

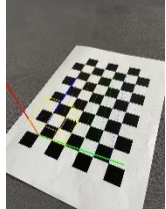
# INFOMCV Assignment 1

## Authors (Group number)

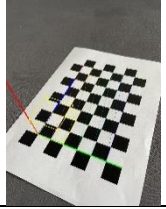
### Calibration

(For the three runs, add (1) intrinsic camera matrix, and (2) test image with cube. Approx 1/2 page.)

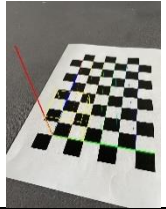
#### Run 1:

$K = \begin{bmatrix} 1.61902958e + 03 & 0 & 8.74749587e + 02 \\ 0 & 1.65045381e + 03 & 8.40772020e + 02 \\ 0 & 0 & 1 \end{bmatrix}$ <p>Image resolution = 1536x2048 Distortion Coefficients: [0.17007429, -0.83407409, -0.07708688, 0.00609084, 0.85301362]</p>	<p>[Test image with cube]</p> 
---	---

#### Run 2:

$K = \begin{bmatrix} 1.60656592e + 03 & 0 & 7.92451180e + 02 \\ 0 & 1.61264267e + 03 & 9.37357740e + 02 \\ 0 & 0 & 1 \end{bmatrix}$ <p>Image resolution = 1536x2048 Distortion Coefficients: [0.16636992, -0.58233604, -0.02508869, 0.00143695, 0.99074407]</p>	<p>[Test image with cube]</p> 
---	--

#### Run 3:

$K = \begin{bmatrix} 1.61698435e + 03 & 0 & 7.67012136e + 02 \\ 0 & 1.62692193e + 03 & 9.21875534e + 02 \\ 0 & 0 & 1 \end{bmatrix}$ <p>Image resolution = 1536x2048 Distortion Coefficients: [0.14088779, -0.28789262, -0.02859055, -0.0056761, 0.13105917]</p>	<p>[Test image with cube]</p> 
---	---

### Discussion

There is a consistent trend of decreasing focal lengths from run 1 to run 3. This indicates that there is a refinement/optimization of the calibration process due to a reduction in the magnification of the image. From run 1 to run 2, there is already a decrease in the focal lengths. This is most likely due to the removal of manually interpolated corner points of the chessboard, which is not as accurate as the automatic detection from the `cv.findChessboardCorners()` function. Run 3 indicates further reduction in focal length, indicating continued optimization of the calibration.

The principal points (Cx, Cy) stay mainly within the same range across the three runs. However, there is quite a large discrepancy in the number of images across runs 1,2 and 3. The range remaining consistent from runs 2 to 3 might indicate that there is a stabilization or convergence towards the center of the image, albeit with low confidence.

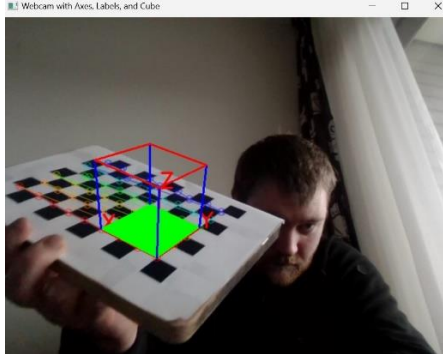
The distortion coefficients reduce across the three runs. This reflects consistent optimization from run 1 to run 3 for improved radial and tangential distortion correction.

Overall, the reduction in focal length and distortion coefficients is consistent across the three runs which indicates an optimization in the camera calibration process.

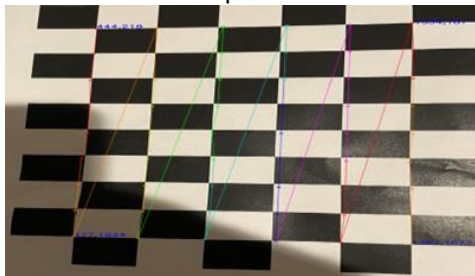
## Choice tasks

The choice tasks we implemented were:

1. Real-time performance with webcam in online phase – To achieve this, we started webcam capture and created a while loop with the code for finding chessboard corners, drawing chessboard corners, calibrating the camera and rendering the 3D axes with labels as well as the cube. The image was overlayed with the webcam video capture.

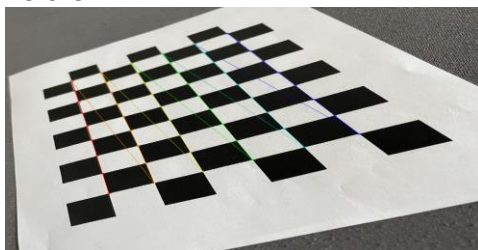


2. The localization of the interpolated points was made more accurate by using the `cv.cornerSubPix()` function. The `winSize` parameters were adapted to improve the accuracy of the corner interpolation. The effect of this function is:



3. To implement the iterative detection, we used a loop structure to get all photos from folder and detect them automatically or manually, before detection, we calculated the value of blur score and brightness of each image and set limit of them to ensure the low-quality pictures are rejected.
4. To enhance the input to reduce the number of input images that are not correctly processed by `findChessboardCorners`, we used `GaussianBlur()` function from OpenCV which can make the picture look smoother and reduce noise, so that make the result more accurate:

Before:



After:

