COMPUTER VISION - LAB 2

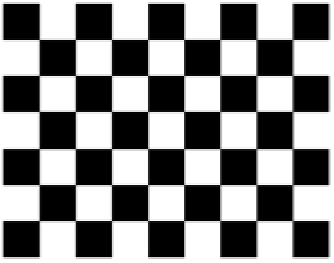
Topics: Camera calibration

Goal: Calibrate a camera.

Prerequisites (if you cannot satisfy them, download the images here: https://drive.google.com/file/d/1wMIDAyndQrBZWW4RreL9Co00hQvJdpmj/view?usp=sharing):

1. Get a camera (it could be a smartphone, webcam, etc.)
2. Get a checkerboard (e.g., download and print the following page: https://drive.google.com/file/d/1FjLl0cktJvATN7ovgtlBw8WGVBrVTIg4/view?usp=sharing) and measure the number of square intersections per row, col and the size of a square edge in meters (the checkerboard should have different numbers of rows and cols).

Example of a checkerboard with 6 rows and 8 cols of square intersections:



3. Place the checkerboard on the desk and do not move it anymore (it should be as flat as possible).
4. Acquire images of the checkerboard with different camera positions. The checkerboard should appear in as much positions, scales, angles as possible.
5. Copy the images on your computer using your preferred protocol (cloud/email/usb)

Write a program that:

1. Loads the checkerboard images.
2. Detects the checkerboard intersections per image. The OpenCV function you are looking for is cv::findChessboardCorners() (optional: consider to use the cv::cornerSubPix() function to refine the corner detections)
3. Calibrates the camera by using the intersections found. The OpenCV function you may want to use is cv::calibrateCamera()
4. Print to output the estimated intrinsic and distortion parameters with correct names (see related lesson slides)
5. Computes the mean *reprojection* error. (how is it calculated?).

6. Choosing among the input images, prints the names of the image for which the calibration performs best and the image for which it performs worst. What is the parameter you are using to perform this choice?
7. Undistorts and rectifies a new image acquired with the same camera (if you used the default one, there should be a picture for this at the link above). The maps to be applied to the image can be found by using cv::initUndistortRectifyMap() (hint: this is not enough to compute the undistortion and rectification. Read the documentation carefully).
8. Compare the result in a split view using the highgui module.

RESULT: