Practice 1.1
Miguel Angel Montoya

Camera: Pixel 1 XL
Factory specs: Pixel size 1.55µm, f 4.7mm, 4048x3036
Chessboard data: 7x9 squares, measuring 20mmx20mm

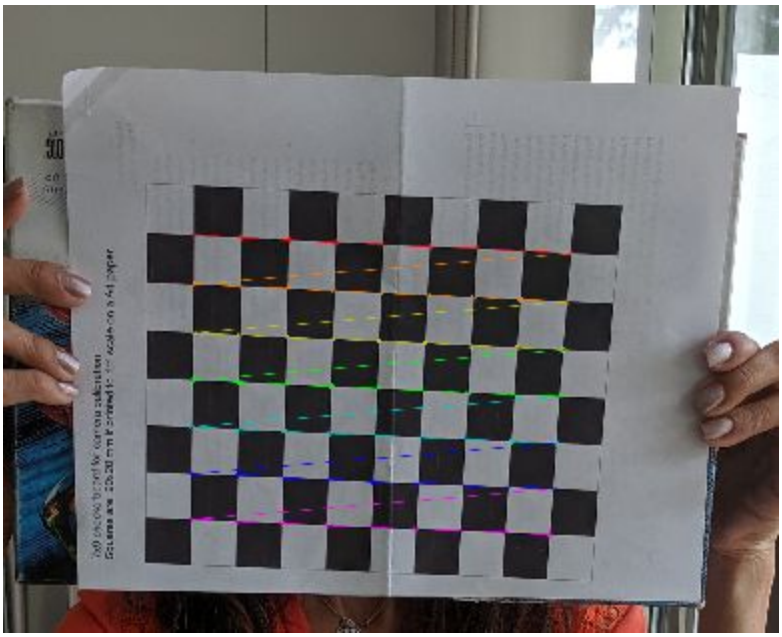
Procedure

1. 18 photos were taken of the chessboard from different angles and altitudes a meter of distance from the camera (Inserted two photos as example)


2. Defined the square object points coordinates

The object pints are the relative coordinates of the corners of the chessboard assuming Z doesn't change. E.g: (0, 0, Z), (1, 0, Z), (1, 1, Z) ... (9, 7, Z)

3. Used openCV *findChessboardCorners* for each picture. Obtained values were stored in a list only if the corners were found in the image.



4. Used openCV *calibrateCamera* with the found corners and the previously defined object points to obtain the camera matrix

```
~/Documents/Semester9_Code/P1 python calibrate.py
[[3.20169597e+03 0.00000000e+00 1.46581716e+03]
 [0.00000000e+00 3.22880409e+03 1.95383573e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

For starters, we can assume the matrix it's correct (Well, maybe just a bit off), because both c, if doubled, produce a supposed image resolution (3907x2931) similar to the factory one (4048x3036), just a difference of 145 and 105 pixels.

5. Adjusted obtained values

As you can see, f_x and f_y are not equal, that would mean that the sensor pixels aren't squares, but rectangles instead. That would mean the the pixels aren't really 1.55µm x 1.55µm.

The obtained values were converted to mm by multiplying each f by the pixel size times 1000 (This to convert µm to mm)

```
p = 0.00000155
fx = fx*p*1000
fy = fy*p*1000
```

The new f_x = 4.962628748750081mm and f_y = 5.004646336884995. Which is actually just ~+0.26 and ~+0.3 off the supposed factory value.

6. Obtained pixel coordinates of a corner of the chessboard via GIMP.

Analyzing one of the taken pictures in GIMP, the coordinates of a corner of a square were obtained: a(730, 990) and b(737, 1094)

7. Used Thales theorem and Pythagoras theorem to calculate distance between the two corners

$a_y' = a_y * z / f_y$
$b_y' = b_y * z / f_y$
$a_x' = a_x * z / f_x$
$b_x' = b_x * z / f_x$
$d_y = b_y' - a_y'$
$d_x = b_x' - a_x'$
$h = \text{sqrt}(d_x^2 + d_y^2)$
h = 20.82850627837143mm
Obtained value it's just 0.8mm off. This difference can be the result of not properly taking the picture 1 meter from the camera