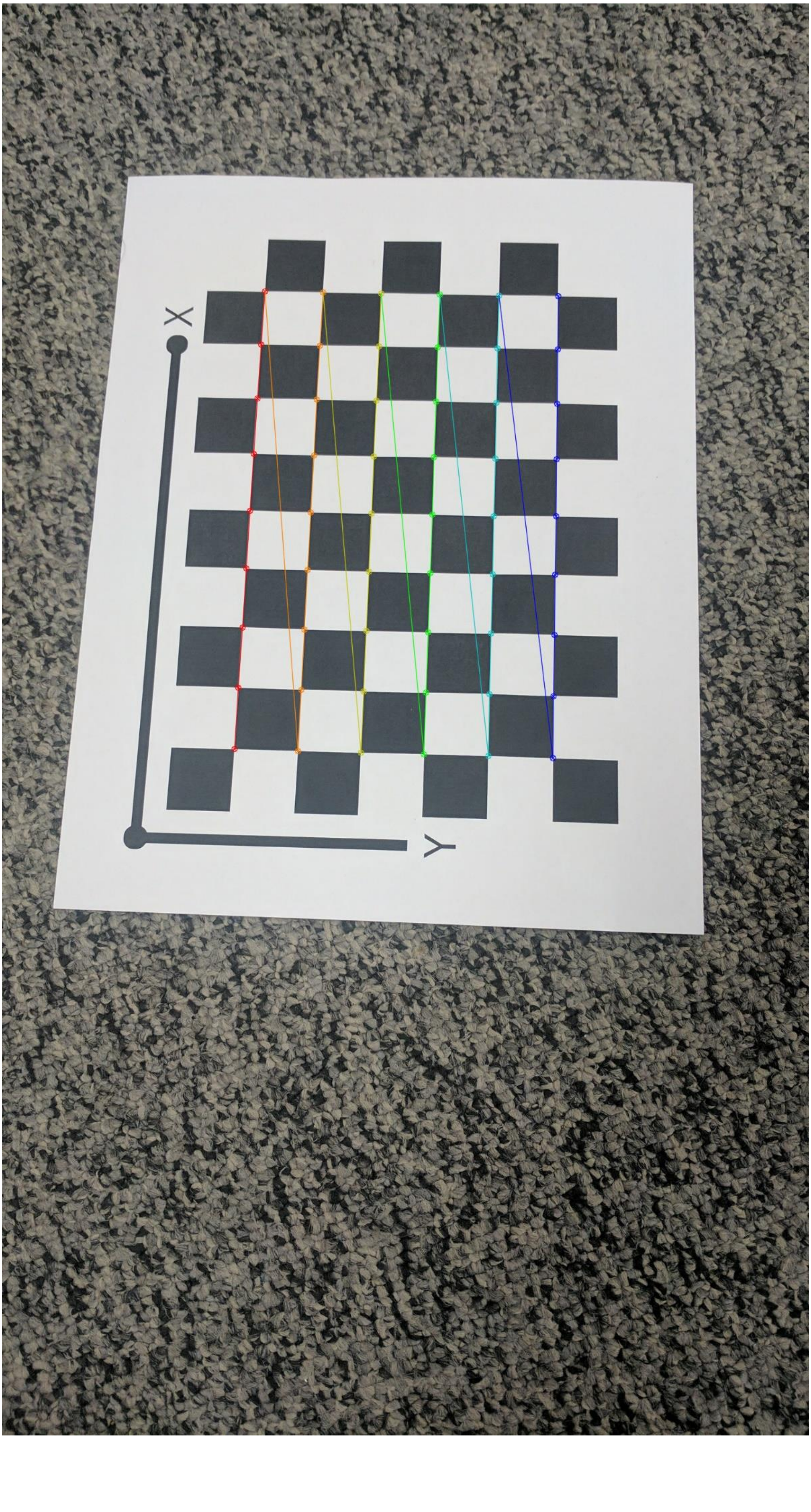
Total re-projection error for image 13: 0.1034

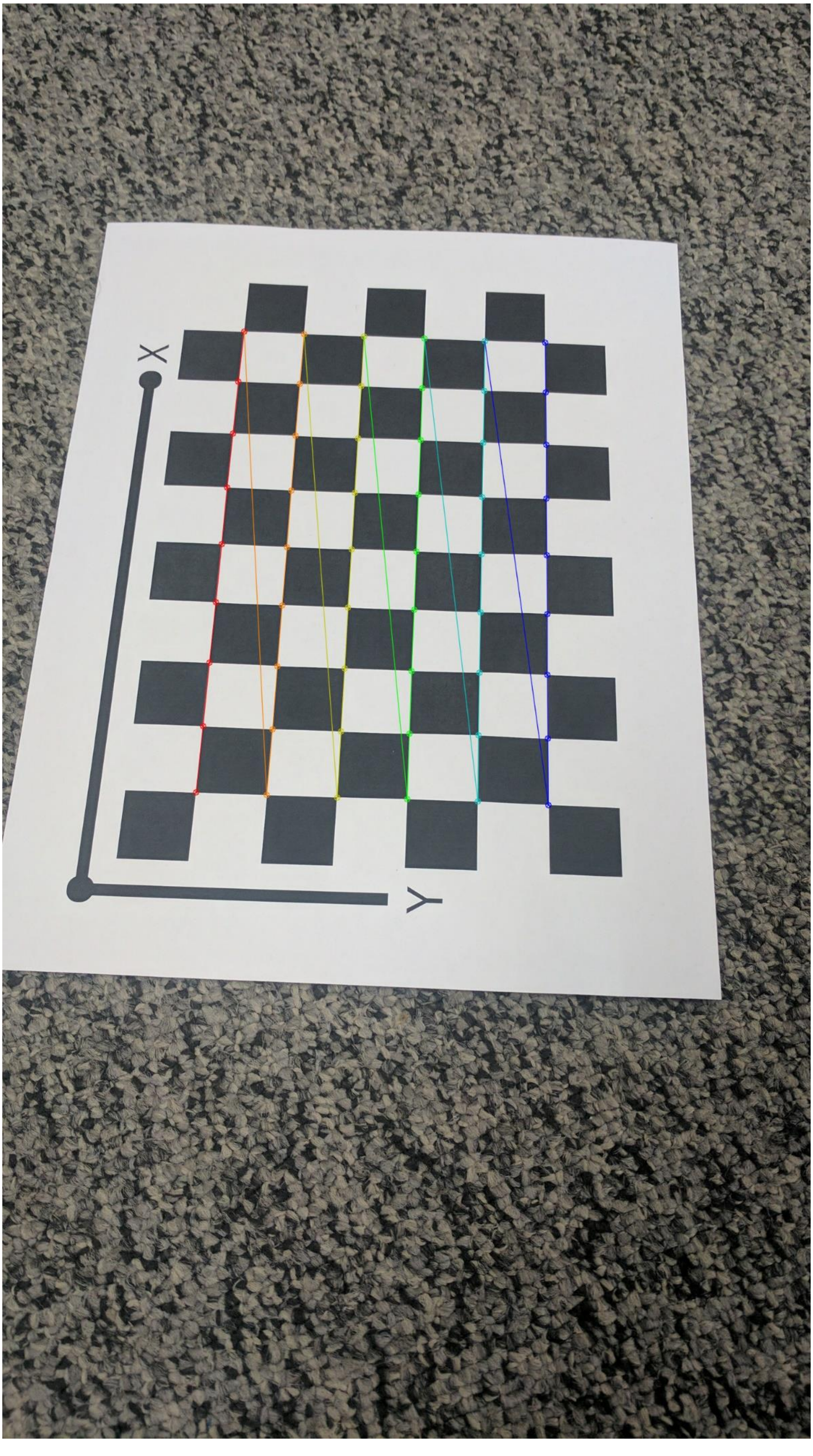
Final Intrinsic Matrix (K):

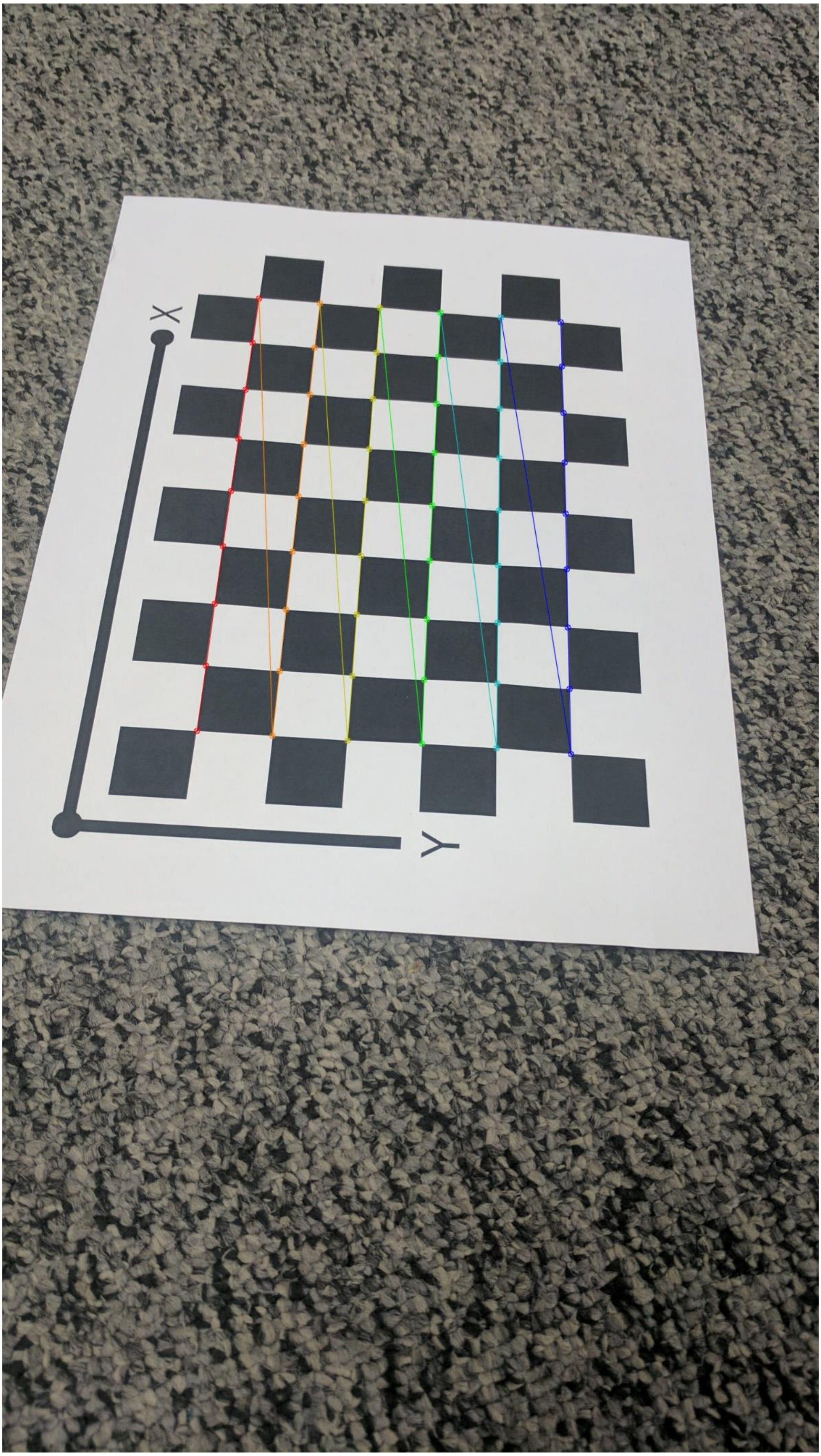
[[2.04272943e+03 0.00000000e+00 7.64360022e+02]

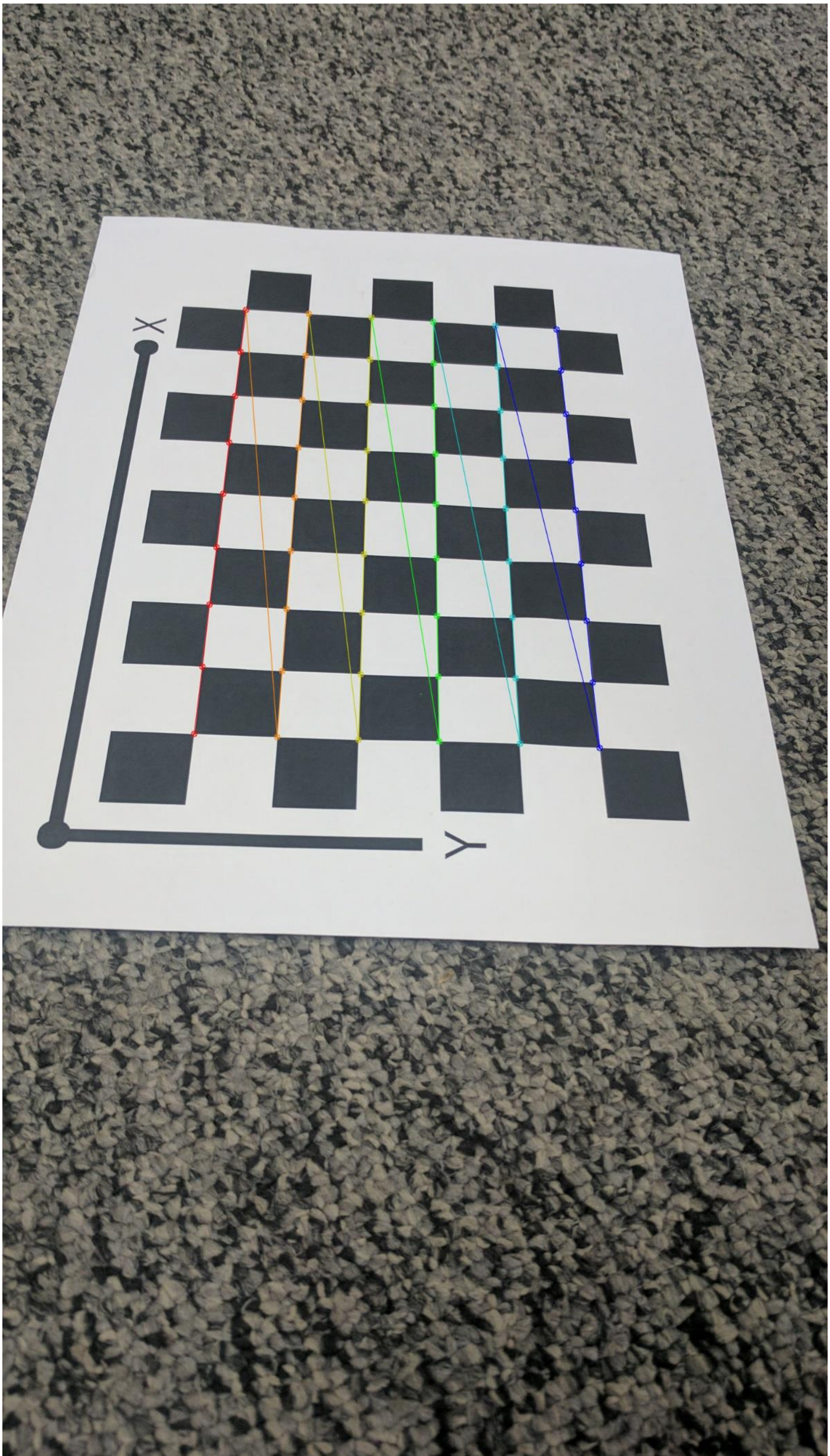
 [0.00000000e+00 2.03501640e+03 1.35902591e+03]

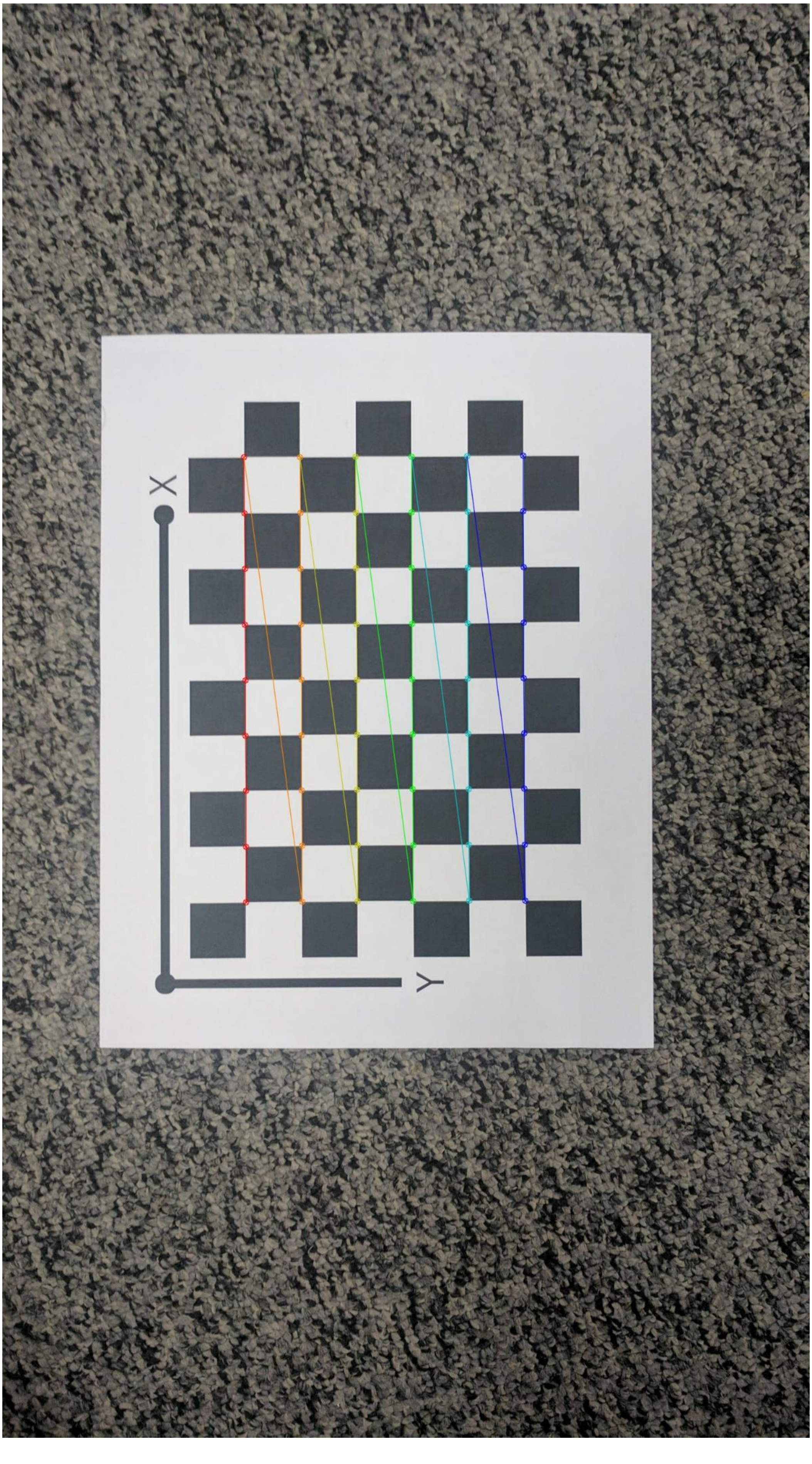
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

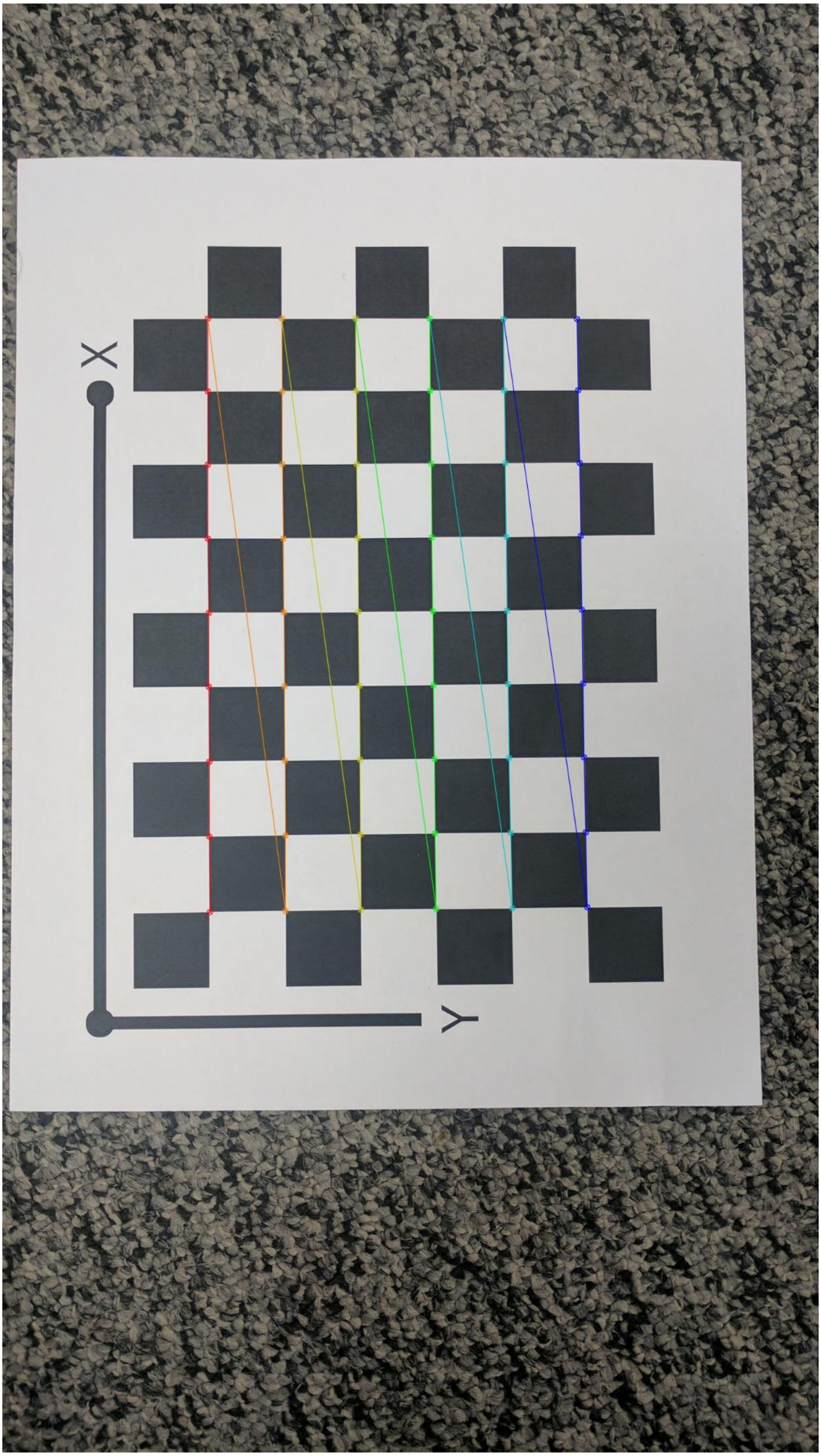

X
Y

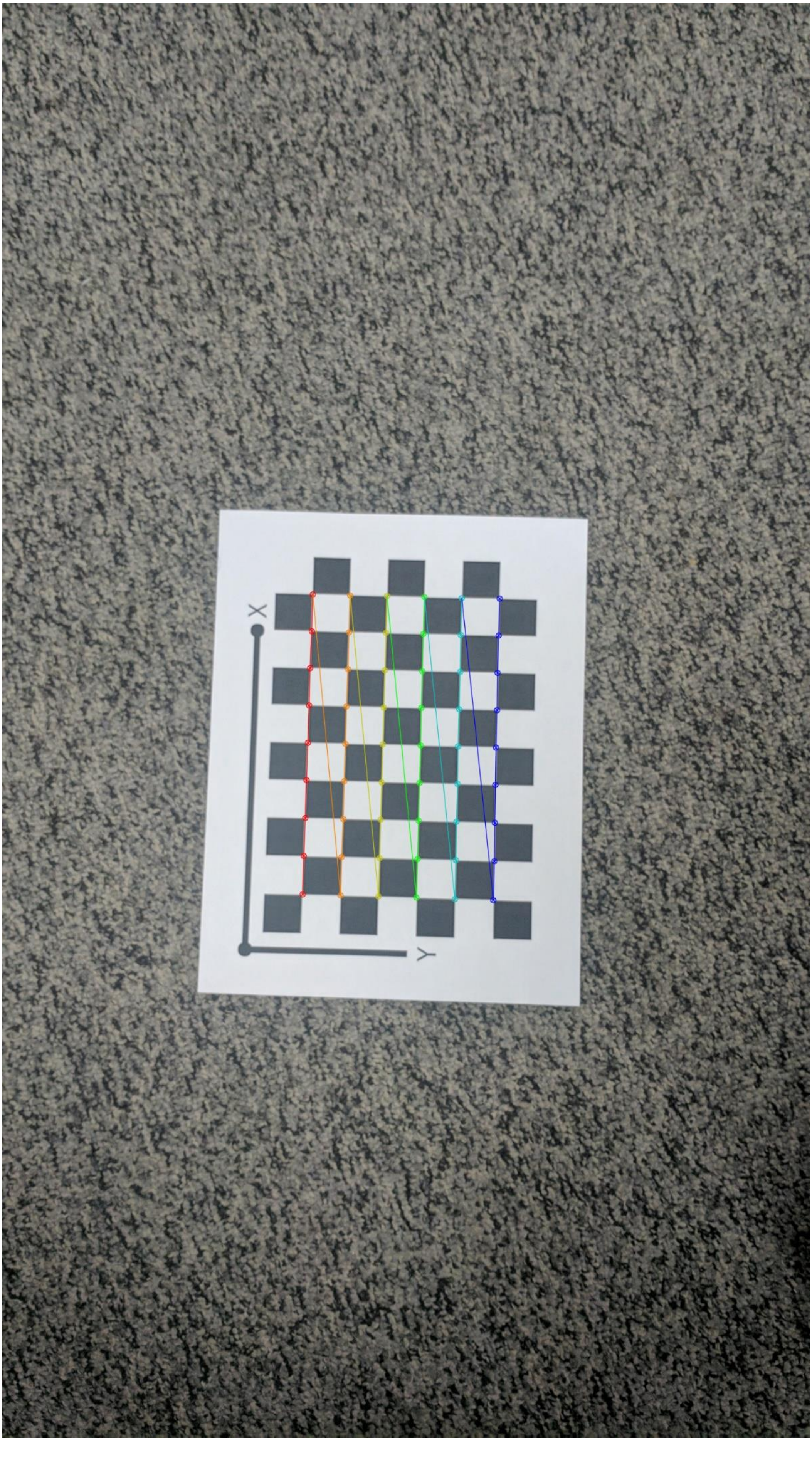

X
Y

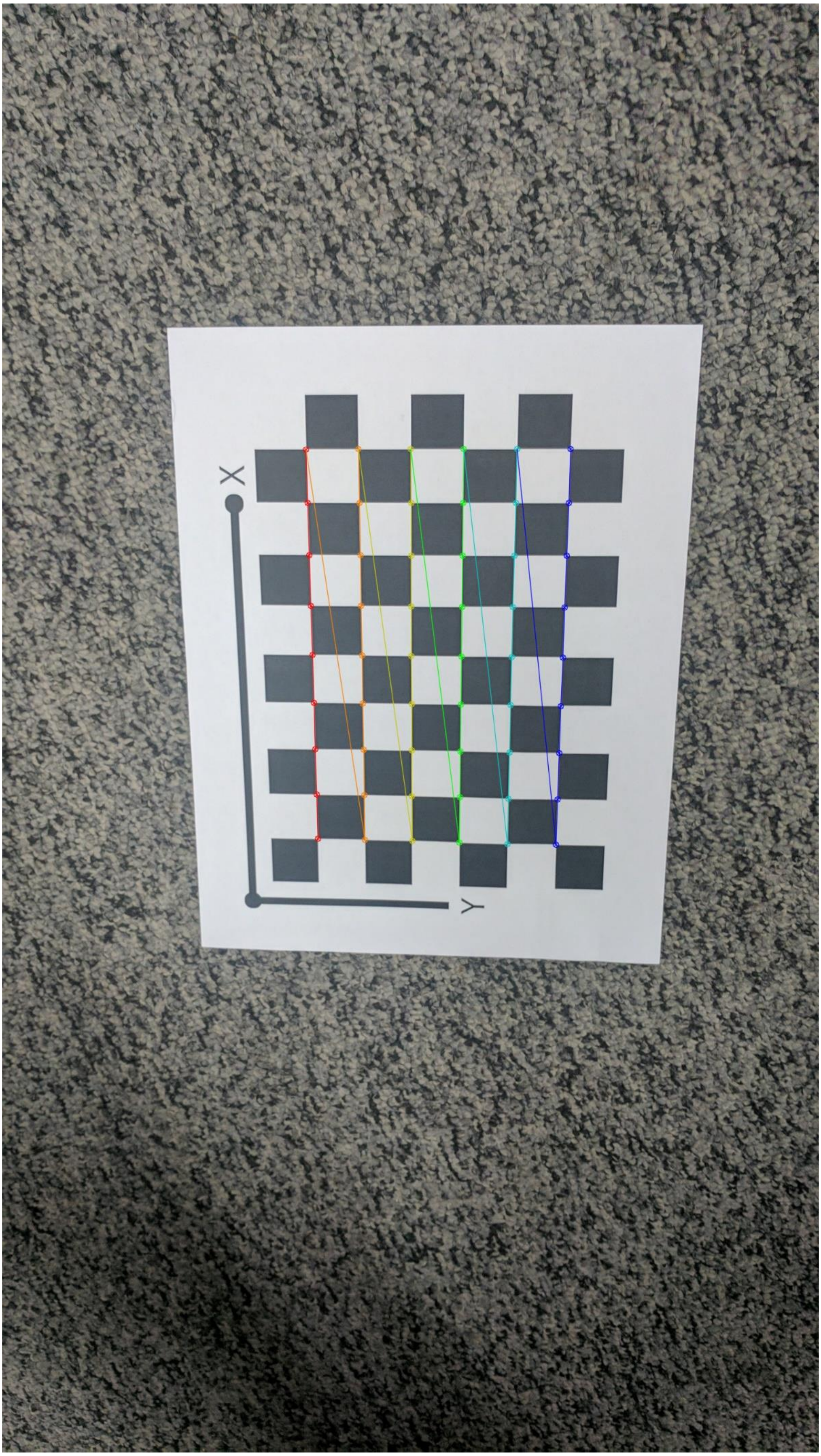

X
Y

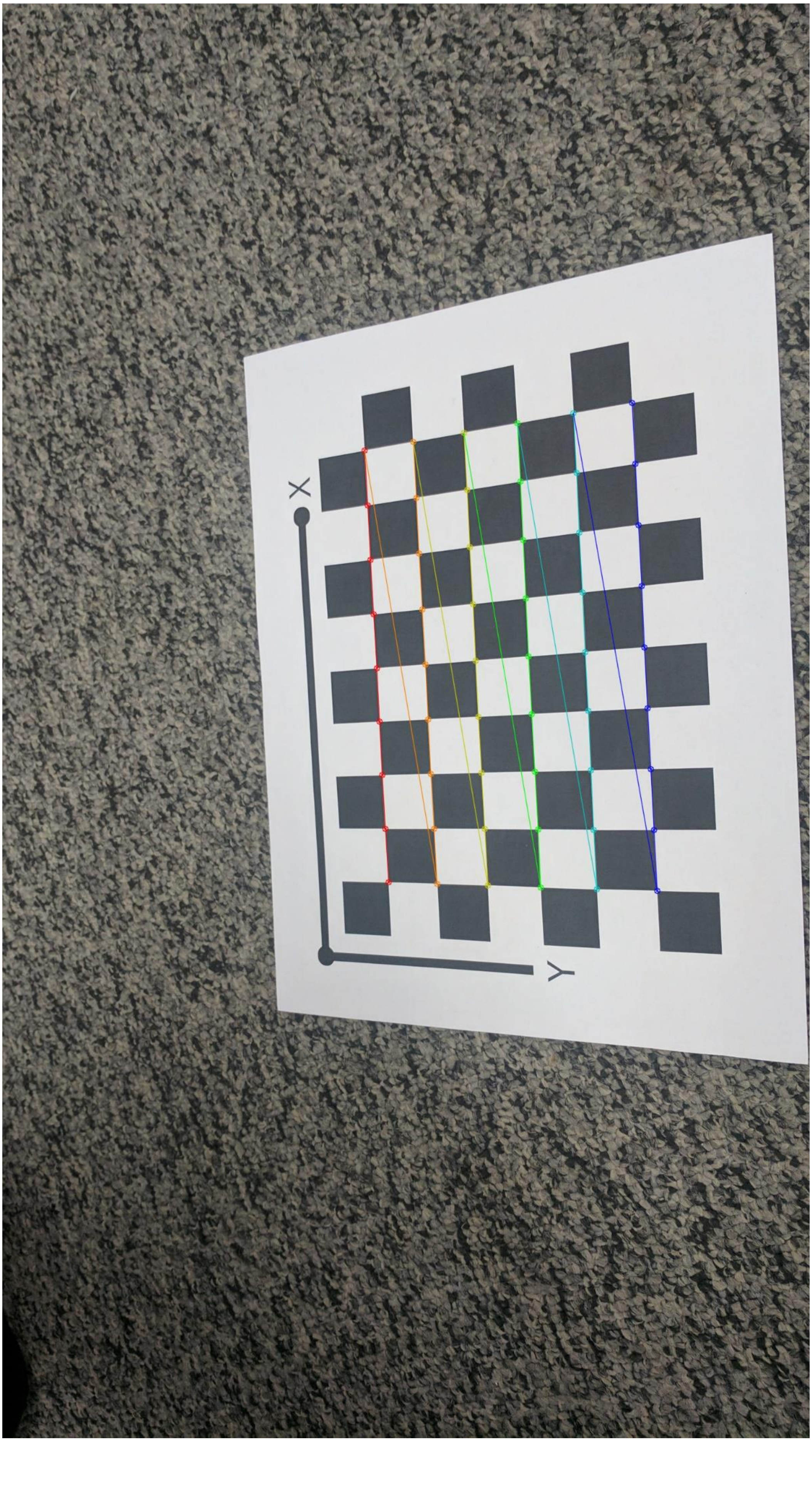

X
Y

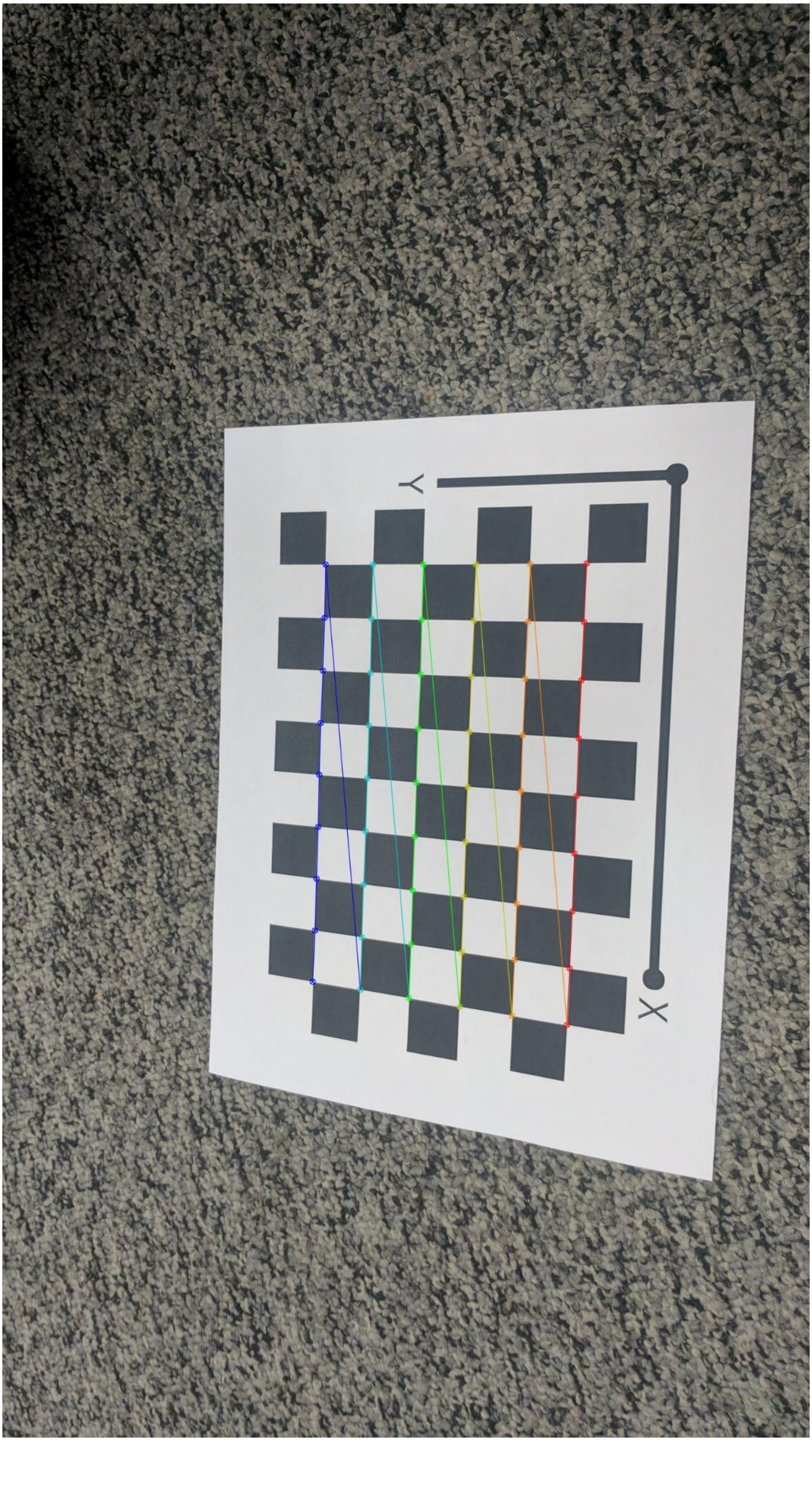

X
Y

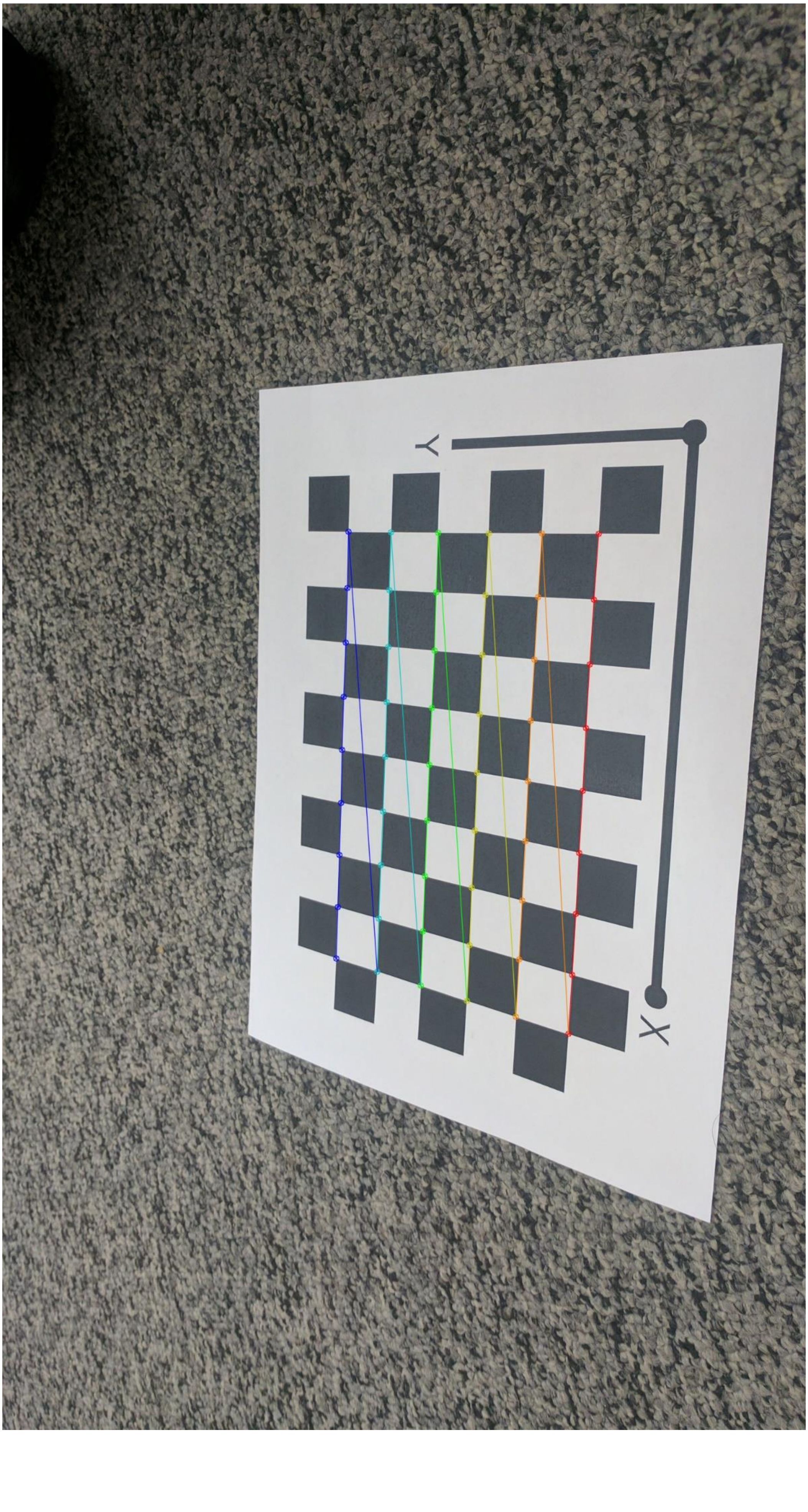

X
Y

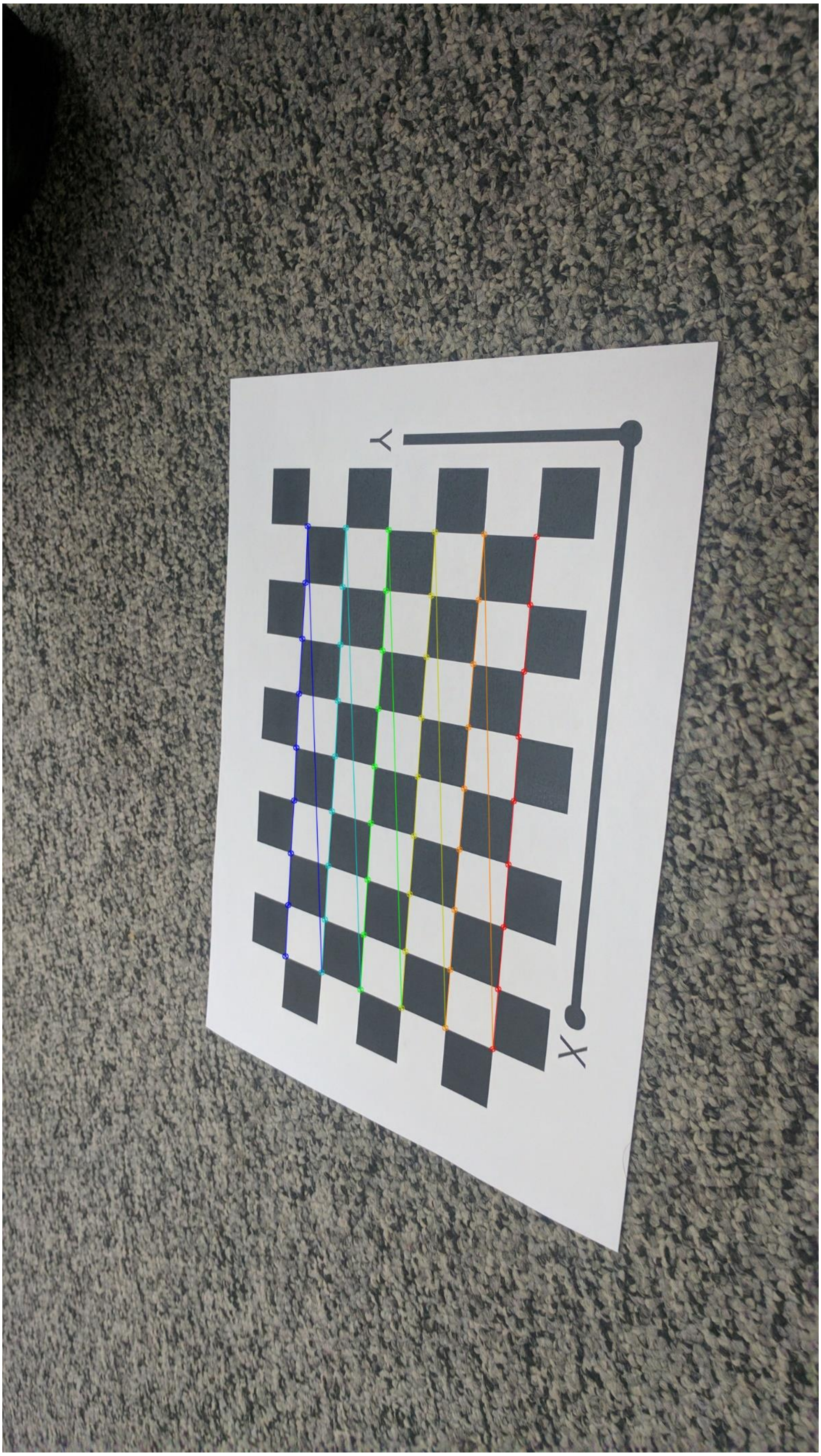

X
Y

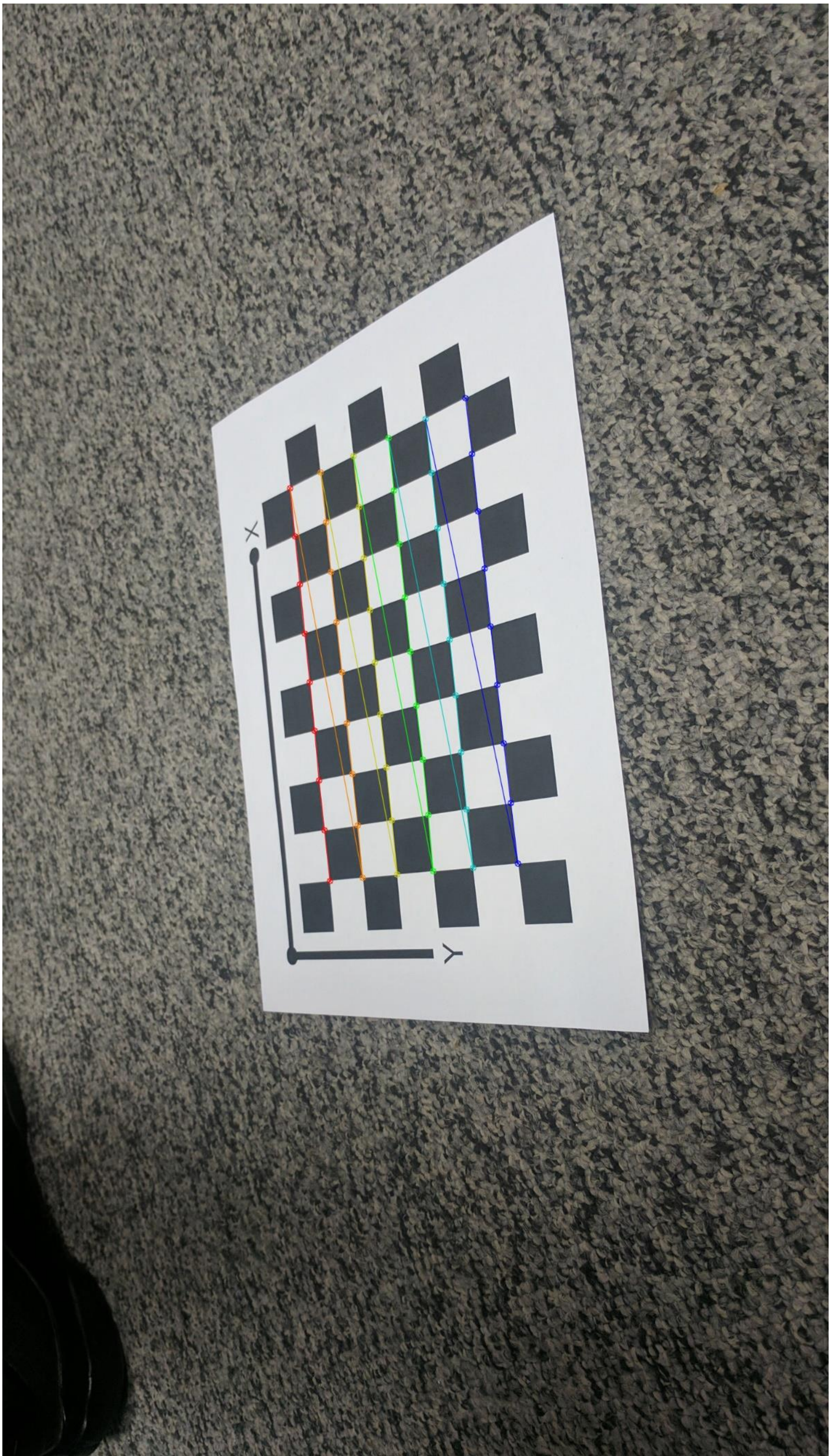

X
Y


X
Y


Y
X


Y
X


Y
X


X
Y