Anna Dollbo 1843672
Flavius Marc 0862037

Report Assignment 1: Camera Geometric Calibration

Run 1

Images used: all 20 images, containing both zoomed in and zoomed out images

Intrinsic camera matrix K:

9.9291393685510593e+02	0	6.4287814586092554e+02
0	9.9553258469310686e+02	3.6528194027188732e+02
0	0	1

Cube drawing:

The offline calibration has been made using a dataset of images and then the cube is projected onto the images in the same dataset during the online phase. This can be seen in Image 1.

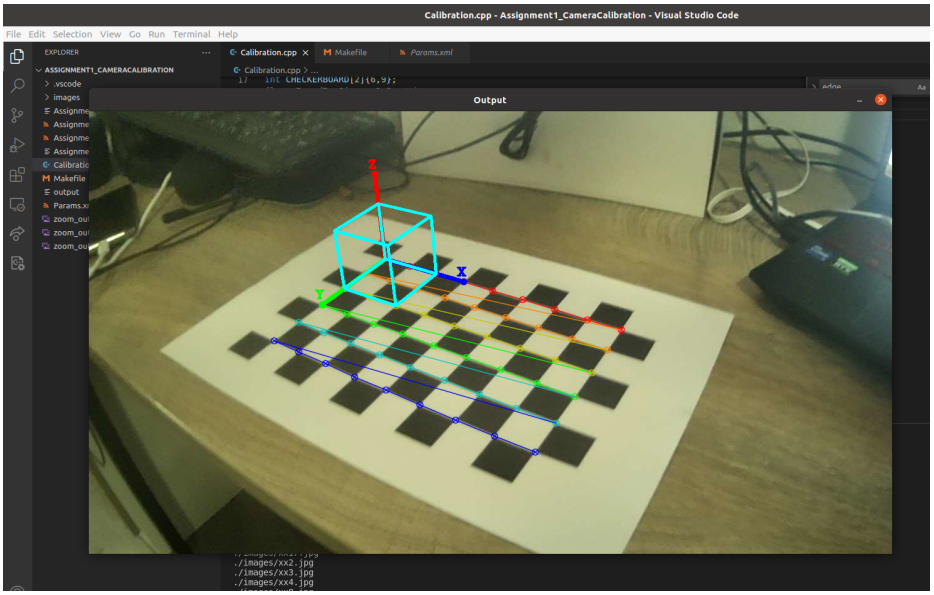

Image 1. Cube drawing

Run 2

Images used: all 20 images, preprocessed with median filtering

Intrinsic camera matrix K:

9.9291182592633311e+02	0	6.4287952389373822e+02
0	9.9553108021096386e+02	3.6528469170713089e+02
0	0	1

Choice task: image preprocessing with median filtering

Median filtering is an image processing method that reduces noise from an image while still preserving edges. For the choice task, the median filter was applied to all images before going through the iteration process of discarding images that contribute to high reprojection error. The median filter

was implemented with the OpenCV function `medianBlur()`, and the difference in the corner detection as obtained through function `findChessboardCorners()` with processing and without process was examined by plotting the edges as circles. This can be seen in Image 2, where green circles represent the corners while using median filtering, and the red circles represent the corners without any image pre-processing.

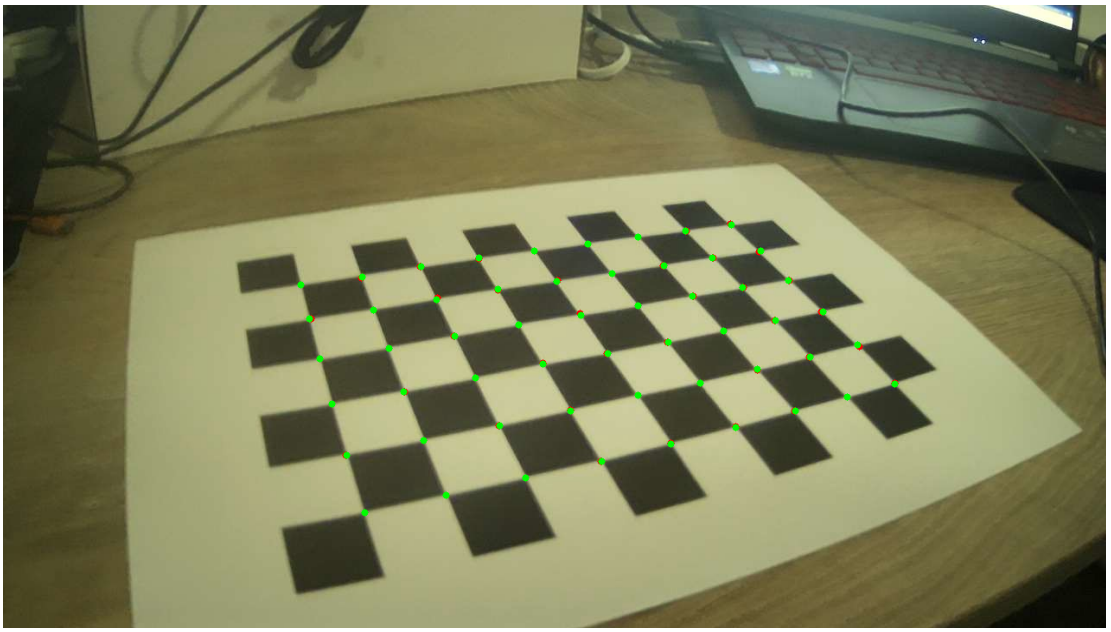

Image 2. corner detection with median filtering (green) and without (red)

Run 3

Images used: 17 images that are zoomed in, thus excluding the zoomed out images

Intrinsic camera matrix K:

9.8578755536532526e+02	0	6.4200479442641642e+02
0	9.8950159544141707e+02	3.7227045502885841e+02
0	0	1

Choice task: live performance with webcam

For the second choice task, live performance of cube drawing using a webcam was selected. This task has a similar procedure to cube drawing, where the 3D world coordinates of image and the box are first initialized. The function `solvePnP()` is then used to obtain the extrinsic parameters, which are the rotation and translation vectors. This is done by passing the intrinsic camera matrix that was found in the offline camera calibration phase (using image dataset) to the function. The function `projectPoints()` is then used to obtain the 2D points from the 3D points in order to draw them on the image plane. Diverse functions are then used to project the cube to the webcam image plane or onto an image of the checkerboard.

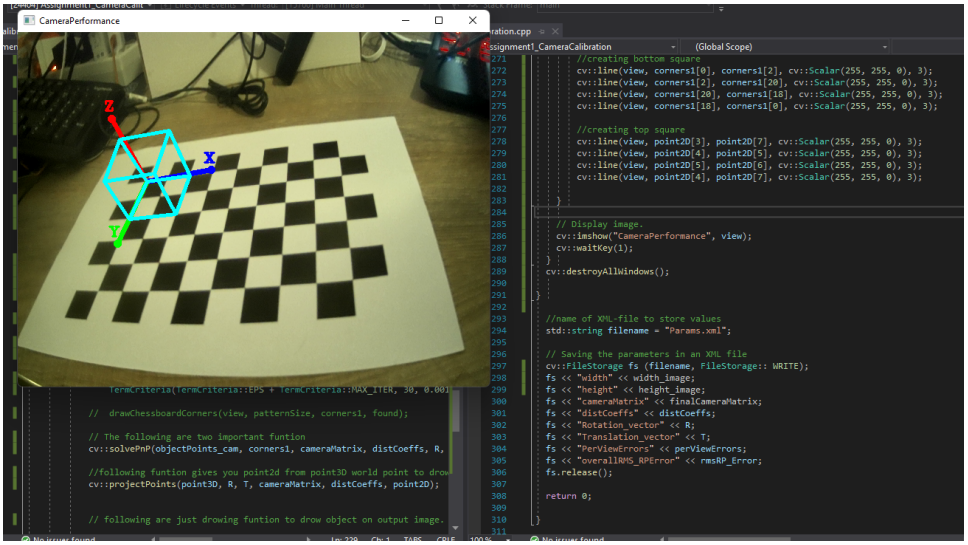

Image 3: screenshot of live webcam performance

Intrinsic camera matrix K explanation

The intrinsic camera matrix K transforms 3D world camera coordinates to 2D homogeneous image coordinates. The values at position [0][0] and [1][1] in the K matrix correspond to the focal length fx and fy. This is the distance between the pinhole and the image plane. The values at position [0][2] and [1][2] tell us the offsets of the principal point, which is the point where the principal axis (from the pinhole) intersects the image plane. The principal point offset values tell us the location of the principal point relative to the image's origin, which in this case is the top-left corner. The value of parameter s at position [0][1] in the K matrix is 0, indicating no distortion in the projected image. The other values of zero in the matrix correspond to the identity matrix.

Discussion

The differences between the three runs are most stark when comparing Run 1 or Run 2 to Run 3. Run 1 and 2 uses the same set of images, but run 2 uses an image preprocessing method of median filtering. This gives a very slight change in intrinsic camera matrix values, which is only visible at the fifth decimal point. The difference between the two runs can also be seen in overall RMS reprojection error obtained at the end of the offline phase (Run 1 error: 0.206856, Run 2 error: 0.206842). The difference is minimal, and it thus does not appear like the median filtering contributes to a better calibration for this image set. However, we cannot exclude that it could have a larger effect for a different dataset of checkerboards, taken in different light conditions and with a different camera.

The dataset contains 20 images in total, where 17 of the images are zoomed in so that the checkerboard covers most of the screen, and three images are more zoomed out. In Run 3 the three zoomed out images were excluded from the dataset. This led to a different intrinsic camera matrix and also to a larger RMS reprojection error (Run 3 error: 0.232734) than the previous two runs. It thus appears like the zoomed out images contribute to a better calibration than the zoomed in images.

Using `CALIB_USE_INTRINSIC_GUESS`, the values of fx and fy, cx and cy are estimated based on an initialisation of the intrinsic values, and cx and cy are not calculated from the image resolution. This means the values of fx and fy will not be equal, which can be attributed to the mechanical nature of the cameras and so will never be equal in practice.