天津工業大學
TIANGONG UNIVERSITY

实 验 报 告

课程名称： Computer Vision

专业班级：Artificial Intelligence

组　　别： 2020

学 号：W2010816010

任课教师：Zhu Xinjun

开课时间:2023- 2024年秋学期

开课学院:人工智能学院

Experiment 4

Camera Calibration Experiment

一、Purpose:

The purpose of this experiment is to calibrate a camera using a set of images of a chessboard pattern. Camera calibration is essential in computer vision and image processing to correct for distortions in images caused by camera lenses. The calibrated parameters, such as the camera matrix and distortion coefficients, can be used to undistort images for more accurate image processing and computer vision applications.

二、Steps:

a) Termination Criteria:
- A termination criteria is set using cv2.TERM_CRITERIA_EPS and cv2.TERM_CRITERIA_MAX_ITER with parameters 30 and 0.001, respectively. This criteria is used during the corner refinement process.

b) Object Points:
- A set of 3D object points (objp) is prepared, representing the corners of a chessboard pattern in the real world. These points are fixed for each image.

c) Image Points:
- For each image, the program attempts to find chessboard corners using cv2.findChessboardCorners.

d) Corner Refinement:
- If corners are found, they are refined using cv2.cornerSubPix.

e) Data Storage:
- Object points and image points are stored for calibration.

f) Camera Calibration:
- Using cv2.calibrateCamera, the camera matrix (mtx), distortion coefficients (dist), rotation vectors (rvecs), and translation vectors (tvecs) are calculated.

g) Data Saving:
- The calibration results are saved in a YAML file named "calibration_matrix.yaml."

三、Observation：

- The experiment assumes the presence of images in the current directory with a '.jpg' extension.
- The code successfully detects chessboard corners in the images and performs camera calibration.
- The number of images used for calibration is printed at the end of the process.
- Detected corners are visualized by drawing them on the images during the process.

四、Code and Result：

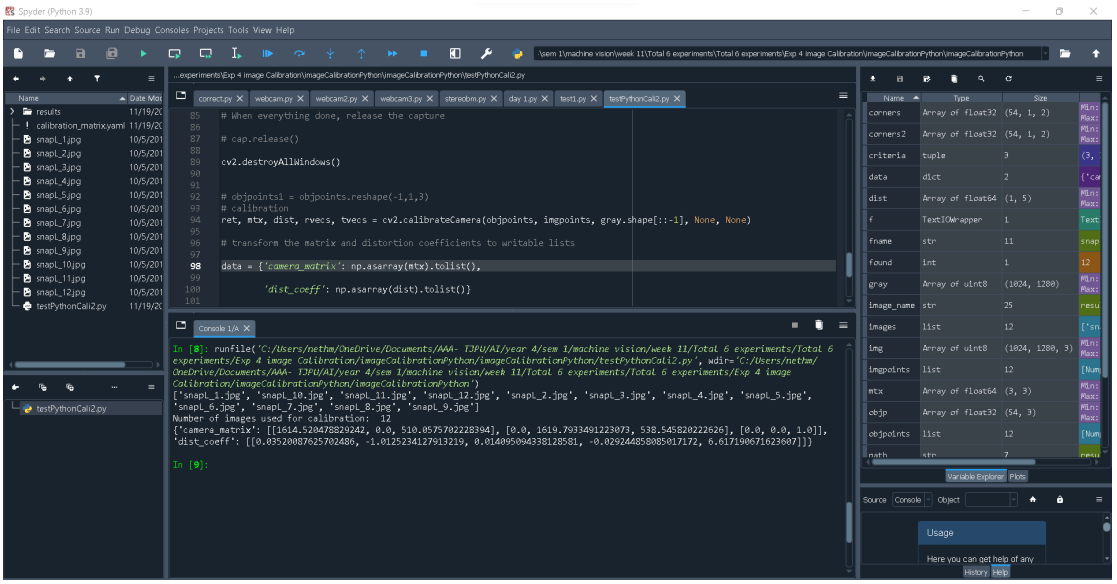
```
import numpy as np
import cv2
import glob
import yaml
import pathlib

# termination criteria

criteria = (cv2.TERM_CRITERIA_EPS + cv2.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((9*6,3), np.float32)

objp[:,:2] = np.mgrid[0:9,0:6].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('*.jpg')

path = 'results'

pathlib.Path(path).mkdir(parents=True, exist_ok=True)


found = 0
print(images)
for fname in images:  # Here, 10 can be changed to whatever number you like to choose

    img = cv2.imread(fname) # Capture frame-by-frame

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Find the chess board corners

    ret, corners = cv2.findChessboardCorners(gray, (9,6), None)
```

```
    # If found, add object points, image points (after refining them)
    if ret == True:
        objpoints.append(objp)   # Certainly, every loop objp is the same, in 3D.
        corners2 = cv2.cornerSubPix(gray,corners,(11,11),(-1,-1),criteria)
        imgpoints.append(corners2)
        # Draw and display the corners
        img = cv2.drawChessboardCorners(img, (9,6), corners2, ret)
        found += 1
        cv2.imshow('img', img)
        cv2.waitKey(500)
        image_name = path + '/calibresult' + str(found) + '.png'
        cv2.imwrite(image_name, img)

print("Number of images used for calibration: ", found)


cv2.destroyAllWindows()

ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, gray.shape[::-1], None, None)

# transform the matrix and distortion coefficients to writable lists

data = {'camera_matrix': np.asarray(mtx).tolist(),
        'dist_coeff': np.asarray(dist).tolist()}

print(data)

with open("calibration_matrix.yaml", "w") as f:
    yaml.dump(data, f)
```

Result in the console

```
['snapL_1.jpg', 'snapL_10.jpg', 'snapL_11.jpg', 'snapL_12.jpg', 'snapL_2.jpg', 'snapL_3.jpg', 'snapL_4.jpg', 'snapL_5.jpg', 'snapL_6.jpg', 'snapL_7.jpg', 'snapL_8.jpg', 'snapL_9.jpg']
Number of images used for calibration:  12
{'camera_matrix': [[1614.520478829242, 0.0, 510.0575702228394], [0.0, 1619.7933491223073, 538.545820222626], [0.0, 0.0, 1.0]], 'dist_coeff': [[0.03520087625702486, -1.0125234127913219, 0.014095094338128581,
```

-0.029244858085017172, 6.617190671623607]]}



五、Conclusion：

- Camera calibration is a crucial step in computer vision applications to correct for distortions caused by camera lenses.
- The experiment successfully achieves its purpose by calibrating the camera using a set of chessboard images.
- The calibrated parameters are saved for future use, enabling accurate image processing in applications such as object recognition or measurement.