Exp 7:

SAVE YOUR PYTHON FILE IN A FOLDER “exp7”. SAVE ALL THE THREE PICS OF THE CHESSBOARD IN A FOLDER NAME “images” INSIDE “exp7”.

**Input image:**



**Code:**

import cv2

import numpy as np

import os

import glob

# Define the size of the checkerboard

CHECKERBOARD = (6, 9)

# Criteria for corner detection

criteria = (cv2.TERM\_CRITERIA\_EPS + cv2.TERM\_CRITERIA\_MAX\_ITER, 30, 0.001)

# Initialize object points in 3D

objectp3d = np.zeros((1, CHECKERBOARD[0] \* CHECKERBOARD[1], 3), np.float32)

objectp3d[0, :, :2] = np.mgrid[0:CHECKERBOARD[0], 0:CHECKERBOARD[1]].T.reshape(-1, 2)

# Initialize lists to store 3D and 2D points

threedpoints = []

twodpoints = []

# Get image filenames from the current directory

exp7 = glob.glob('\*.jpg')

# Iterate through each image

for images in exp7:

# Read the image

image = cv2.imread(images)

# Convert the image to grayscale

grayColor = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Find chessboard corners

ret, corners = cv2.findChessboardCorners(grayColor, CHECKERBOARD, cv2.CALIB\_CB\_ADAPTIVE\_THRESH + cv2.CALIB\_CB\_FAST\_CHECK + cv2.CALIB\_CB\_NORMALIZE\_IMAGE)

if ret == True:

# Refine corner locations

corners2 = cv2.cornerSubPix(grayColor, corners, (11, 11), (-1, -1), criteria)

# Draw chessboard corners on the image

image = cv2.drawChessboardCorners(image, CHECKERBOARD, corners2, ret)

# Append 3D and 2D points

threedpoints.append(objectp3d)

twodpoints.append(corners2)

# Display the image with chessboard corners

cv2.imshow('img', image)

cv2.waitKey(0) # Wait for 500 milliseconds

cv2.destroyAllWindows()

# Calibrate the camera

h, w = grayColor.shape[:2]

ret, matrix, distortion, r\_vecs, t\_vecs = cv2.calibrateCamera(threedpoints, twodpoints, (w, h), None, None)

# Print camera matrix, distortion coefficients, rotation vectors, and translation vectors

print("Camera matrix:")

print(matrix)

print("\nDistortion coefficients:")

print(distortion)

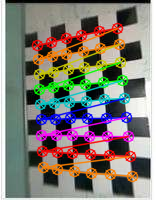
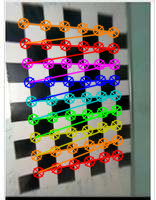
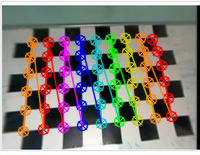
print("\nRotation vectors:")

print(r\_vecs)

print("\nTranslation vectors:")

print(t\_vecs)

**Output:**



Camera matrix:

[[35.5807773 0. 84.17582373]

[ 0. 35.96744246 95.12206623]

[ 0. 0. 1. ]]

Distortion coefficients:

[[ 4.71978156e-04 -4.36697358e-04 -4.04211977e-03 3.96502009e-04

3.65738539e-05]]

Rotation vectors:

(array([[-0.08552087],

[ 0.05355343],

[ 1.5084932 ]]), array([[-0.13626586],

[-0.0160124 ],

[ 3.07631327]]), array([[-0.03199432],

[ 0.08260852],

[-0.07334969]]))

Translation vectors:

(array([[ 4.63152182],

[-3.73966166],

[ 2.41039238]]), array([[2.32044167],

[4.0100118 ],

[2.40652674]]), array([[-3.15652078],

[-3.43870518],

[ 2.4599727 ]]))