

计算机视觉实践-练习 1

122106222784 贺梦瑶

1、实验目标

- 理解关键点检测算法 DOG 原理。
- 理解尺度变化不变特征 SIFT。
- 采集一系列局部图像，自行设计拼接算法。
- 使用 Python 实现图像拼接算法。

2、实现说明

```
1. # 拼接函数
2. def stitch(self, images, ratio = 0.75, reprojThresh = 4.0, showMatches = True):
3.     imageB, imageA = images
4.     # 计算特征点和特征向量
5.     kpsA, featureA = self.detectAndDescribe(imageA)
6.     kpsB, featureB = self.detectAndDescribe(imageB)
7.     # 匹配两张图片的特征
8.     M = self.matchKeypoints(kpsA, kpsB, featureA, featureB, ratio,
                               reprojThresh)
9.     # 没有获得匹配点
10.    if not M:
11.        return None
12.
13.    matches, H, status = M
14.    # 将图片 B 进行视角变换, 获取中间结果
15.    result = cv2.warpPerspective\
16.        (imageB, H, (imageA.shape[1] + imageB.shape[1], imageA.s
17.         hape[0]))
18.    # 将图片 B 传入
19.    result[0:imageA.shape[0], 0:imageA.shape[1]] = imageA
20.    # 检测是否需要显示图片匹配
21.    if showMatches:
22.        # 生成匹配图片
23.        vis = self.drawMatches(imageA, imageB, kpsA, kpsB, matches, status)
24.        # 返回结果
25.        return result, vis
26.
```

```
27.         # 返回匹配结果
28.         return result
```

```
1. # 检测特征点并计算描述子
2. def detectAndDescribe(self, image):
3.     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
4.     descriptor = cv2.SIFT_create()
5.     kps, features = descriptor.detectAndCompute(gray, None)
6.     kps = np.float32([kp.pt for kp in kps])
7.     return kps, features
```

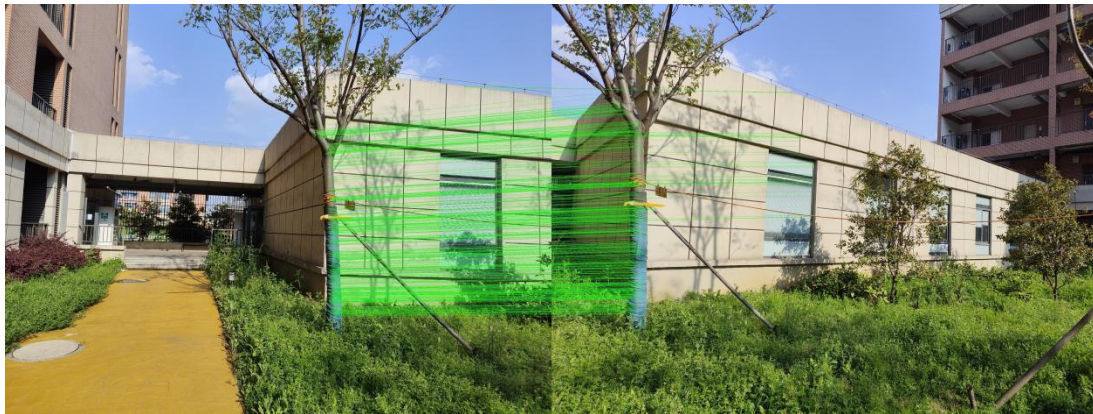
```
1. # 关键点匹配
2. def matchKeypoints(self, kpsA, kpsB, featureA, featureB, ratio, repro
    jThresh):
3.     # 建立匹配器
4.     matcher = cv2.BFMatcher()
5.     # 使用 KNN 检测来自 AB 图的 SIFT 特征匹配
6.     rawMatches = matcher.knnMatch(featureA, featureB, 2)
7.     # 过滤
8.     matches = []
9.     for m in rawMatches:
10.         if len(m) == 2 and m[0].distance < m[1].distance * ratio
            :
11.             matches.append((m[0].trainIdx, m[0].queryIdx))
12.     # 获取匹配对的点坐标
13.     if len(matches) > 4:
14.         ptsA = np.float32([kpsA[i] for (_, i) in matches])
15.         ptsB = np.float32([kpsB[i] for (i, _) in matches])
16.         # 计算 H 矩阵
17.         H, status = cv2.findHomography(ptsB, ptsA, cv2.RANSAC, r
            eprojThresh)
18.         return matches, H, status
```

3、结果截图

(1) 需拼接图像



(2) 特征匹配结果



(3) 图像拼接结果



4、运行说明

在代码文件所在的同一文件夹下存储需要拼接的图像，其中 1.jpg 和 2.jpg 为需要拼接的图像，sift.jpg 为匹配过程图像，result.jpg 为匹配结果图像。

