Challenge 1

Student: Xabier Oyanguren Asua
NIU: 1456628 Date: 28/04/2022

1 Objectives

Using the library OpenCV and Python, we will calibrate the camera of a smartphone, obtaining the intrinsic camera matrix and the distortion coefficients related to it. This will allow us undistort images taken with that camera.

2 Data Acquisition

Images were acquired life using the camera of a Samsung Galaxy S III Mini (with a resolution of 640x480 pixels per image), connected using OBS to the computer and controlled directly from the code using the OpenCV library in Python cv2.

A Python script was generated to start taking photos until a chess board is detected, moment in which the edges of the chess squares are automatically detected and plotted on the taken image (all using cv2), as can be seen in Figure 1. This take is shown to the user, who can decide whether to accept the image or not. Then the photo take is resumed. This way, a user pre-defined number of photos are taken (25 in our case).

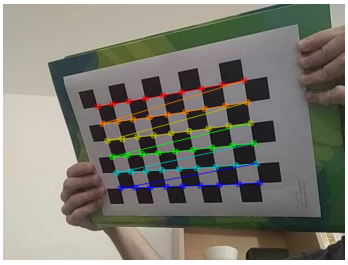


Figure 1: Example capture with the detected corners.

3 Procedure

Using the fact that we know the physical size of the chess squares (thus its physical space coordinates can be known), using the cv2 function `calibrateCamera()` we can compute the intrinsic camera matrix and the distortion coefficient with an iterative method. Once we find them, the script saves the found matrix and coefficients. We can then compute the mean square re-projection error for the chess board world-points into the image plane.

Once we have the camera matrix and the distortion coefficients, we can undistort the images taken by the camera (in particular also the chess board images), employing the `undistort()` function of cv2. Before that however, we can get with `getOptimalNewCameraMatrix()` the region of interest within the undistorted images (such that we can slice out in the undistorted images the rows and columns that contain black pixels that have no information about the image in itself).

4 Results

The obtained intrinsic camera matrix with 25 images is:

$$\begin{pmatrix} 557.6298063 & 0 & 338.31170543 \\ 0. & 551.62747365 & 264.4032827 \\ 0. & 0. & 1. \end{pmatrix}$$

which according to the OpenCV documentation [2] means the focal lengths are fx=557.63 and fy=551.63 and optical centers cx=338.31 and cy=264.40.

We get the distortion coefficients to be:

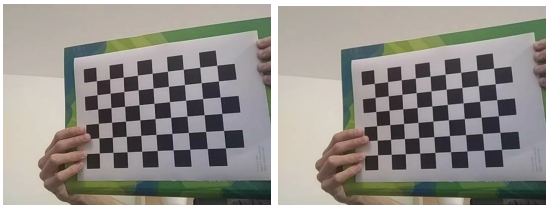
$$[k_1, k_2, p_1, p_2, k_3] =$$

$$[0.03099855, -0.24781441, 0.0113263, 0.01044171, 0.22909384]$$

where k_j stands for the j-th radial distortion coefficient while p_j is the j-th tangential distortion coefficient.

The calibration was possible with a MSE for reprojection given distortion correction of: 0.070109.

You can find an example undistorted chess board image in Figure 2.



(a) Untreated image. (b) Undistorted image.

Figure 2: Result of undistortion using calibrated parameters.

Clearly there is no substantial visual difference, this might be because the camera had already its defects attenuated both by hardware or software.

5 Code

The employed code can be found fully commented in my Github repository:
https://github.com/Oiangu9/_Miscellaneous/blob/main/Challenge1_CV_Code.py

References

[1] The calibration part of the code was written inspired on the tutorial given in the OpenCV documentation: https://docs.opencv.org/3.4/dc/dbb/tutorial_py_calibration.html

[2] https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga3207604e4b1a1758aa66acb6ed5aa65d