



南京邮电大学
Nanjing University of Posts and Telecommunications

(2023-2024 学年 第 2 学期)
计算机视觉实验报告

题 目 相机标定

所在学院 自动化学院、人工智能学院

专 业 人工智能

年级班级 B210416

学 号 B21080526

姓 名 单家俊

授课教师 范保杰

2024 年 5 月 16 日

实验二 Matlab 相机标定

一、实验目的

学会使用相机标定工具箱

二、实验内容

1. 标定相机内参
2. 标定相机外参

三、实验说明

实验代码：

```
import cv2 # 导入 OpenCV 库。
import numpy as np # 导入 NumPy 库，用于高效的矩阵和数组操作。
import os # 导入 OS 库，用于与操作系统交互。
import glob # 导入 glob 库，用于文件路径名的模式匹配。

# 定义棋盘格的尺寸，这里是 6x9。
CHECKERBOARD = (6, 9)
# 设置寻找角点的终止准则。
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)
# 创建列表以存储棋盘格的 3D 点。
objpoints = []
# 创建列表以存储棋盘格角点在图像中的 2D 点。
imgpoints = []
# 初始化 3D 点的坐标，这里仅初始化 x 和 y，z 设置为 0。
objp = np.zeros((1, CHECKERBOARD[0] * CHECKERBOARD[1], 3), np.float32)
objp[0, :, :2] = np.mgrid[0:CHECKERBOARD[0],
0:CHECKERBOARD[1]].T.reshape(-1, 2)
prev_img_shape = None # 初始化之前图像的尺寸。
# 使用 glob 模式匹配来获取所有棋盘格图片的路径。
# images = glob.glob('./images/IMG_20240402_160207.jpg')
# for fname in images:
img = cv2.imread('picture1.png') # 读取每一张图片。
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # 将图片转换为灰度图，因为寻找角点在灰度图上进行。

# 寻找棋盘格角点。
```

```

ret, corners = cv2.findChessboardCorners(gray, CHECKERBOARD,
cv2.CALIB_CB_ADAPTIVE_THRESH + cv2.CALIB_CB_FAST_CHECK +
cv2.CALIB_CB_NORMALIZE_IMAGE)
# 如果找到足够数量的角点，则细化它们的位置并将它们添加到列表中。
if ret == True:
    objpoints.append(objp)
    corners2 = cv2.cornerSubPix(gray, corners, (11, 11), (-1, -1), criteria)
    imgpoints.append(corners2)

    # 将找到的角点绘制到图片上，以便可视化。
    img = cv2.drawChessboardCorners(img, CHECKERBOARD, corners2, ret)

cv2.imshow('img', img) # 显示图片。
cv2.waitKey(0) # 等待用户按键。

cv2.destroyAllWindows() # 关闭所有 OpenCV 窗口。

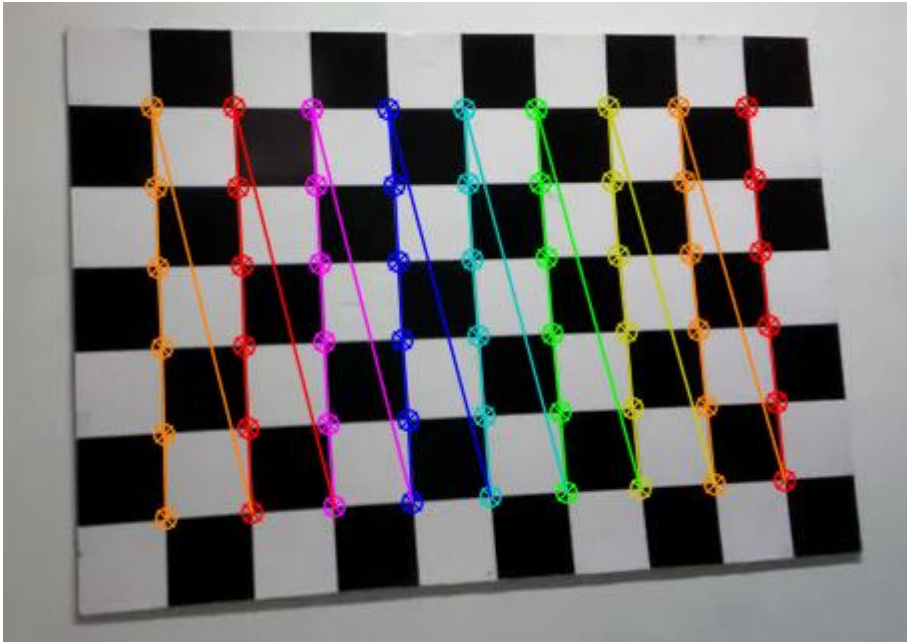
h, w = img.shape[:2] # 获取最后处理的图片的尺寸。

# 使用 3D 点和对应的 2D 图像点进行相机标定。
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints,
gray.shape[:-1], None,
None)

# 打印相机标定的结果。
print("Camera matrix : \n")
print(mtx) # 相机矩阵
print("dist : \n")
print(dist) # 畸变系数
print("rvecs : \n")
print(rvecs) # 旋转向量
print("tvecs : \n")
print(tvecs) # 平移向量

```

实验结果：



Camera matrix :

```
[[464.49405703    0.        256.69513204]
 [   0.        450.33390296  211.82106216]
 [   0.         0.         1.         ]]
```

dist :

```
[[ 0.14528703 -0.02491773 -0.00448671 -0.0158665  -0.11402032]]
```

rvecs :

```
(array([[ -0.21669059],
        [-0.04920857],
        [ 1.53448393]]),)
```

tvecs :

```
(array([[ 7.13158331],
        [-2.03725579],
        [12.66491579]]),)
```

进程已结束，退出代码为 0

四、实验心得

在这次相机标定实验中，我使用了 OpenCV 库和 NumPy 库，对相机进行了标定。通过处理多张棋盘格图像，成功计算出了相机的内参矩阵、畸变系数、旋转向量和平移向量。

首先，我编写了代码以读取棋盘格图像，并将其转换为灰度图。然后，使用 `cv2.findChessboardCorners` 函数检测棋盘格的角点，并通过 `cv2.cornerSubPix`

函数进一步细化角点的位置。在成功检测到角点后，我将这些角点存储起来，用于相机标定。在处理完所有图像后，我使用 `cv2.calibrateCamera` 函数进行相机标定，得到了相机矩阵（内参矩阵）、畸变系数、旋转向量和平移向量。

标定结果显示了相机的内外参数。相机矩阵（`mtx`）表示了相机的焦距和光心位置，畸变系数（`dist`）显示了相机镜头的径向和切向畸变情况。旋转向量（`rvecs`）和平移向量（`tvecs`）则描述了相机在三维空间中的姿态和位置。

通过实验，我体会到准确检测和细化棋盘格角点是确保标定结果准确的关键。使用 `cv2.cornerSubPix` 函数来细化角点位置，显著提高了检测精度，进而提高了相机标定的准确性。此外，为了获得准确和稳定的标定结果，使用多张不同角度和位置的棋盘格图像进行标定是必要的。多样化的图像能够提供更多的特征点，提高标定的鲁棒性和精度。

实验中得到的畸变系数显示了相机镜头的径向和切向畸变情况，了解和校正这些畸变对于图像处理和计算机视觉应用至关重要。得到的相机矩阵和畸变系数可以用于图像校正和 3D 重建。通过校正图像中的畸变，可以显著提高图像的质量和精度，为后续的计算机视觉任务打下基础。

这次实验让我掌握了相机标定的基本方法，理解了相机内外参数对图像处理的影响。这为我今后在计算机视觉领域的研究和应用打下了坚实的基础。为了进一步提高标定结果的准确性，可以使用更多的棋盘格图像进行标定，尤其是增加不同视角和距离的图像。此外，在角点检测前，对图像进行一些预处理，如去噪和平滑处理，可以提高角点检测的成功率和精度。总的来说，这次实验让我对相机标定有了更深入的了解，并为今后的图像处理工作提供了宝贵的经验。