

Computer Vision Assignment One @ ETH Zurich

Calibration

Ossama Ahmed

October 4, 2018

1 Introduction

A camera is used to project 3D world points onto 2D image plane. Camera calibration is an essential step in many computer vision applications and it can be considered the first step in most of them. Calibration is the process to find the internal camera parameters that is responsible for this projection process. These parameters are:

1. - Image center
2. - Focal length
3. - Lens distortion parameters

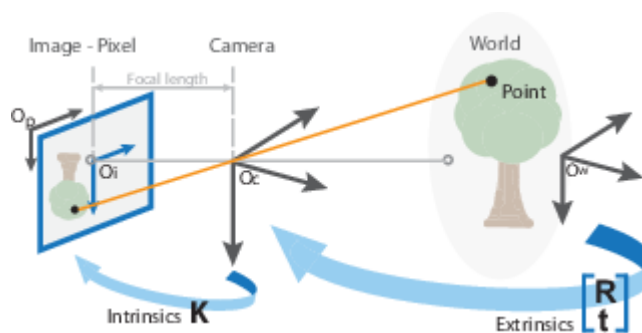


Figure 1: Calibration parameters

In this assignment two widely known calibration algorithms, Direct Linear Transform (DLT) and Gold Standard Algorithm, are implemented in MATLAB and discussed in comparison to using the standard calibration toolbox for MATLAB, Bouget's Calibration Toolbox.

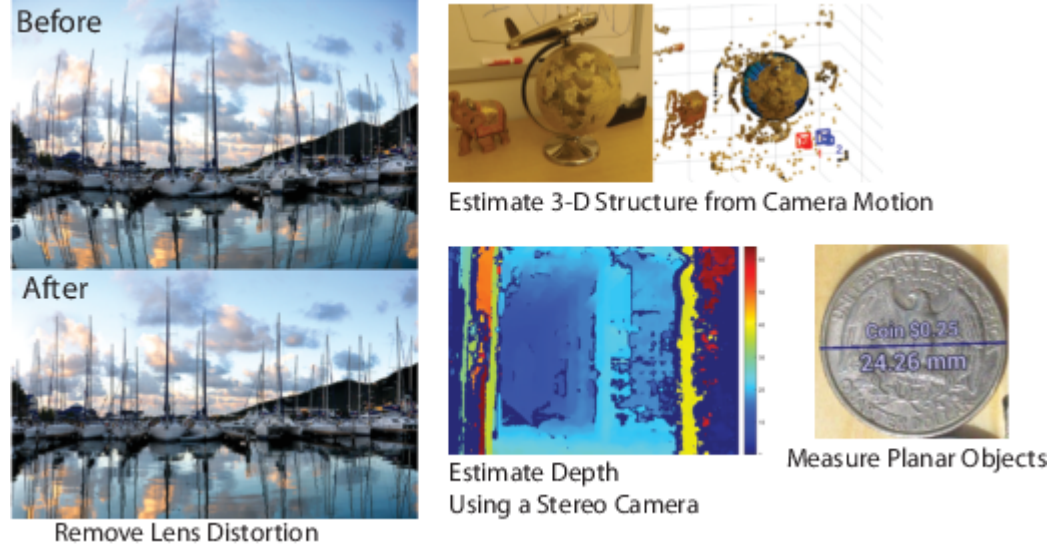


Figure 2: Camera calibration usages

2 Data Normalization

Data normalization is the first step that was done before implementing the two calibration algorithms mentioned above. Since the points were entered in cartesian coordinates, they were converted to homogenous vectors by just appending 1 in the $n+1$ dimension. Therefore we ended up with $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{X} = (x, y, z, 1)^T$. To transform the points centroid entered manually on the image to the center of the coordinate system, the current centroid was calculated by averaging out all the points coordinates in every direction so that we can shift the points by the mean.

Secondly, to ensure that the distance from the origin was $\sqrt{2}$ for the image points and $\sqrt{3}$ for the object points, I calculated the current average distance of all points to the origin to calculate the scale to multiply the points with.

After calculating the centroid of the image and object points as well as the scale to ensure the average distance to the origin is $\sqrt{2}$ and $\sqrt{3}$, the transformation matrices are calculated as the following:

$$U = \begin{bmatrix} xy_{scale} & 0 & -xy_{scale} * -xy_{centroid_x} \\ 0 & xy_{scale} & -xy_{scale} * -xy_{centroid_y} \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$T = \begin{bmatrix} xyz_{scale} & 0 & 0 & -xyz_{scale} * -xyz_{centroid_x} \\ 0 & xyz_{scale} & 0 & -xyz_{scale} * -xyz_{centroid_y} \\ 0 & 0 & xyz_{scale} & -xyz_{scale} * -xyz_{centroid_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3 Direct Linear Transform (DLT)

The direct linear transform algorithm is used to minimize the algebraic error of the linear system of the camera model. Following the steps specified in the assignment results in the computation of the intrinsic camera matrix K, a rotation matrix R and the camera center C which is then used to calculate the translation matrix since $t = -R * C$.

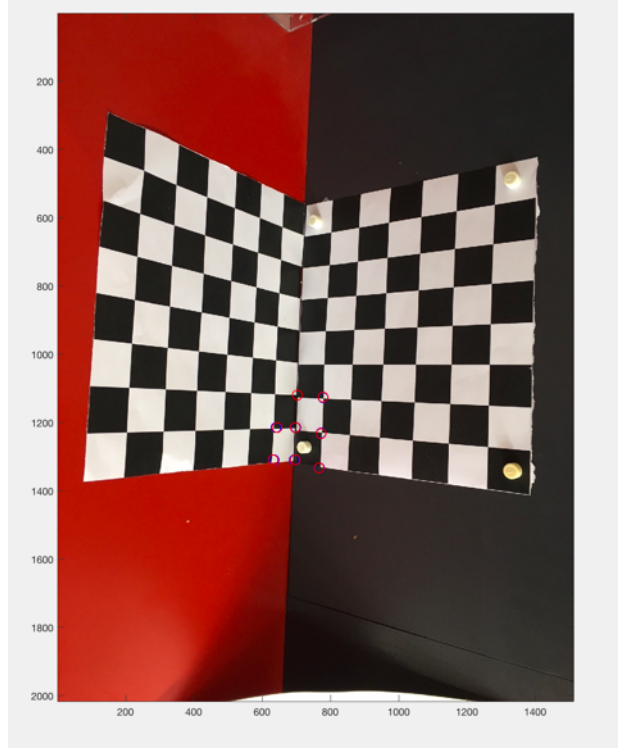


Figure 3: Reprojected points of the DLT algorithm; the red points are the original picked ones and the blue points are the reprojected ones

The following parameters are calculated using the DLT algorithm:

$$K = \begin{bmatrix} -752.8 & -19.8 & -4.9 \\ 0 & 1087.4 & 1081.8 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} -0.9931 & 0.1148 & 0.0225 \\ 0.0190 & -0.0309 & 0.9993 \\ 0.1154 & 0.9929 & 0.0285 \end{bmatrix}$$

$$t = \begin{bmatrix} 290.4564 \\ -64.9369 \\ -311.3866 \end{bmatrix}$$

Finally, I calculated the average error by calculating the average distance between the reprojected points and the original points. $average_{error} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = 2.0837pixels$

Using the unnormalized points gave a smaller error strangely $average_{error} = 2.0474pixels$, this might be due to overfitting. Also, the negative values in the intrinsic matrix K along the diagonal might be because of the choice of the image and the axis drawn, the lower line is not a straight line like the sample image.

4 Gold Standard Algorithm

The gold standard algorithm is used to refine the estimated camera matrix \hat{p} by minimizing the cost function which is the distance between the measured clicked point and the reprojection of the same point. $cost = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ We use the `fminsearch` function in MATLAB to optimize for this cost function.

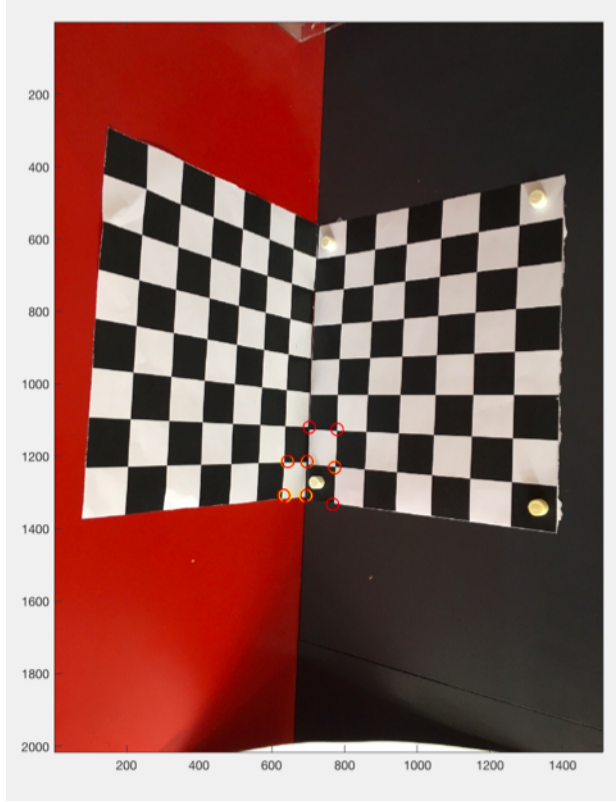


Figure 4: Reprojected points of the golden standard algorithm; ; the red points are the original picked ones and the yellow points are the reprojected ones

The following parameters are calculated using the Gold standard algorithm:

$$K = \begin{bmatrix} -635.1 & -1.8 & -59.5 \\ 0 & 1057.2 & 1131.5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} -0.9931 & 0.0218 & 0.0040 \\ 0.0025 & -0.0670 & 0.9977 \\ 0.0220 & 0.9975 & 0.0669 \end{bmatrix}$$

$$t = \begin{bmatrix} 359.4419 \\ -50.7024 \\ -303.5279 \end{bmatrix}$$

Finally, I calculated the average error by calculating the average distance between the reprojected points and the original points.

$$average_{error} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = 1.8200 pixels$$

5 Bouget's Calibration Toolbox

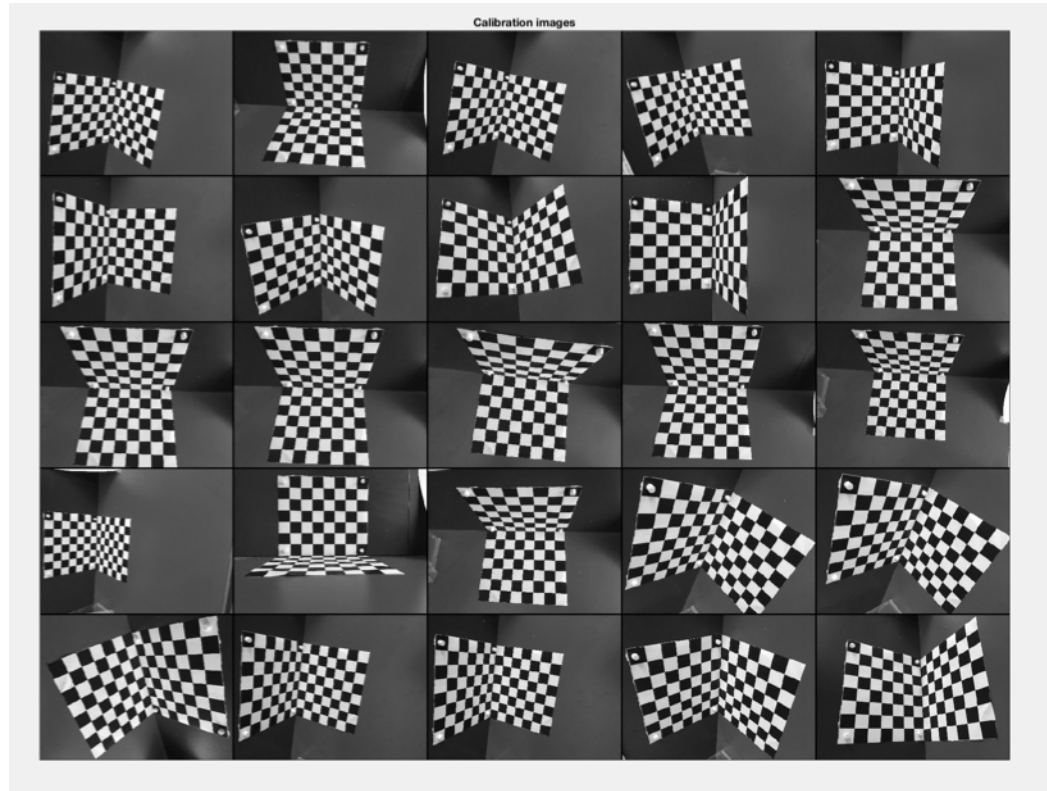


Figure 5: Input Images

I used 25 images to follow the tutorial and calibrate the camera, the only used one of the checkerboards printed in the images for calibration and marking the corners since I had to use the same origin for all the images.

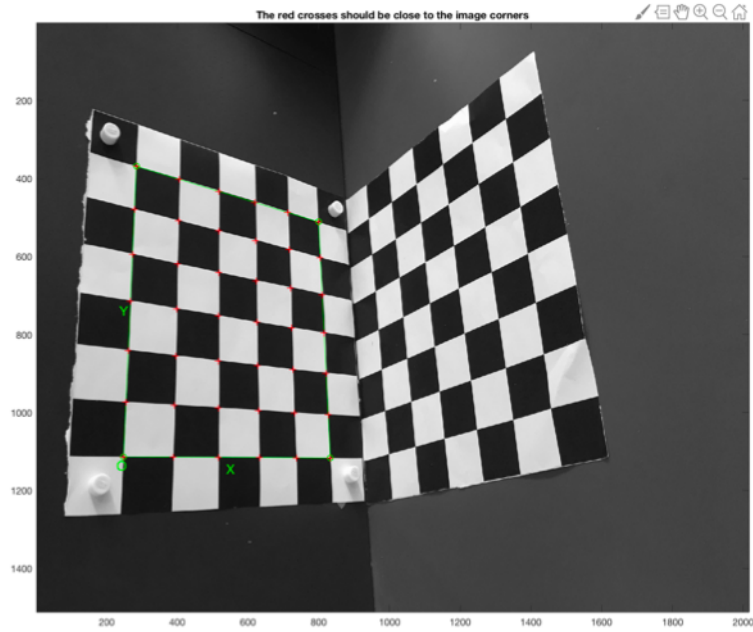


Figure 6: Marking corners

After marking the corners and using the toolbox calibration algorithm, I got the following results:

$$focallength = [1703.086 \quad 1703.086]$$

$$principlepoint = [1007.5 \quad 755.5]$$

$$skew = [0.0]$$

$$distortion = [0.0 \quad 0.0 \quad 0.0 \quad 0.0]$$

$$error = [0.7523 \quad 0.9289]$$

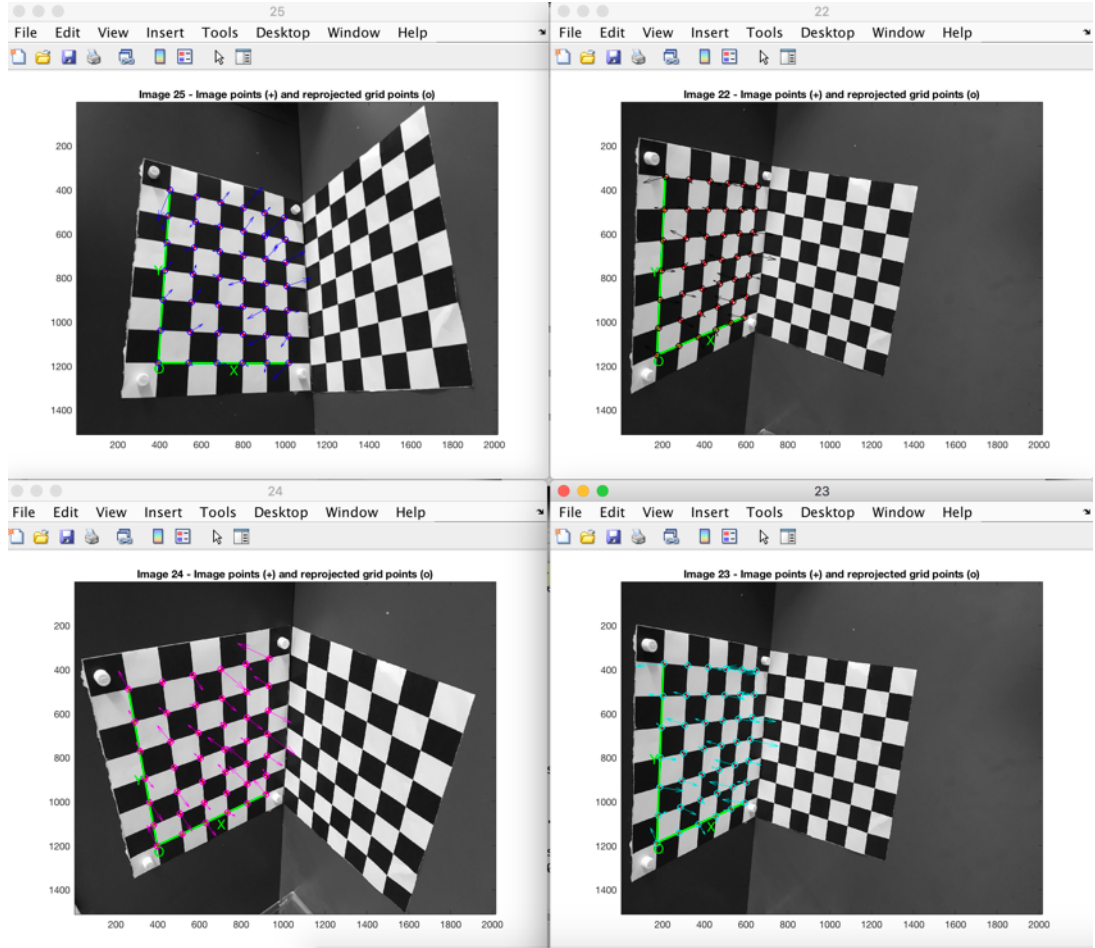


Figure 7: After calibration and reprojection

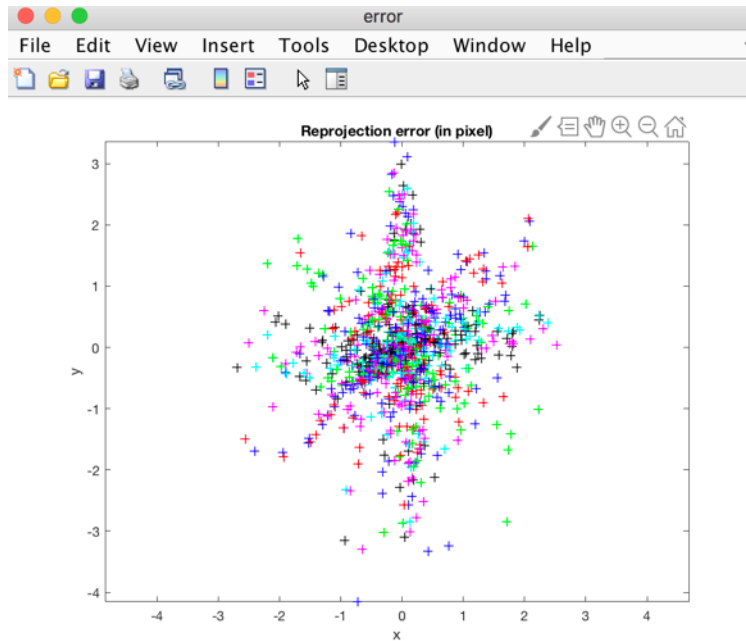


Figure 8: Calibration error

6 Conclusion

The error in the gold standard algorithm is much lower than the DLT algorithm error by almost 13% and using the bouget calibration toolbox was more accurate decreasing the error by more than 50%. One of the reasons is that using the toolbox, there is more data points available compared to just using 8 data points from one image (in the case of the DLT and the gold standard ones). Also I found it challenging to manually entering the cartesian coordinates for the image that I chose since the axes drawn on the picture didnt match the checkerboard edges; which resulted in some negative values in the x direction of the intrinsic matrix, given that there is an illusion that part of the checkerboard is behind the camera.