Report: Camera Calibration

Rishabh Singh

October 4, 2019

Direct Linear Transform

The corners of the square grid were clicked to obtain the image coordinates x_i and world coordinates X_i, where $i = 140$. The given coordinates were first normalised by finding the appropriate T and U matrices. These matrices were obtained by Maximum Likelihood Estimates of mean and covariance obtained from the samples of image and world coordinates respectively. Based on the equation stated in the exercise, the matrix A was computed for every correspondence between the world and the image coordinates. Singular Value Decomposition of A yields a *12x12* matrix V whose last column gives the elements of the normalised camera matrix. The resultant matrix can be de-normalised to obtain camera matrix P.

$$P = \begin{bmatrix} 74.3415 & -3.2343 & -12.0737 & -549.2062 \\ 24.7110 & 40.6195 & 51.5154 & -715.5119 \\ 0.0241 & 0.0385 & -0.0161 & -0.6324 \end{bmatrix}$$

The product of P with the world coordinates gives the re-projected points in the image coordiantes as shown in Fig.1.
The decomposition of the camera matrix P yields calibration matrix K, rotation matrix R and translation vector t, the obtained values of which are shown below:

$$K = 1.0e + 03 * \begin{bmatrix} 1.3423 & 0.0058 & 0.8014 \\ 0 & 1.3366 & 0.5719 \\ 0 & 0 & 0.0010 \end{bmatrix}$$

$$R = \begin{bmatrix} -0.8492 & 0.5281 & -0.0091 \\ -0.1694 & -0.2885 & -0.9424 \\ -0.5003 & -0.7987 & 0.3344 \end{bmatrix}$$

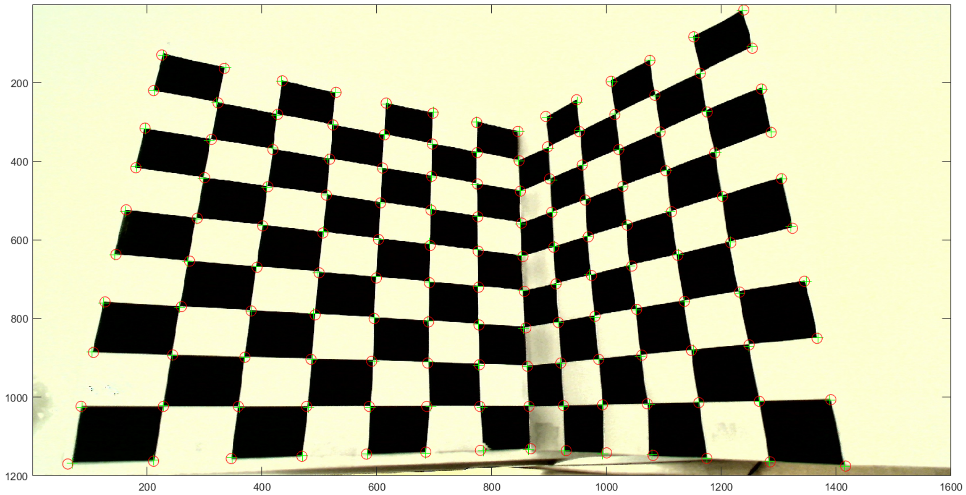


Figure 1: Superimposition of clicked points (green marks) and re-projected points (red circles) obtained by Direct Linear Transform

$$t = \begin{bmatrix} 0.6314 \\ 5.4900 \\ 13.1148 \end{bmatrix}$$

Gold Standard Algorithm

First, an estimate of the camera matrix is obtained using Direct Linear Transform approach as mentioned above. The in-built MATLAB function *fminsearch* is used to search a parameter, say p for which the value of the cost function *J(p)* is minimized. This returns a normalised camera matrix $\tilde{P}$ for which the re-projected error is minimum for the given set of points. The matrix obtained is denormalised again to find the camera matrix P as shown below:

$$P = \begin{bmatrix} 74.3552 & -3.3288 & -11.9756 & -549.7509 \\ 24.5850 & 40.5379 & 51.6421 & -716.0117 \\ 0.0240 & 0.0385 & -0.0160 & -0.6330 \end{bmatrix}$$

P is used for finding the projection of world coordinates in the image plane as shown below: Decomposition of P yields matrices K, R and t as shown below:

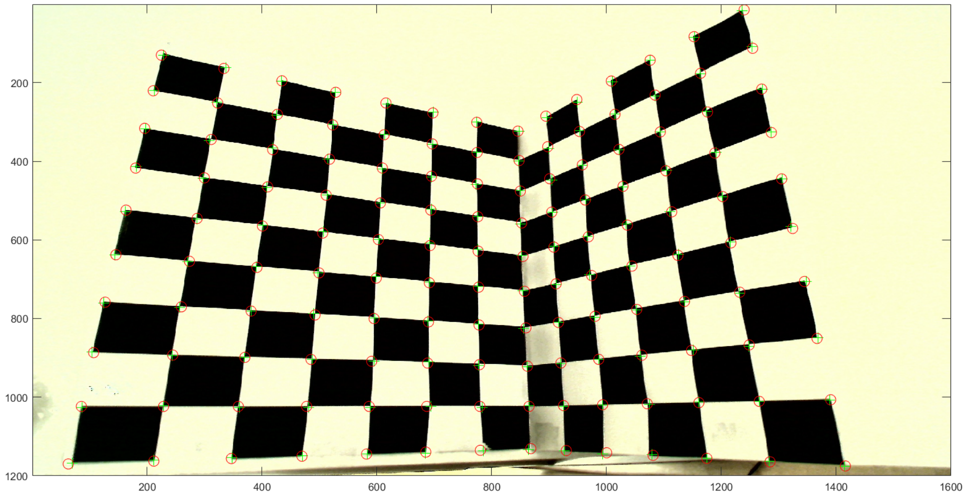


Figure 2: Superimposition of clicked points (green marks) and re-projected points (red circles) obtained by the Gold Standard Algorithm

$$K = 1.0e+03 * \begin{bmatrix} 1.3481 & 0.0053 & 0.8002 \\ 0 & 1.3410 & 0.5720 \\ 0 & 0 & 0.0010 \end{bmatrix}$$

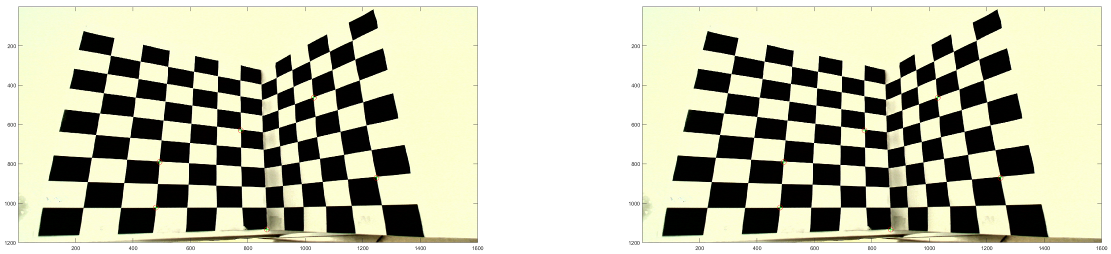
$$R = \begin{bmatrix} -0.8498 & 0.5271 & -0.0092 \\ -0.1681 & -0.2876 & -0.9429 \\ -0.4997 & -0.7997 & 0.3330 \end{bmatrix}$$

$$t = \begin{bmatrix} 0.6462 \\ 5.4892 \\ 13.1629 \end{bmatrix}$$

Discussion

N=6: The performance of both the algorithms was assessed on a relatively smaller sample size, $N = 6$. The results are shown in Fig.3:

Reprojection Error: The reprojection error ($N = 140$) obtained from direct linear transform (=2.6840) is more than that obtained through the gold standard algorithm (=2.6646). The error is more pronounced in the case $N = 6$ where DLT has an error of 0.8230 while GSA has 0.6294. My intuition for the better performance of the latter is that the gold standard algorithm incorporates non-linearity in terms of a distance metric, while DLT tries to find



(a) Direct Linear Transform(DLT) (b) Gold Standard Algorithm (GSA)

Figure 3: DLT has a higher error (0.8230) than GSA (0.6294) for N=6

solution through linear equations. I tried to read more on this hypothesis and figured out that DLT assumes that the camera projection matrix P has 12 degrees of freedom when it has only 6(3 for 3D rotation and 3 for 3D translation). This 6DOF matrix is obtained by approximation, which is not efficiently captured through DLT equations. On the other hand, non linear methods like gold standard algorithm take into account the 6 degrees of freedom of the camera matrix and hence are more accurate.

Running time: For all the points in the grid (i.e. $N = 140$), I tried computing the running time of both the algorithms. While DLT took $0.005159s$ to run, GSA took $0.097387s$, which is roughly $18times$ slower. This overhead might be due to the optimization problem solved by GSA. One can say that if the difference in error is not significant, one can opt for DLT for real time performance.

Normalization: In case where coordinates were not normalized before running DLT or GSA, high error in reprojection were obtained. This high error is due to the difference in scale of image coordinates, noise in the source and target coordinates etc. Thus normalization is essential for numerical stability, accurate estimation and faster solutions.

Why DLT?: Due to its simplicity, low computational cost, and ease of programming.

Why GSA?: More accurate, increase in the computation time but as the calculation only occurs during calibration it is an acceptable loss in performance