

COMPUTER ENGINEERING MASTER DEGREE

COMPUTER VISION

Report about Lab2

Raffaele DI NARDO DI MAIO 1204879

May 11, 2020

1 Goal of the experience

The goal of the Lab2 was to calibrate a camera, using a set of images of the same checkerboard, seen from different perspectives, taken with the same camera. After generating the calibration parameters of the used camera, the goal was to find best and worst input images in terms of mean reprojection error. The last thing to do was to use the previously computed calibration parameters to rectify an image of a room, taken by the same camera we calibrated.

2 Code organization

The code of this lab is organized in four main files (for which I provide also the documentation through doxygen):

- **Lab2.cpp** and its header file **Lab2.h**.
They are associated to *main()* and to a function used to print names of the input files.
- **Calibration.cpp** and its header file **Calibration.h**.
These contain the function *calibrate()* that, using other functions, performs checkerboard corner detection on the input images and rectifies the test image.

3 Command line parameters

The program needs to have, as command line arguments, the path of the checkerboard images folder and the path of the test images folder, in this order. For example, the sequence of command line arguments, considering "data/checkerboard_images" as folder of chessboard input images and "data" as folder of test images, is:

`data/checkerboard_images data`

4 Experimental results

In the extraction of corners from the input images, I've tried two approaches to compute them. They give me different parameters and mean reprojection errors (see Table 1 and Table 2). These approaches are:

1. Direct computation through `cv::findChessboardCorners()` function.
2. Improvement of the precision of the previously detected corners using `cv::cornerSubPix()`.

		Approach 1	Approach 2
Intrinsic parameters	α_u	1245.72	1245.11
	α_v	1245.19	1244.22
	u_c	978.874	974.987
	v_c	686.44	684.104
Radial distortion	k_1	-0.3085	-0.308478
	k_2	0.139678	0.138993
	k_3	-0.0367748	-0.036138
Tangential Distortion	p_1	$5.86927 \cdot 10^{-6}$	$-9.79746 \cdot 10^{-5}$
	p_2	$7.81765 \cdot 10^{-6}$	$3.00305 \cdot 10^{-4}$

Table 1: Estimated calibration parameters.

	Approach 1	Approach 2
Average of mean reprojection errors of all images	0.516669	0.294035
Worst image	0027_color.png Error: 1.54774	0045_color.png Error: 1.91905
Best image	0002_color.png Error: 0.226561	0039_color.png Error: 0.0727016

Table 2: Mean reprojection errors with 2 approaches.

The program also has some commented pieces that I've used to compute and store on disk all checkerboard images, with highlighted chessboard corners(e.g. Figure 1 and Figure 2), and also the final image with comparison between the original image and its rectified version (see Figure 3).

For Approach 2, the image in Figure 2 is better than the one in Figure 1, in terms of mean reprojection error. This is because of the angle of chessboard with respect to the space, and because of the smoothness of the image. I noticed also that using too many chessboard images, we are going to overfit the calibration.

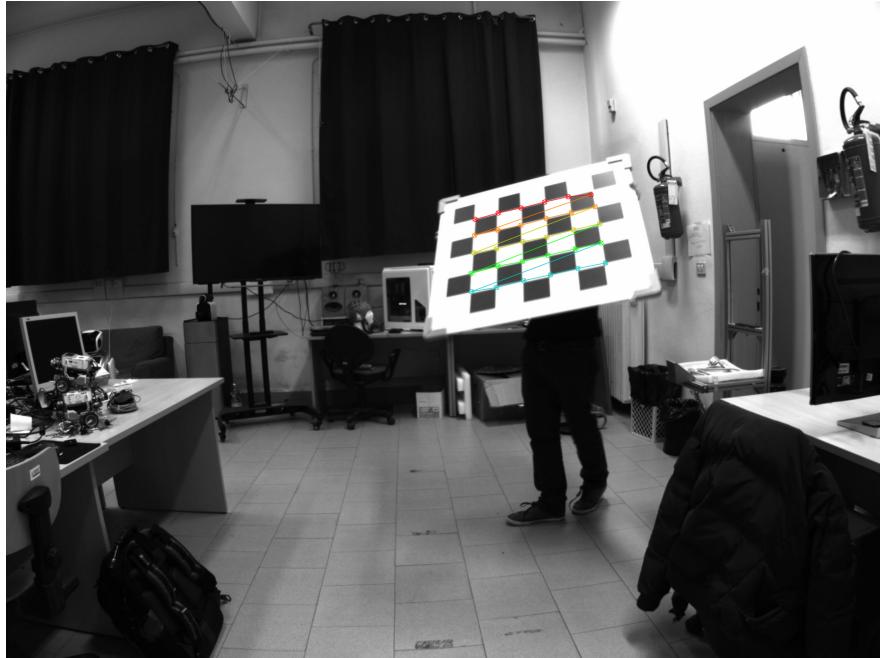


Figure 1: Worst image for Approach 2 in terms of mean reprojection error.

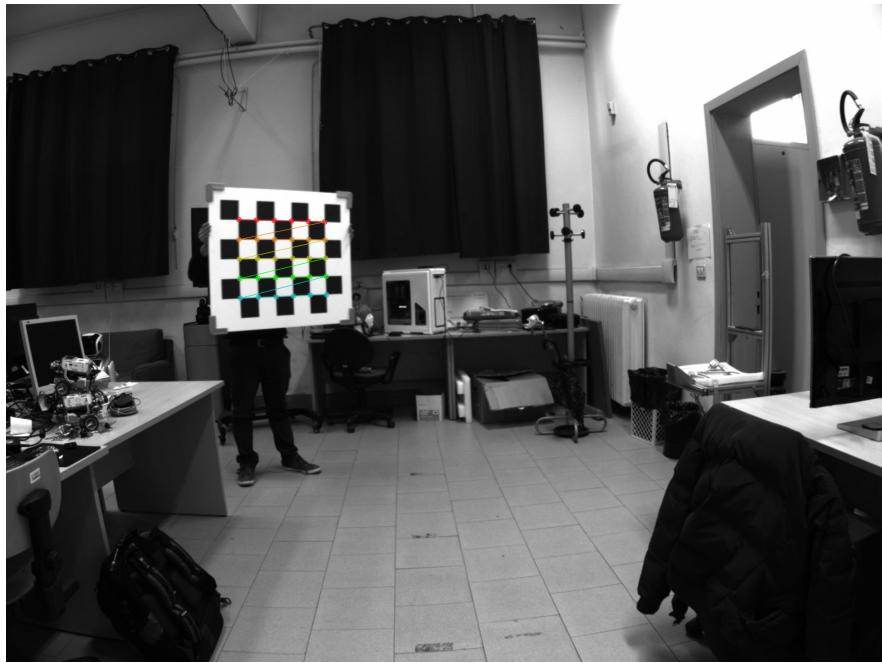


Figure 2: Best image for Approach 2 in terms of mean reprojection error.



Figure 3: Original test image (on the left) vs Rectified image (on the right).