Name : Aum M. Dhabalia **Date :** 09/09/2024

Roll No. : 21BEC027

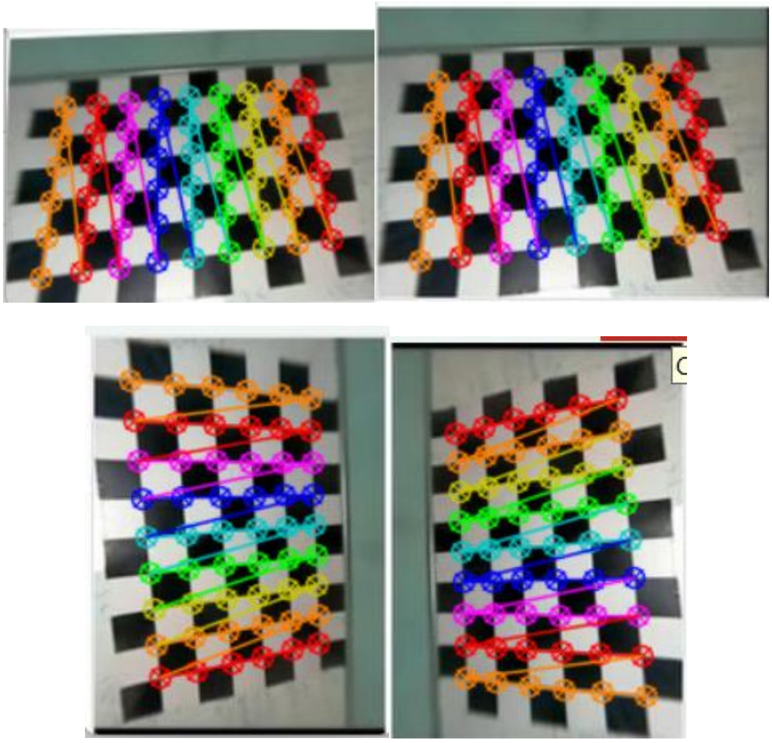
Experiment…5 – Camera Calibration

Objective : Calibrate camera and extract intrinsic and extrinsic parameters of the camera.

Import necessary libraries…

```
'''
Created on 9 September 2024 Mon 3:06:44 pm
'''
import cv2
import numpy as np
import os
import glob

CHECKERBOARD = (6,9)
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER,30,0.001)
objpts = []
imgpts = []
#World coordinates
objp3d = np.zeros((1,CHECKERBOARD[0]*CHECKERBOARD[1],3),np.float32)
objp3d[0,:,:2] = np.mgrid[0:CHECKERBOARD[0],0:CHECKERBOARD[1]].T.reshape(-1,2)
prev_img_shape = None
images = glob.glob('*.jpg')

for filename in images:
    image = cv2.imread(filename)
    Ig = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    ret,corners = cv2.findChessboardCorners(Ig,CHECKERBOARD,None)
    if ret == True:
        objpts.append(objp3d)
        corners2 = cv2.cornerSubPix(Ig,corners,(11,11),(-1,-1),criteria)
        imgpts.append(corners2)
        image = cv2.drawChessboardCorners(image,CHECKERBOARD,corners2,ret)
    cv2.imshow('img',image)
    cv2.waitKey(1000)
cv2.destroyAllWindows()

h, w = image.shape[:2]
ret,mtx,dist,r_vecs,t_vecs = cv2.calibrateCamera(objpts,imgpts,Ig.shape[::-1],None,None)

print("Camera matrix:")
print(mtx)
print("\nDistortion coefficient:")
print(dist)
print("\nRotation Vectors:")
print(r_vecs)
print("\n Translation Vectors:")
print(t_vecs)

image = cv2.imread(r"D:\Nirma Files\Computer Vision\Experiments\img1.jpg")
h,w = image.shape[:2]

newcamtx,roi = cv2.getOptimalNewCameraMatrix(mtx,dist,(w,h),1,(w,h))
dst = cv2.undistort(image,mtx,dist,None,newcamtx)

x,y,w,h = roi
dst = dst[y:y+h,x:x+w]
cv2.imwrite("CalibratedResult.jpg",dst)
cv2.imshow("Chess Board",dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

Output



Observation :

- The images of chessboard pattern for different cameras resulted in different camera calibration matrices.
- The circles marked are chessboard corners.
- Rotation vector, translation vector and distortion coefficient obtained were different for different cameras.

Conclusion:-

As the experiment performed,

- Calibration matrix was calculated by taking images of chessboard pattern using camera.
- different images of chessboard pattern were taken from different angle maintaining constant distance between camera and chessboard pattern.

Libraries and functions used are matplotlib, OpenCV, numpy, warpPerspective().