

Report of Laboratory 2 (Computer Vision)

Oriolo Andrea, 1197672

The following report talks about Camera Calibration

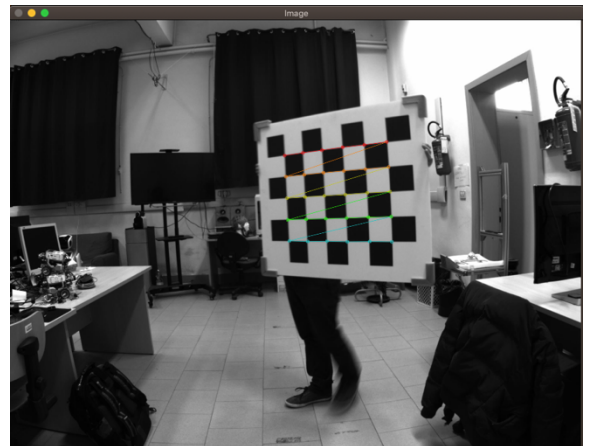
To avoid some problems with the path of the images to be tested, the program requires them as input from the terminal

First, the coordinates of the real points were defined based on the size of the checkerboard, that is 6x5 and cell size 0.11.

Once this was done, all the internal corners of the test images were taken, as shown in the figure

```
for(int i=0; i<images.size(); i++){  
    frame = cv::imread(images[i]);  
  
    success = cv::findChessboardCorners(frame, patternSize, corner_pts, CV_CALIB_CB_ADAPTIVE_THRESH | CV_CALIB_CB_FAST_CHECK |  
        CV_CALIB_CB_NORMALIZE_IMAGE );
```

Once these angles have been saved, it is possible to calibrate the camera through *calibrateCamera* function in order to have the intrinsic and extrinsic parameters of the camera in the form of a cameraMatrix, distortion coefficient, rotation and translation vectors.



To compute the mean of the reprojection error, projected Points were calculated using the *projectPoints* function, which computes projections of 3D points to the image plane given intrinsic and extrinsic camera parameters.

The error of each individual image was calculated between the points of the image itself and the projected points just computed and the minimum and maximum errors were saved to understand which image influenced the best and worst way for calibration.

With all image errors, the mean of the reprojection error was computed, which describes the differences between the detected point and the world point.

Function *initUndistortRectifyMap* computes the joint undistortion and rectification transformation and represents the result in the form of maps for remap.

After this function you need to run the *remap* function that transforms the source image using the specified map computes with previous function function.

The result obtained is the following:

