camera_calibratio

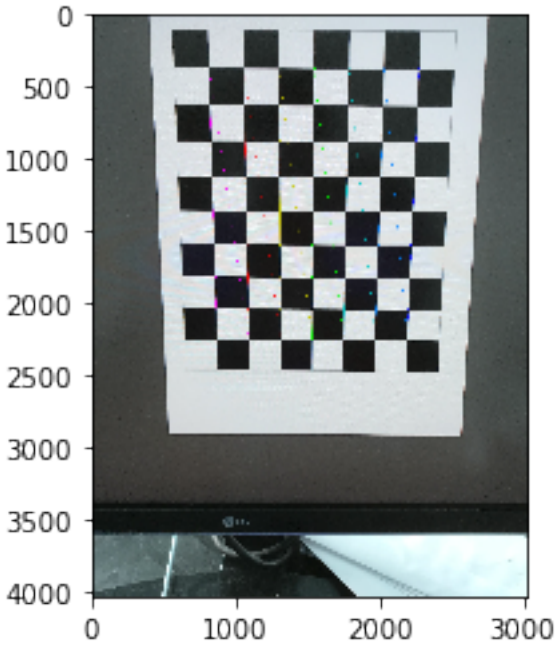
October 18, 2021

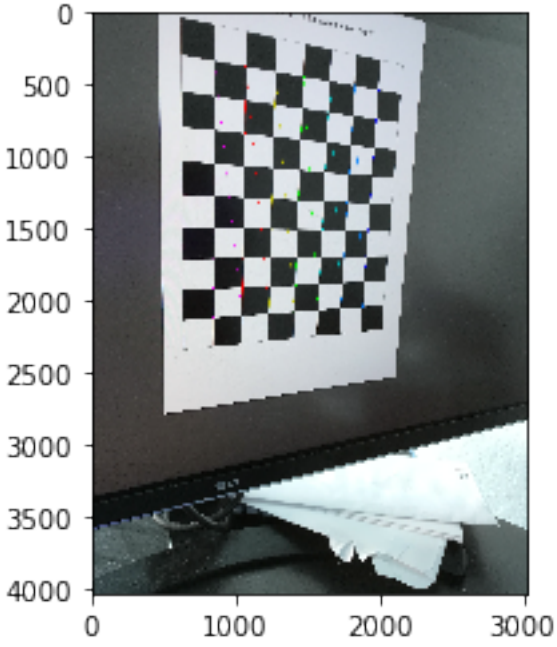
[ ]:
```
import numpy as np
import cv2 as cv
import glob
import matplotlib.pyplot as plt
```

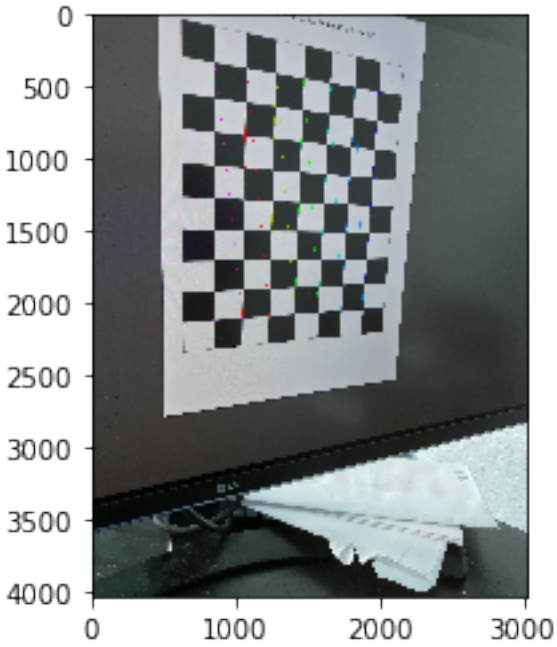
1 Iphone 7 plus camera calibration

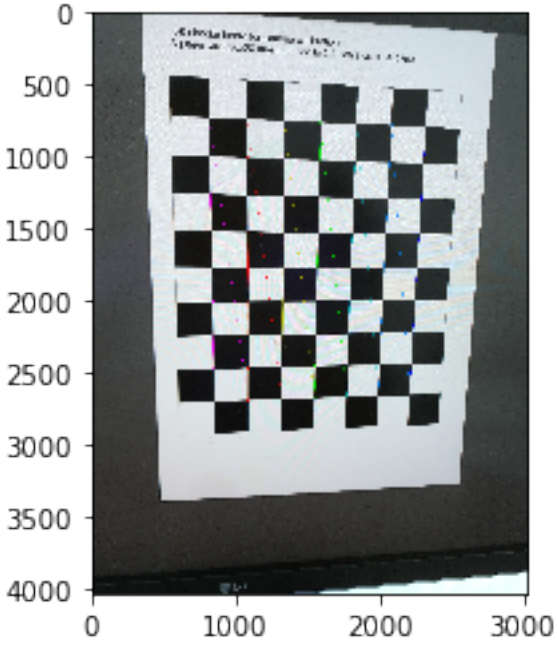
1.1 Reading checkboard images and calculating corners

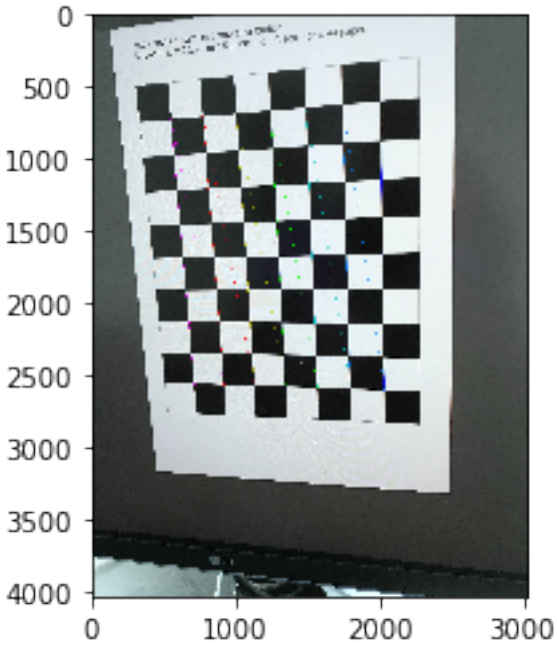
[ ]:
```
checker_h = 9 #Horizontal lines
checker_v = 7 #Vertical lines
# termination criteria
criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 30, 0.001)
# prepare object points, like (0,0,0), (1,0,0), (2,0,0) ....,(6,5,0)
objp = np.zeros((checker_v*checker_h, 3), np.float32)
objp[:, :2] = np.mgrid[0:checker_h, 0:checker_v].T.reshape(-1, 2)
# Arrays to store object points and image points from all the images.
objpoints = []  # 3d point in real world space
imgpoints = []  # 2d points in image plane.
images = glob.glob('./images/*.jpg')
for fname in images:
    img = cv.imread(fname)
    gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
    # Find the chess board corners
    ret, corners = cv.findChessboardCorners(gray, (checker_h, checker_v), None)
    # If found, add object points, image points (after refining them)
    if ret == True:
        objpoints.append(objp)
        corners2 = cv.cornerSubPix(gray, corners, (11, 11), (-1, -1), criteria)
        imgpoints.append(corners)
        # Draw and display the corners
        cv.drawChessboardCorners(img, (checker_h, checker_v), corners2, ret)
        # cv.imshow('img', img)
        plt.imshow(img)
        plt.show()
        # cv.waitKey(500)
cv.destroyAllWindows()
```

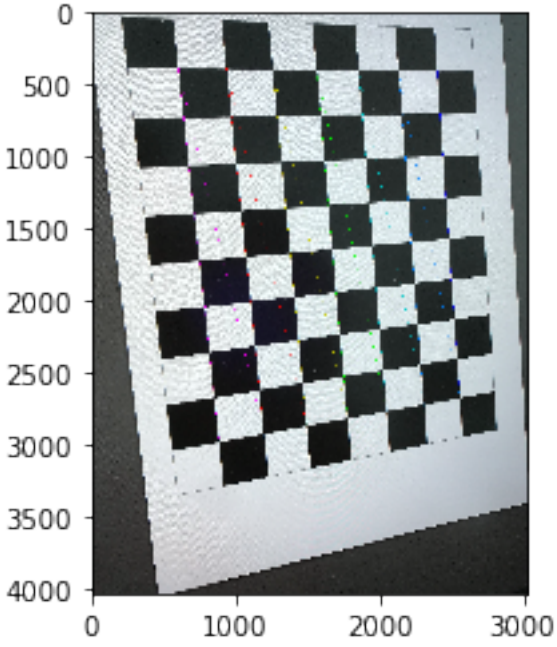

0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000

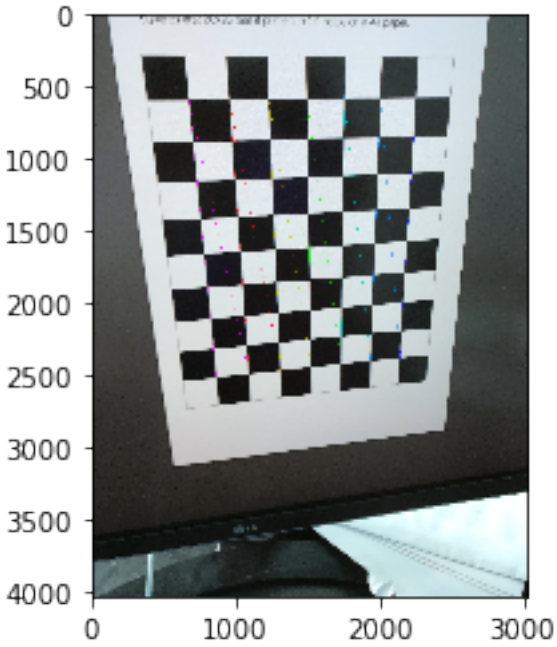

0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000

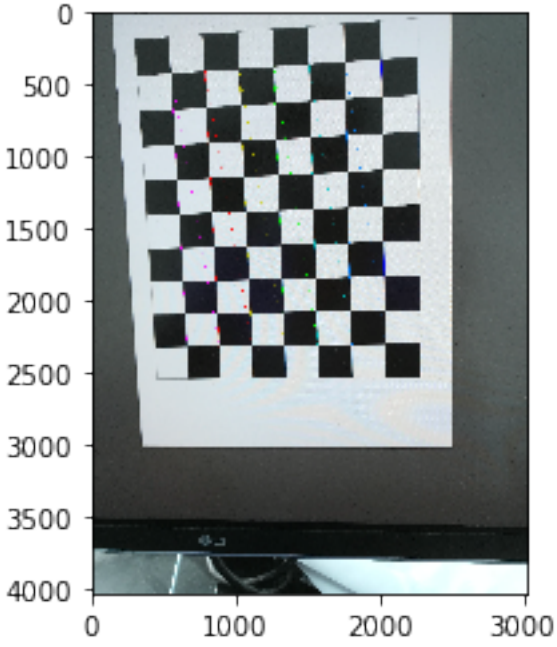

0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000

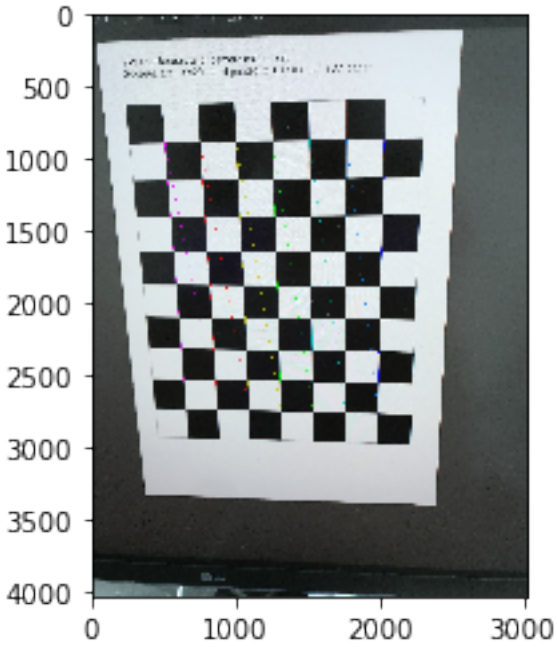

0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000



1.2 Perform calibration to get intrinsic parameters

[ ]:
```
ret, mtx, dist, rvecs, tvecs = cv.calibrateCamera(objpoints, imgpoints, gray.
 ↪shape[::-1], None, None)
print(mtx)
print(f'focal x: {mtx[0][0]}')
print(f'focal y: {mtx[1][1]}')
print(f'optical center x: {mtx[0][2]}')
print(f'optical center y: {mtx[1][2]}')
```

```
[[3.32965131e+03 0.00000000e+00 1.52788321e+03]
 [0.00000000e+00 3.33568206e+03 1.97474704e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
focal x: 3329.6513117460936
focal y: 3335.682059858252
optical center x: 1527.8832108670595
optical center y: 1974.7470364946348
```