2 Computer Vision – Pierre Beckmann

2.1 Data Normalization

The transfromation matrices are given by:

$$\mathbf{T} = \begin{bmatrix} s_{2D} & 0 & c_x \\ 0 & s_{2D} & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} . \mathbf{U} = \begin{bmatrix} s_{3D} & 0 & 0 & c_X \\ 0 & s_{3D} & 0 & c_Y \\ 0 & 0 & s_{3D} & c_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} .$$

($\mathbf{T}$ for two dimensional image points and $\mathbf{U}$ for three dimensional object points.)
With:

- c_x and c_y are respectively the means of the first and second components of the two dimensional points:
$c_x = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad c_y = \frac{1}{n}\sum_{i=1}^{n} y_i;$
- c_X, c_Y and c_Z are respectively the means of the first, second and third components of the three dimensional points:
$c_X = \frac{1}{n}\sum_{i=1}^{n} X_i, \qquad c_Y = \frac{1}{n}\sum_{i=1}^{n} Y_i \qquad c_Z = \frac{1}{n}\sum_{i=1}^{n} Z_i;$
- $s_{2D} = \dfrac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}[(x_i - c_x)^2 + (y_i - c_y)^2]}}{\sqrt{2}};$
- $s_{3D} = \dfrac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}[(X_i - c_X)^2 + (Y_i - c_Y)^2 + (Z_i - c_Z)^2]}}{\sqrt{3}}$

2.2 Direct Linear Transformation

Using the DLT algorithm we computed the normalized $\hat{\mathrm{P}}$ matrix and proceded by denormalizing it to get the P matrix. Next we computed matrices K and R, the translation t and the error.

Using 6 points (the extreme see Fig.1 in yellow) corners to compute the camera matrix we visualize the reprojected points of all corners on the checkerboard (Fig.1).

Figure 1: **Reprojected Points using the DLT Algorithm.** In yellow circles we have the 6 used points for the camera matrix, in black x's the correct location of the points, in red diamonds the reprojected points.

In order to compare the use of normalized points and unnormalized points we used the DLT algorithm with all the 60 corners of the checkerboard therefore inducing a bigger possible error (Fig.2).

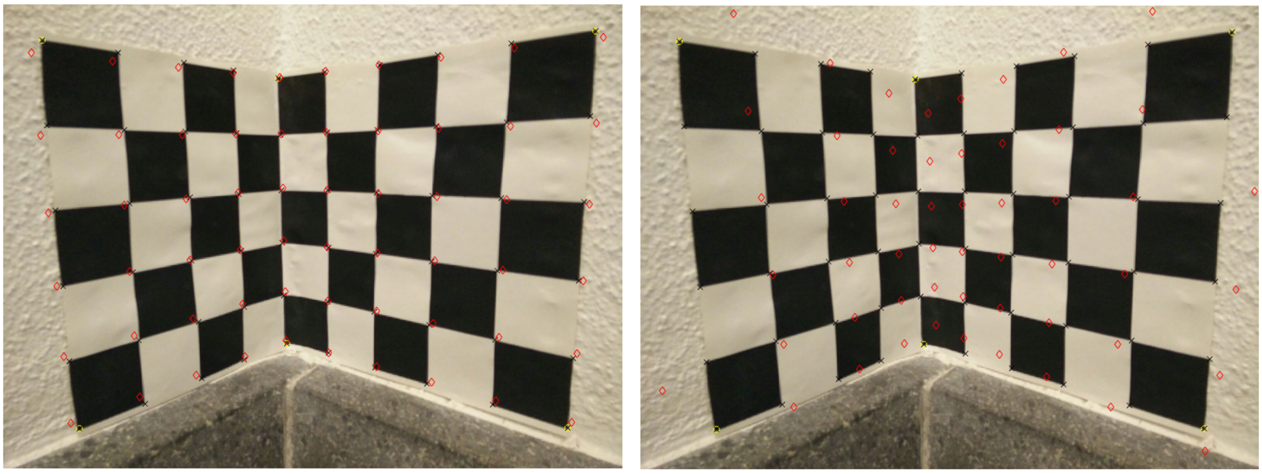

Figure 2: **Using unnormailzed points.** The DLT algorithm was used with all the 60 corners rather than 6 of the checkerboard in both cases.

2.3 Gold Standard Algorithm

In the Gold Standard algorithm we minmize the following error $\sum_i^n d(x_i, \hat{x}_i)^2 + ws^2 + w(\alpha_x - \alpha_y)^2$, s being the *skew* parameter and $\alpha_x = fm_x$ and $\alpha_y = fm_y$ representing the focal length of the camera in terms of pixel dimensions in the x and y direction respectively. The DLT algorithm is used to get an initial camera matrix for the optimization. The camera matrix was denormalized and factorized as in the DLT algorithm.

The obtained camera matrix using the same 6 points as in Fig.1 yields the results seen on Fig.3.

We obtained the following K matrix using the Gold Standard Algorithm:

$$K = \begin{bmatrix} 1270.6 & 0 & 815.1 \\ 0 & 1270.6 & 598.0 \\ 0 & 0 & 1 \end{bmatrix}$$

We identify $\alpha_x = \alpha_y = 1270.6$ and $s = 0$. These results were obtained because of the error optimization we chose.

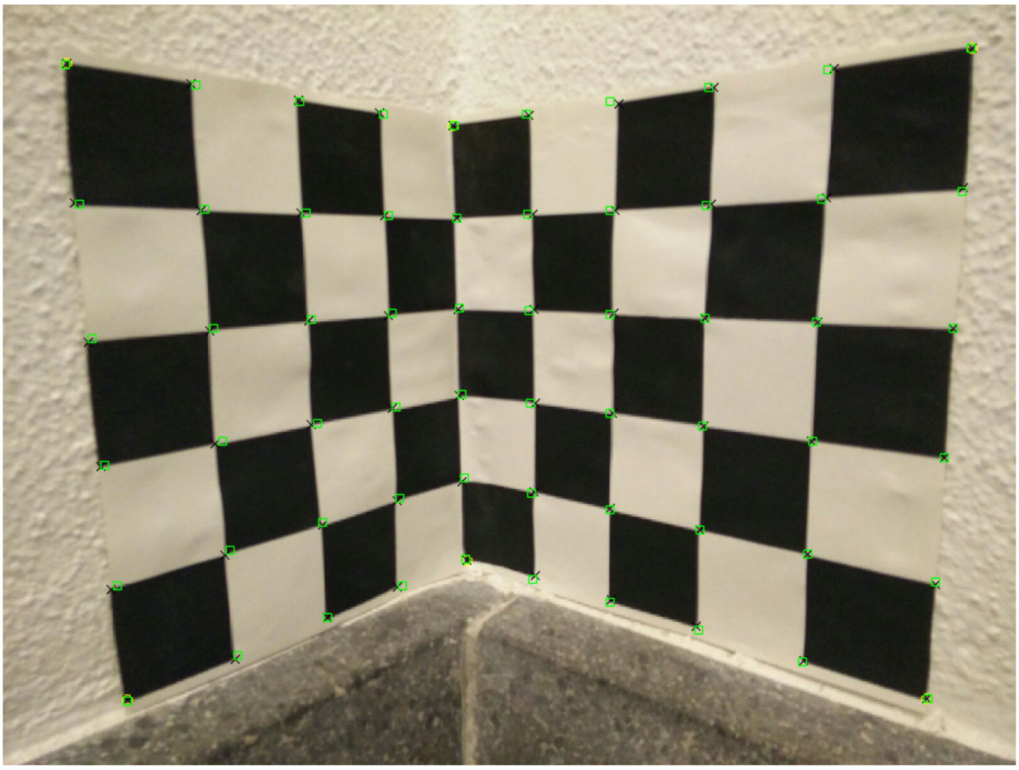

Figure 3: **Reprojected Points using the Gold Standard Algorithm.** In yellow circles we have the 6 used points for the camera matrix, in black x's the correct location of the points, in green squares the reprojected points.

2.4 Bouget's Calibration Toolbox

Using 2 pictures of the same checkerboard (a bug wouldn't let me add more than 2 pictures) We obtained the following results:

$$\alpha_x = 1174.4 \pm 91.5722, \alpha_y = 1295.7 \pm 282.9661, s = 0$$

These are coherent with the results obtained using the Gold Standard Algorithm.

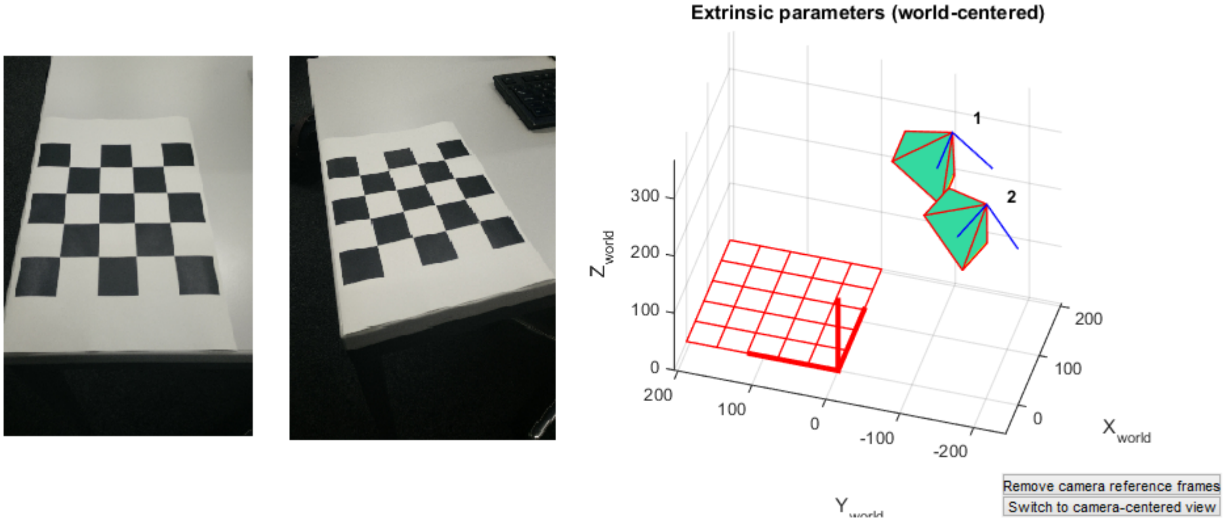

Figure 4: **Bouget Toolbox.** Two Pictures of the same checkerboard of two different perspectives and the obtained extrinsic paramaters (world centered).