# Report Lab 8

The project is organized in different files:

- the main;
- one header file in which there are the declarations of the functions;
- another cpp files in which the functions are defined.

In the cpp file there are some functions, as for example, the one that allows you to open the images that are in the directory and have the pattern passed as command line arguments when the executable is called and the one that allows to detect the corners by using the function findChessboardCorners.

In the main file, at the beginning, the command line arguments are checked, then the images are opened. Next, a grid of 3D points (for each point the z coordinate is set to 0) is loaded into a vector; the grid size is determined by the values in the CHECKBOARD array (which contains the numbers of square intersections per row and column). So, the function for finding the corners is invocated. It was not necessary to use the function cornerSubPix because the detections were reasonably good. Then the camera is calibrated by the function calibrateCamera. At this point, the mean reprojection error is calculated. The reprojection error is the measure of the distance between the reprojection of a model estimation and its corresponding true projection. For calculating the mean reprojection error, for each image the projectPoints function is invoked with the grid, the corresponding rotation and translation vectors, the intrinsic parameters of the camera, the distortion coefficients and the projectedPoints in which it saves the result as input; then the error is calculated as the L2 NORM between the detected corners of the image and the projectedPoints, divided by the number of detected corners; the error is thus sum with the total error and the detected corners are summed with the total points. Finally, the mean reprojection error is calculated as the total error divided by the total points.

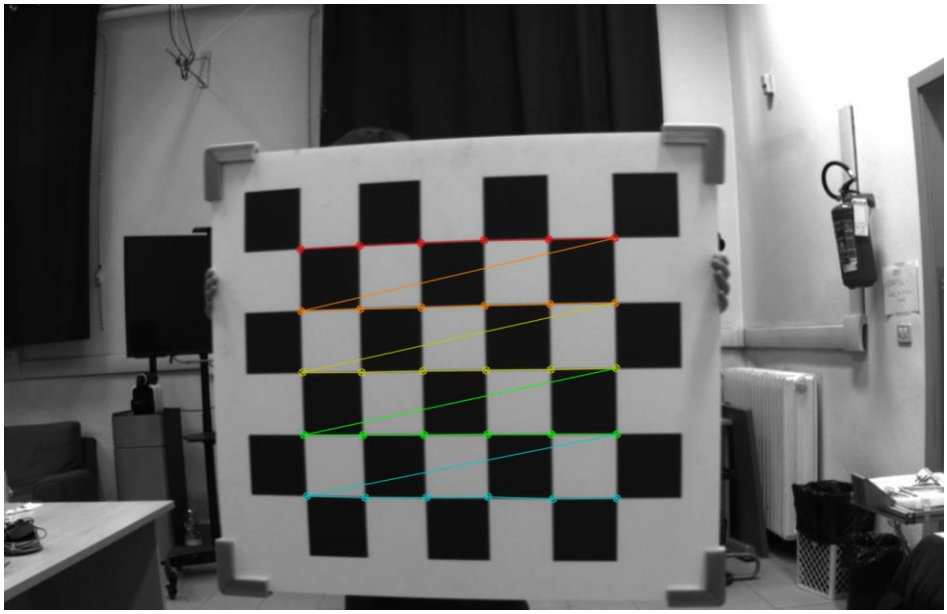
At this point, for each image, is created a final image which is the concatenation of the original one (distorted image) and the corresponding undistorted image. For finding the undistorted images, it is called the initUndistortRectifyMap for getting the rectification maps, then for each image remap function is called for finding the undistorted image. Before the concatenation of the images, these are resized as 1/3 of their dimensions to visualize on the screen the final image. Finally, each final image is shown.

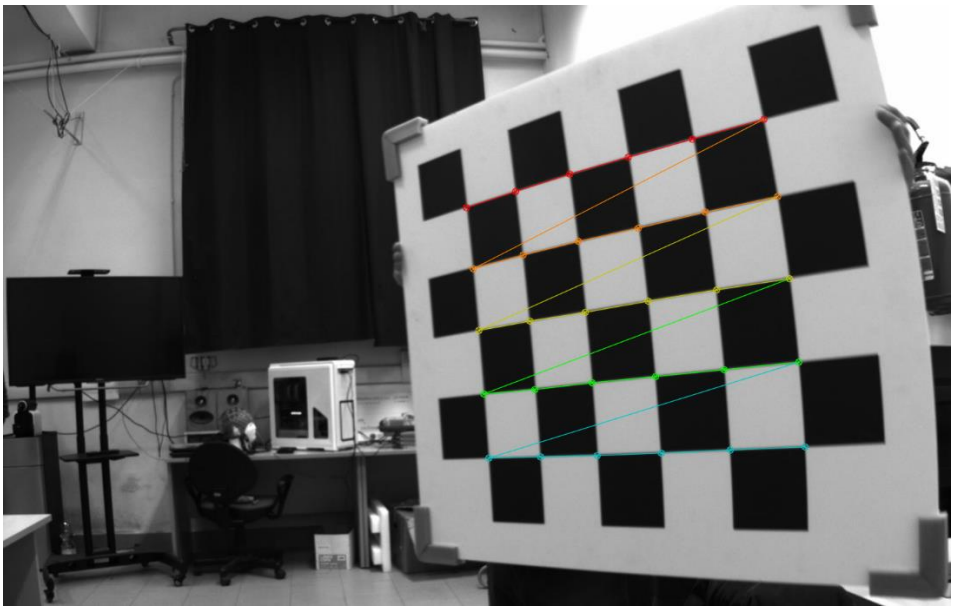
The mean reprojection error calculated with the given images is 0,00435002.

I’ve had some difficulties with understanding how and what parameters pass to the function, but after some time passed in trying and searching, I’ve found the right way.

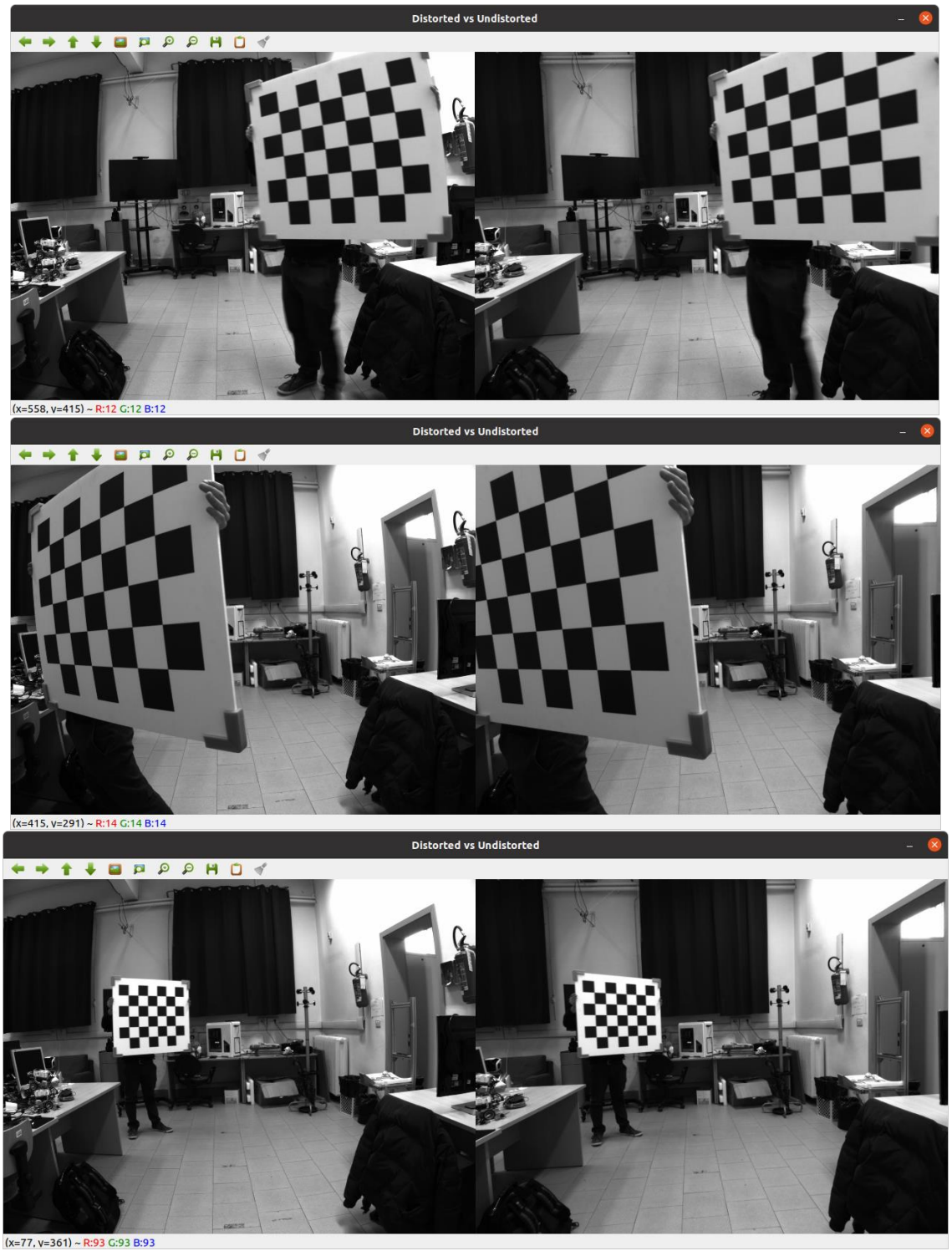
Following some images of:

- examples of detected corners:





- final images:




It is easy to see that the undistorted images have less aperture in angle and in fact things such as the monitors are less visible with respect to the distorted images (they are cut). Also, the objects in the center of the images appear closer in the distorted images. So, to conclude, the undistorted images are “flatter” with respect to the originals.