Computer Vision

Alessia Paccagnella

September 2019

1 Data Normalization

In order to do this task I:
1) Computed the centroids of the 3D and 2D points that I chose, except for the point corresponding to the origin.
2) Shifted the 3D and 2D points so that the centroid was at the origin.
3) We also wanted the average Euclidean distance from the origin to be $\sqrt{2}$ for the 2D points, and $\sqrt{3}$ for the 3D points.
To do this I calculated the average distance from the points $(0,0)^T$ (for 2D points) and $(0,0,0)^T$ (for 3D points). I used them to write the matrixes T and U.

$$T = \begin{bmatrix} \sqrt{2}/\mathrm{dT} & 0 & -\sqrt{2}/\mathrm{dT}*\mathrm{meansXY}(1) \\ 0 & \sqrt{2}/\mathrm{dT} & -\sqrt{2}/\mathrm{dT}*\mathrm{meansXY}(2) \\ 0 & 0 & 1 \end{bmatrix}$$

Where dT = the average distance from the origin, and
meansXY = $\begin{bmatrix} xcentroid & ycentroid \end{bmatrix}$

$$U = \begin{bmatrix} \sqrt{3}/\mathrm{dU} & 0 & 0 & -\sqrt{3}/\mathrm{dU}*\mathrm{meansXYZ}(1) \\ 0 & \sqrt{3}/\mathrm{dU} & 0 & -\sqrt{3}/\mathrm{dU}*\mathrm{meansXYZ}(2) \\ 0 & 0 & \sqrt{3}/\mathrm{dU} & -\sqrt{3}/\mathrm{dU}*\mathrm{meansXYZ}(3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

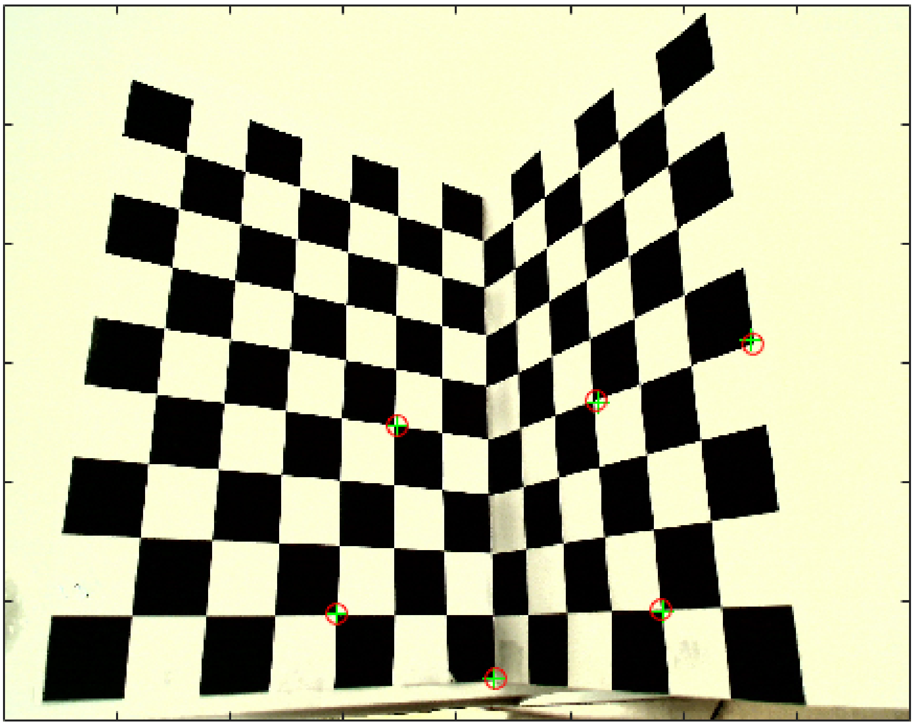
Where dU= the average distance from the origin, and
meansXYZ = $\begin{bmatrix} xcentroid & ycentroid & zcentroid \end{bmatrix}$

After that, I calculated the matrixes of the 2D points normalized and 3D points normalized, that also have a 1 because they are in homogeneous coordinates.

2 Direct Linear Transform

To do this task I computed $\widehat{P}$ normalized with the DLT algorithm. By choosing 6 points, I had 6 corrispondences $\widehat{Xi} \leftrightarrow \widehat{xi}$ so I composed a 12 12 matrix A.

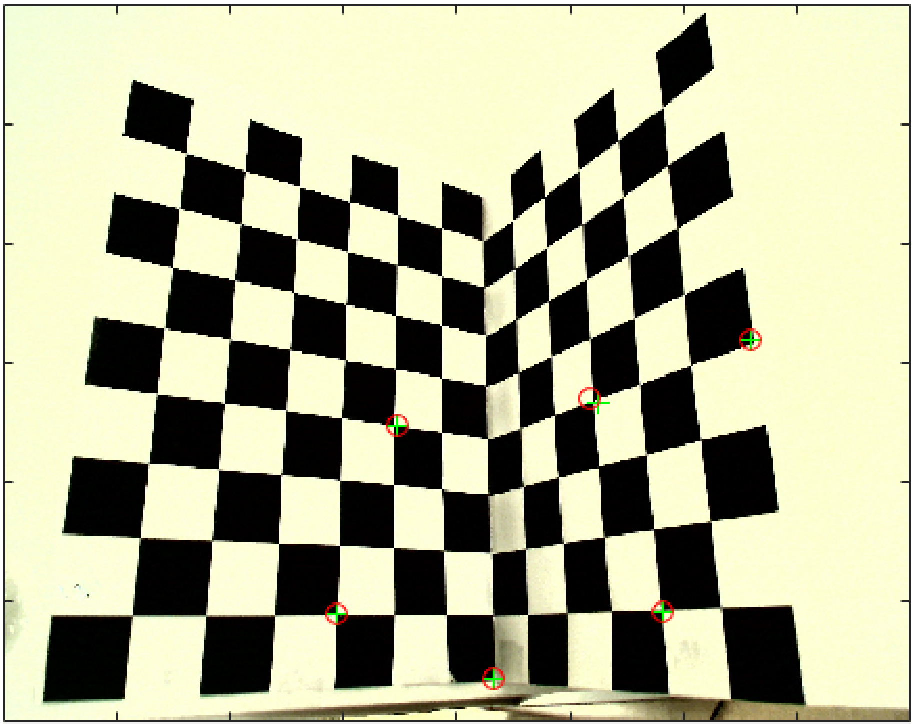
I computed the SVD for A, finding [U,S,V], and found $\widehat{\mathsf{P}}$ normalized which corresponded to the last column of the matrix V.
I wrote the column as a matrix of 3x4.
After that I also computed P denormalized as $P = (\mathsf{T})^{-1}PU$.
I had to decompose P into K, R and t using the QR decomposition. By knowing that $M = KR$ and that M has size 3x3, I used the QR decomposition with the inverse of M, obtaining an ortogonal $(\mathsf{R})^{-1}$ and a triangular$(\mathsf{K})^{-1}$ matrix.
After that I found K and R.
I computed the camera center C such as P*C=0, by using the SVD again.
This let me calculate the translation t=-R*C I calculated the error by error between the computed points and the real points, and I have seen that it's very little (with the points that I took, the average error is 3.6893). When using unnormalized points I didn't notice a worse behaviour.



Chosen and computed points with the dlt algorithm

3 Gold Standard algorithm

To run this algorithm, the points are all normalized and an initial matrix P is calculated. After that, the matrix P is refined by minimizing the error of the reprojection by using the function fminsearch. After that, the matrix P is denormalized and factorized again. At the end I computed the reprojection error again. I obtained another good result with a smaller error than the DLT algorithm (with the points that I took, the average error is 2.5582). When using unnormalized points I did not notice any difference in the behaviour.



Chosen and computed points with the gold standard algorithm