INFOMCV Assignment 2
Authors: Sotiris Zenios, Andreas Alexandrou (Group 23)

Summary of the 3 Runs:

Run 1: Matrix:

```
[[1.38030699e+03 0.00000000e+00 4.47833769e+02]
 [0.00000000e+00 1.36745528e+03 3.49253125e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

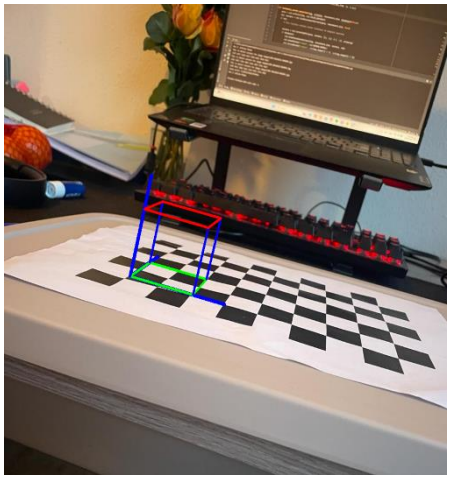

Figure 1: Run 1 Online

Image resolution: 960 x 1280

Total error: 0.44616591107313314

Run 2: Matrix:

```
[[949.22873727   0.         466.93731776]
 [  0.         947.38505019 637.22928655]
 [  0.           0.           1.        ]]
```

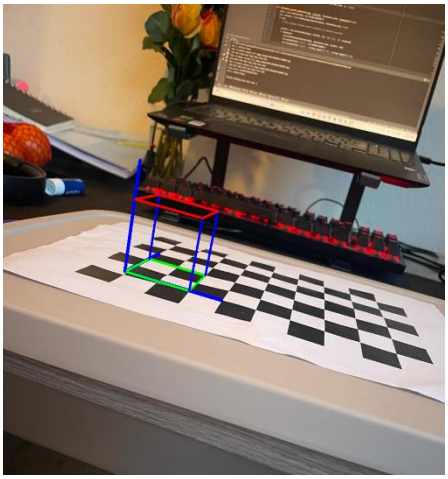

Figure 2: Run 2 Online

Image resolution: 960 x 1280

Total error: 0.06750397280423871

Run 3: Matrix:

```
[[968.28441386   0.         459.50502215]
 [  0.         965.34353058 617.11962889]
 [  0.           0.           1.        ]]
```

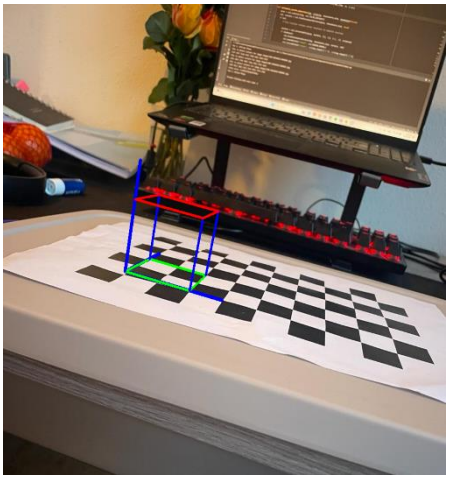

Figure 3: Run 3 Online 1

Image resolution: 960 x 1280

Total error: 0.06797159115388578

Changes in the intrinsics matrix:

The intrinsic matrices obtained from the three calibration runs showed notable differences, primarily reflected in the focal length values (fx and fy) and the coordinates of the optical center (cx and cy). In Run 1 (automatically provided corners, manually provided corners), the focal lengths are substantially higher than in Run 2, suggesting a marginally narrower field of view or closer average positioning relative to the calibration pattern. Run 2, ten images with automatically detected corners, shows a modelate decrease in focal lengths, indicating a broader field of view or a further average distance from the calibration pattern. Run 3, five images with automatically detected corners from Run 2's dataset, resulting in slightly higher focal lengths than Run 2, suggesting a slightly tighter field of view or closer proximity to the calibration object. The optical center (cx and cy) in Run 1 is located more towards the upper left of the image frame, whereas in Run 2, Run 3, cx is similar, but cy is much higher, indicating a lower positioning of the calibration pattern in the image frame or a different camera tilt. These differences in focal lengths and optical center coordinates across runs highlight how different selections of images can influence the calibration process, reflecting the diverse conditions under which the images were taken and the characteristics of the camera lens and sensor.

Given the total calibration errors we can clearly see that in the first run there was a significantly higher error than the rest of the runs, due to the fact that the training dataset included images that we had to manually provide corner points. Run 2 achieves the lowest error, indicating the most accurate calibration among the three. Run 3, shows a slightly higher error than Run 2 because a more selective dataset was used.

Analysis of choices:

Real-time performance with webcam in online phase:

For the real-time performance, 5 images from the webcam were used in Run 3 to calibrate the camera.Then the function live_camera was used, where each frame of the live video was captured and then passed to the function online_phase_live where the corners were detected by the cv2.findChessboardCorners. cv2.findChessboardCorners was used to get the rotation and translation matrix to draw the cube and axes on each frame from the live video.

Rejection of low quality input images in offline phase:

Similarly to Run 2,in RunQualityDetection() the camera was calibrated using 10 images. A threshold of 0.075 was selected in order to find the images with a lower error than that value.The error was calculated by :

```
imgpoints2, _ = cv2.projectPoints(objp, rvec, tvec, mtx, dist)
error = cv2.norm(imgp, imgpoints2, cv2.NORM_L2) / len(imgpoints2)
```

The reprojection error is calculated by projecting the 3D object points (objpoints) into the 2D image plane using the estimated camera parameters and comparing these projected points (imgpoints2) with the actual detected 2D image points (imgpoints). The difference between these sets of points represents the reprojection error, indicating how accurately the camera model predicts the observed image points.The function informs the user which images have a lower error than the threshold and then recallibrates the camera without those images.The function also gives the difference in the mean error between all 10 images and the accepted images.

3D plot with the locations of the camera relative to the chessboard:

In RunCameraLocation() ,after calibrating the camera with 10 images,using a 3D plotting tool, we marked the chessboard's location at the origin of a 3D coordinate system and plotted each camera's position as derived from the negated translation vectors.