Computer Vision
ETH Zürich
Fall 2020

Lab Assignment 1 - Camera Calibration

Mokhles Bouzaien

During this lab, camera calibration will be performed by calculating its intrinsic parameters using two different algorithms: Direct Linear Transform and the Gold Standard algorithm. The given image with a 3D calibration object will be used for that. The first step is to select $n = 6$ different points on the image and their corresponding 3D coordinates.

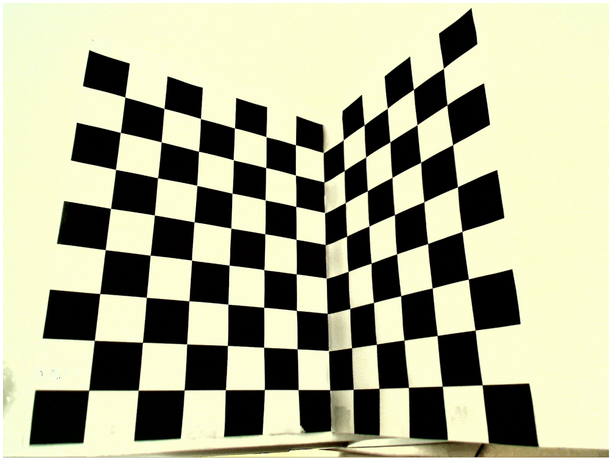


Figure 1: Calibration image

1 Data Normalization (20%)

The aim of this section is to transform input points (x and X) to normalized points ($\hat{x}$ and $\hat{X}$), i.e., the mean of all points is the origin point and the mean distance to the origin is 1. And also return the transformation matrices T and U such that $\hat{x} = Tx$ and $\hat{X} = UX$. To do so, the first step is to select 6 points on the image and compute their centroid $c_{xy} = (c_x, c_y)$ and $c_{XYZ} = (c_X, c_Y, c_Z)$ by averaging all the coordinates and then deducing it from every point. Then, the scales s_{xy} and s_{XYZ} are computed by averaging all the new points' norms. Finally, matrices T^{-1} and U^{-1} are calculated.

$$T^{-1} = \begin{pmatrix} s_{xy} & 0 & c_x \\ 0 & s_{xy} & c_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.0015 & 0 & -1.1469 \\ 0 & 0.0015 & -0.9654 \\ 0 & 0 & 1 \end{pmatrix}^{-1}$$

$$U^{-1} = \begin{pmatrix} s_{XYZ} & 0 & 0 & c_X \\ 0 & s_{XYZ} & 0 & c_Y \\ 0 & 0 & s_{XYZ} & c_Z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.0059 & 0 & 0 & -0.3745 \\ 0 & 0.0059 & 0 & -0.3210 \\ 0 & 0 & 0.0059 & -0.7223 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1}$$

Those tasks are performed in `normalization.m` script.
NB: xy and XYZ indices are respectively used for 2D and 3D points.

2 Direct Linear Transform (40%)

Denormalized Projection Matrix. Given normalized points and the transformation matrices (i.e., T and U), the goal is to solve $A\hat{p} = 0$ to get the normalized projection matrix $\hat{P}$, and then denormalization is performed using the formula $P = T^{-1}\hat{P}U$. To get A, we calculate A_i for each corresponding $\hat{x}_i \leftrightarrow \hat{X}_i$ such that

$$A_i = \begin{pmatrix} \hat{X}_i^T & 0_{1\times 4} & -\hat{x}_i\hat{X}_i^T \\ 0_{1\times 4} & -\hat{X}_i^T & \hat{y}_i\hat{X}_i^T \end{pmatrix} \in \mathbb{R}^{2\times 12}$$

where $\hat{X}_i = (X_i, Y_i, Z_i, 1)^T \in \mathbb{R}^{4\times 1}$ is the homogeneous coordinates vector of the 3D point i. The matrix A is then obtained by vertically stacking all the A_i matrices. Finally, the normalized camera matrix $\hat{P}$ is obtained by reshaping $\hat{p}$ (the last 12×1 column vector of V) into a 3×4 matrix, where $A = USV^T$ is the SVD decomposition.

Projection Matrix Factorization. The second step is to decompose $P = K[R|t] = KR[I| - \tilde{C}] = M[I| - \tilde{C}]$ where K is the calibration matrix containing the internal camera parameters, and R and $\tilde{C}$ containing the external parameters which are related respectively to the camera orientation and position. M is the first 3 columns of P. Then using the QR decomposition, we obtain $M^{-1} = R^{-1}K^{-1}$ where K is triangular and R is orthogonal. Finally, C is such that $PC = 0$ and $t = -RC$.

3 Gold Standard algorithm (40%)

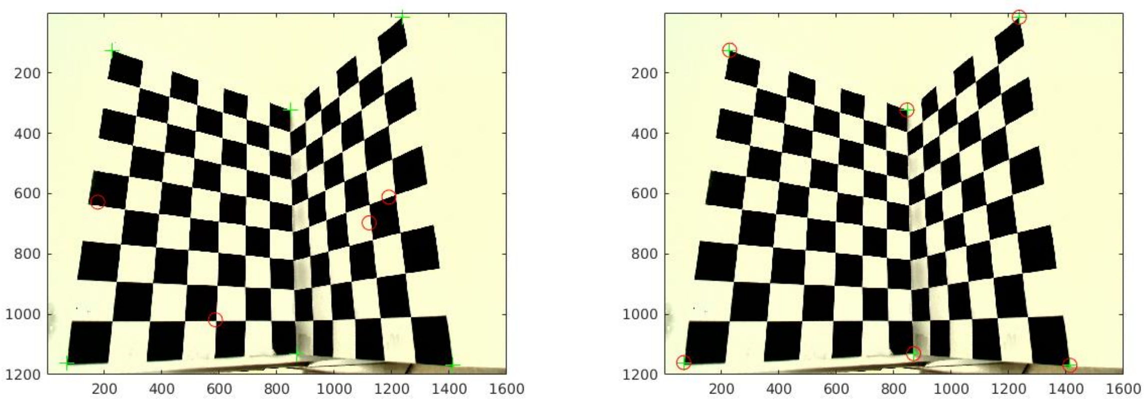
In this part, we will try to find a projection matrix P that minimizes the geometric error

$$e = \frac{1}{n}\sum_i d(\hat{x}_i, \hat{P}\hat{X}_i)^2$$

Steps:

- Initialize by finding a linear solution (normalize & apply DLT)
- Minimize e with respect to $\hat{P}$
- Denormalize $\hat{P}$ i.e., $P = T^{-1}\hat{P}U$

After executing both DLT and GS algorithms, we can see that there is a remarkable difference between their geometric errors, i.e., $e = 10.75$ for DLT and $e = 1.74 \times 10^{-6}$ for GS algorithm. That also can be seen in the figure 2 where the projected points are far from the hand-clicked ones for DLT initialization and almost the same for GS algorithm.



(a) DLT projected points

(b) GS projected points

Figure 2: Difference between DLT and GS algorithms