In [11]:

```
import numpy as np
import cv2
import glob

checkerboard_dims = (8,6)
objp = np.zeros((48,3), np.float32)
objp[:,:2] = np.mgrid[0:8,0:6].T.reshape(-1,2)
criteria = cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001

objpoints = []
imgpoints = []

images = glob.glob("calibration_photos/*.jpg")

for image in images:
    img = cv2.imread(image)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    ret, corners = cv2.findChessboardCorners(gray, checkerboard_dims)

    if ret == True:
        objpoints.append(objp)

        refine_corners = cv2.cornerSubPix(gray, corners, (11,11), (-1, -1
        imgpoints.append(refine_corners)

        cv2.waitKey(500)
# cv2.DestroyAllWindows()

ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints,
np.savez("cam_cab.npz", mtx, dist, encoding="cp1252")
# np.savez_compressed('/home/miranda/bwsi-uav/laboratory_2024/week_1_Hw/c
# loaded = np.load('/home/kat/Downloads/cam_cab.npz')

mean_error = 0
for i in range(len(objpoints)):
    imgpoints2, _ = cv2.projectPoints(objpoints[i], rvecs[i], tvecs[i], m
    error = cv2.norm(imgpoints[i], imgpoints2, cv2.NORM_L2)/len(imgpoints
    mean_error += error
```

In [ ]: