Lab 2

Mattia Giacomello
I.D. 1210988

1 Goal

The objective to achieve is the calibration of a camera, so the goal is create a script that from a set of images can extrapolate some features, calculate the intrinsic parameter of the camera used and the coefficients of the lens distortion.

2 Procedure

The image set used is self taken and is used a chessboard pattern with 7x6 corners of edge 3cm. After the loading of the images the program search for the chessboard pattern through the `findChessboardCorners()` method and save a `vector` of 2D coordinate containing the corners of the chessboard founded. Now that the position of the calibration pattern is known for each image the same pattern is created virtually on a `vector` of 3D point placed on the XY plane, so the Z value for each corner is fixed to 0. This is used to obtain the instrinsic matrix and the lens distortion coefficients till the 3 grade for radial through the `calibrateCamera()` method. This method computes also the extrinsic parameters, rotation and translation.
To reduce the reprojection error is possible to use the `cornerSubPix()` method that increase the accuracy of the corners position for each image through a 11x11 pixels search window. This value is chosen after some trial and gives a good result lowering the reprojection error. If the window is larger the error still lowers but it is a very small change from here on. The other parameters are the recommended ones since no relevant changes are observed modifying them. Observing the root mean square error is possible to notice that this refinement is useful since there is a relevant change without and with, it is halved in this case. The image with the worst RMS is blured due to the focus and this make difficult find the corners, but we can observe that thanks the refinement there is a big improvement. The instrinsic matrices are similar, the most significative change is visible in the third column, the offsets. The lens distortion coefficients have a more drastic change, as we can see the tangent coefficient p_1 even changes sign.
Since the image has a low distortion the effect of this remap is not so obvious at first sight. Looking at the edges of the picture is visible that there is a change since the pixels are stretched, thanks to the radial distortion correction. Moreover the straight lines in the original image are a bit bent while in the remapped image the desk's edge and the chessboard's lines result more straight and the toy cars have more natural proportions. In conclusion, the `cornerSubPix()` function is useful to achieve more accuracy after a first fast search and, even if the distortion coefficients are low, is visible an improvement after the undistortion.

	w/o `cornerSubPix()`	w `cornerSubPix()`
Instrinsic Matrix	$\begin{bmatrix} 2223 & 0 & 1499 \\ 0 & 2222 & 970.9 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 2224 & 0 & 1496 \\ 0 & 2222 & 966.1 \\ 0 & 0 & 1 \end{bmatrix}$
Distortion Coefficients $\begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ p_1 \\ p_2 \end{bmatrix}$	$\begin{bmatrix} -0.1537 \\ 0.2102 \\ -0.1691 \\ 0.0002 \\ 0.0007 \end{bmatrix}$	$\begin{bmatrix} -0.1449 \\ 0.1558 \\ -0.0743 \\ -0.0003 \\ 0.0004 \end{bmatrix}$
RMS	0.8319	0.4444
Best Image	0.3056	0.2313
Worst Image	1.456	0.9014

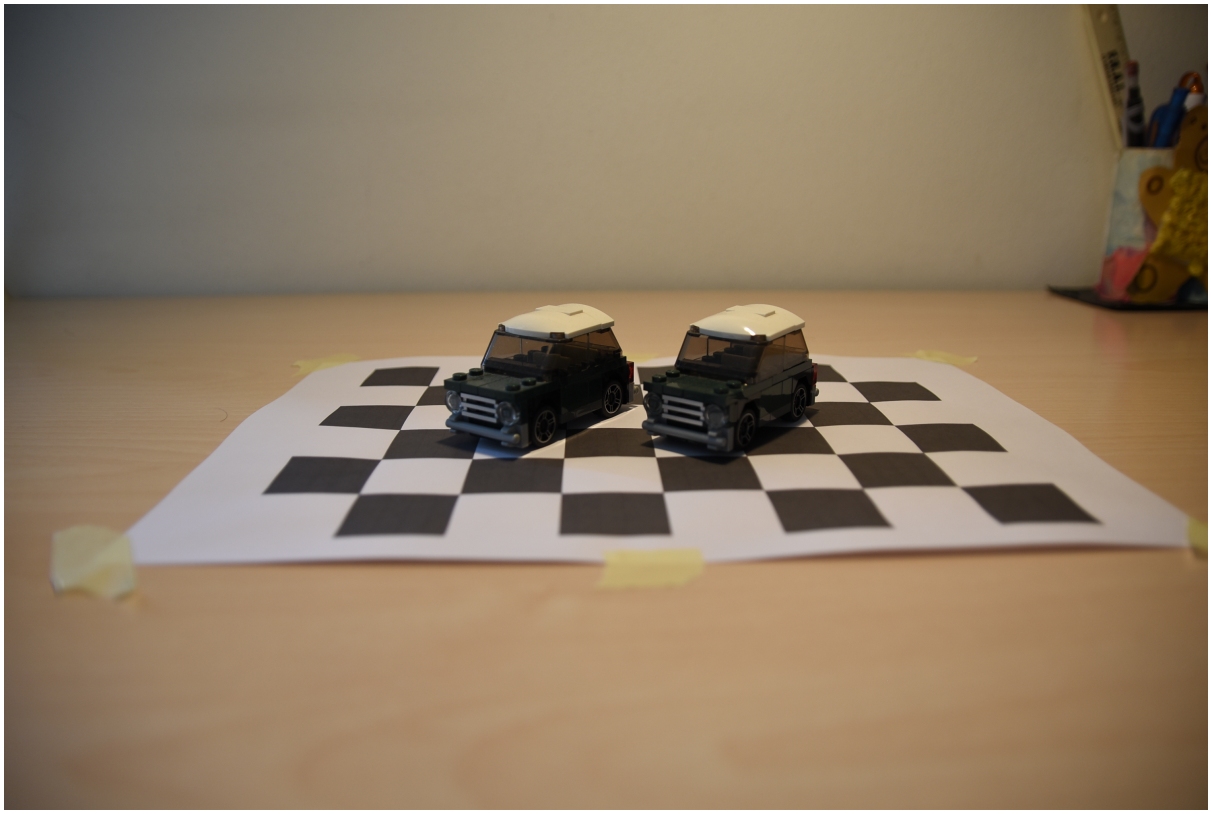

Figure 1: The original distorted image

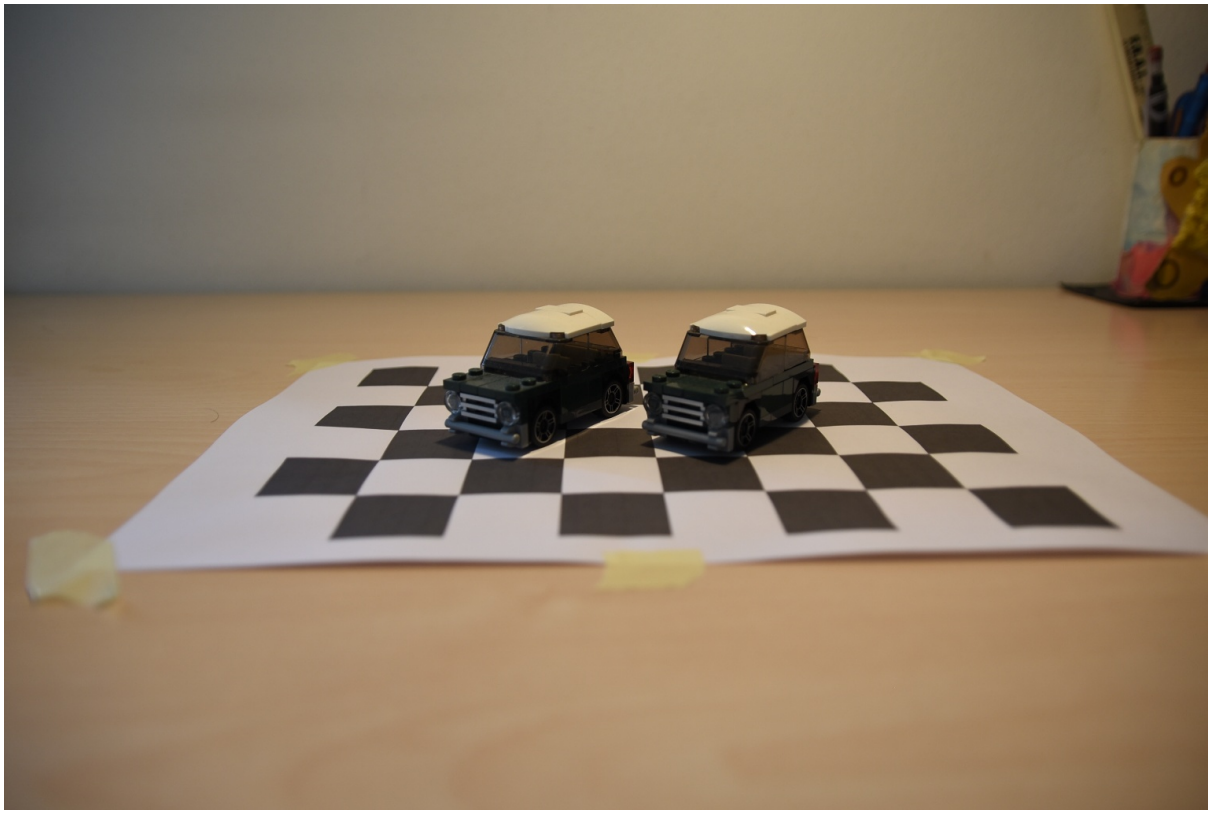

Figure 2: The undistorted image

Figure 3: The worst case in the self taken set