



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 特征点检测与匹配

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

实验三 特征点检测与匹配

一、实验目的

检测图像中的特征点（如 harris 等），并完成两幅或多幅图像中特征点的匹配。

二、实验内容

1. 检测图像中特征点
2. 匹配图像中的特征点

三、实验说明

根据上课内容，深入理解 harris 角点检测的原理，编程实现经典的 harris 角点检测程序，利用欧氏距离作为度量准则，来判断特征点间的相似程度，完成特征点的匹配。

实验代码：

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import cv2

matplotlib.use('TkAgg')
# Harris 角点检测
img = plt.imread("wangba2.jpeg").copy()
img_ = img.copy()
gray = np.float32(cv2.cvtColor(img, cv2.COLOR_BGR2GRAY))
harris = cv2.cornerHarris(gray, 2, 3, 0.04)
harris = cv2.dilate(harris, None) # 膨胀, 方便显示
img[harris > 0.01 * harris.max()] = [255, 0, 0]

plt.subplot(1, 2, 1)
plt.axis("off")
plt.imshow(img_)
plt.subplot(1, 2, 2)
plt.axis("off")
plt.imshow(img)
plt.show()
# ORB 特征点提取
img = plt.imread("wangba.jpeg").copy()

orb = cv2.ORB_create() # 可以自定义很多参数
kp = orb.detect(img) # 特征点
```

```

kp_img = cv2.drawKeypoints(img, keypoints=kp, outImage=None, color=300)

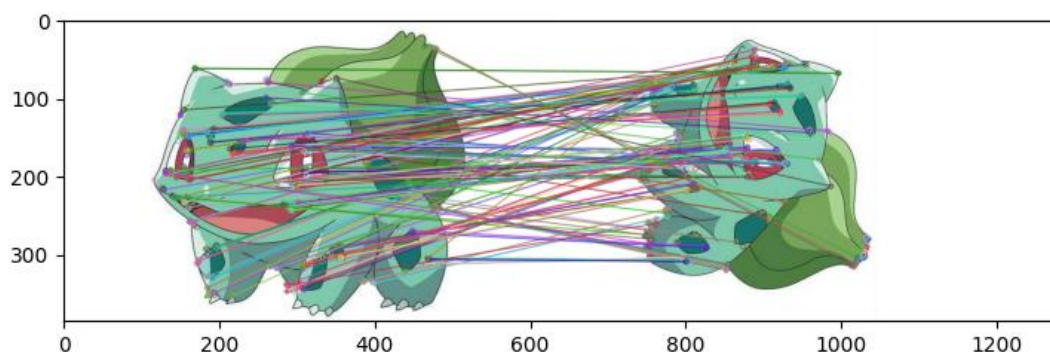
# 绘制
plt.subplot(1, 2, 1)
plt.axis("off")
plt.imshow(img)
plt.subplot(1, 2, 2)
plt.axis("off")
plt.imshow(kp_img)
plt.show()

# ORB 特征点匹配
img1 = plt.imread("wangba.jpeg").copy()
img2 = plt.imread("wangba2.jpeg").copy()
orb = cv2.ORB_create()
# 特征点、描述子
kp1, des1 = orb.detectAndCompute(img1, None)
kp2, des2 = orb.detectAndCompute(img2, None)
# 匹配
match = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True).match(des1, des2)
match = sorted(match, key=lambda x: x.distance)
# 取最近的minN 个绘制
minN = len(match)
# minN = 40
img3 = cv2.drawMatches(img1, kp1, img2, kp2, match[:minN], None)
plt.imshow(img3)
plt.show()

```

实验结果:





四、实验心得

在这次实验中，我使用了 Harris 角点检测器来检测图像中的特征点，并完成了两幅或多幅图像中特征点的匹配。通过实验，我对特征点检测和匹配的过程有了更深入的理解。

特征点检测是图像处理中的关键步骤。Harris 角点检测器能够有效地检测图像中的角点，这些角点在不同视角和光照条件下具有较高的稳定性和可辨识度。在实验中，Harris 角点检测器准确地识别出图像中的显著特征点，为后续的特征描述和匹配提供了良好的基础。

通过实验结果的可视化，我观察到大部分特征点能够正确匹配，但也存在少量误匹配，主要出现在图像边缘区域或纹理较为均匀的区域。为提高匹配精度，未来可以尝试更多的特征点检测算法，结合多种特征描述子，并优化特征匹配算法，使用如 RANSAC 等鲁棒性算法过滤掉错误匹配。

这次实验让我掌握了 Harris 角点检测器的使用方法，也深入理解了特征点匹配的原理和挑战，为我今后在图像处理和计算机视觉领域的研究打下了基础。

五、思考题

Harris 特征点的不足与改进对策？

不足：

1. 对尺度变化不敏感：

Harris 角点检测器对图像的尺度变化不敏感，这意味着在不同尺度下检测到的特征点可能会有所不同。对于图像中存在不同大小物体的情况，Harris 角点检测器可能无法稳定检测出相同的特征点。

2. 对旋转不敏感：

Harris 角点检测器对图像的旋转不敏感，导致在旋转后图像中的特征点检测结果可能会发生变化，这限制了其在旋转不变性要求较高的应用场景中的有效性。

3. 噪声敏感性：

Harris 角点检测器对图像噪声较为敏感，特别是在高噪声环境下，可能会产生大量的误检特征点，影响后续的特征匹配和分析。

4. 特征点的分布不均匀：

Harris 角点检测器在检测特征点时，可能会在高频区域检测到大量特征点，而在低频区域检测到的特征点较少，导致特征点分布不均匀，影响匹配效果。

改进对策：

1. 使用多尺度特征检测器：

为解决尺度不敏感的问题，可以使用多尺度特征检测器，如 SIFT（尺度不变特征变换）和 SURF（加速稳健特征），它们能够在不同尺度下检测稳定的特征点，增强对尺度变化的鲁棒性。

2. 引入旋转不变性：

为解决旋转不敏感的问题，可以采用 ORB（定向快速和旋转不变的特征）等旋转不变特征检测器，ORB 在特征检测和描述过程中加入了旋转不变性，使得特征点在旋转图像中也能保持稳定。

3. 噪声预处理和后处理技术：

为减小噪声的影响，可以在图像预处理阶段使用去噪算法，如高斯滤波、中值滤波等，减少图像噪声。另外，可以结合 RANSAC（随机抽样一致性）等算法，在特征匹配阶段过滤掉噪声导致的误匹配特征点，提高匹配的准确性。

4. 结合其他特征点检测器：

为解决特征点分布不均的问题，可以结合 FAST（特征加速段检测器）等快速特征检测器，均衡特征点在图像中的分布。通过多种检测器的结合，可以获取更多的有效特征点，提高匹配的效果。