

CV Assignment3

姓名	学号	邮箱
刘思远	221220067	liusiyuan@smail.nju.edu.cn

此次作业是本学期《计算机视觉表征与识别》课程的最后一次作业，与前两次相比，感觉难度较大。

Task1: Image Stitching

该任务要完成对两张图像的拼接，根据实验手册，主要包括特征提取、特征匹配、单应矩阵的构建和图像融合等步骤。

特征点提取

由于SIFT算法相比ORB、BRIEF等方法，对尺度变化和旋转变换具有更强的鲁棒性，所以使用SIFT作为特征提取工具。

为了提升特征检测的密度和鲁棒性，设置了较高的特征数threshold ($n_{features} = 8000$)，同时设置了较高的调整后对比度以及边缘相应参数。

特征点匹配

使用FLANN（近似最近邻快速库）实现特征匹配，并结合Lowe's Ratio Test筛选匹配结果，以降低错误匹配率。匹配对的距离比值的threshold设置为0.7，以确保匹配的可靠性和准确性。

单应矩阵

在得到匹配的点对之后，使用RANSAC算法来估计图像之间的单应矩阵。为了适应更加复杂的图像case，设置了较高的迭代次数和置信度。

图像变换和融合

为了提高拼接图像的流畅度，减少视觉上的拼接痕迹，采用了如下3个方法来加以改进：

- 计算变换后的输出画布大小，构造平移矩阵，调整坐标偏移；

```
1 | translation = np.array([[1, 0, -x_min], [0, 1, -y_min], [0, 0, 1]])
```

- 在图像重叠区域，构造线性渐变的mask，然后加上gaussian blur来实现平滑过渡。

```
1 # 创建从左到右的线性渐变
2 for x in range(w):
3     if x <= min_x:
4         mask[:, x] = 0.0
5     elif x >= max_x:
6         mask[:, x] = 1.0
7     else:
8         # 在重叠区域内创建平滑过渡
9         alpha = (x - min_x) / (max_x - min_x)
10        mask[:, x] = alpha
11
12 # 应用高斯模糊使过渡更平滑
13 mask = cv2.GaussianBlur(mask, (21, 21), 10)
```

此外，观察到按照上述方法拼接后的图像会存在黑边，也就是黑色无效区域，带来了很差的视觉效果。因此，在上述步骤基础上，最后基于阈值分割法提取有效边界并裁剪，以此来消除黑边。

Task2: Multi-Image Stitching

首先，抛开算法内容和实现不谈，一个重要的问题是内存占用。由于部分cases里面要拼接的图像数量很多，而在运算时要同时在内存里面存储已加载的所有图片的运算结果，所以很容易就会因为超出内存限制而导致程序被kill掉。为了解决此问题，主要采取了以下方式降低内存占用：

1. 对输入图像进行缩放来减少内存占用

```
1 def resize_image_if_needed(img, max_dimension=400): # 大幅降低到400
2     """如果图像太大则缩放"""
3     h, w = img.shape[:2]
4     max_dim = max(h, w)
5     if max_dim > max_dimension:
6         scale = max_dimension / max_dim
7         new_w = int(w * scale)
8         new_h = int(h * scale)
9         img = cv2.resize(img, (new_w, new_h), interpolation=cv2.INTER_AREA)
10        print(f"图像从 {w}x{h} 缩放到 {new_w}x{new_h}")
11    return img
```

- 使用轻量级的SIFT来进行特征检测（即把 $n_{features}$ 减少到1500）
- 及时清理临时变量并调用垃圾回收机制。

接着，重点介绍一下自己是如何确定图像的拼接顺序的。

(1) 基于图像之间的特征匹配关系，构建连接图，其中节点代表单个图像，边的权重代表匹配点的数量，只有当匹配点数量超过某个threshold的时候，才会建立边；

(2) 如果图不连通，则选取最大的连通分量拼接；

(3) 选择度数最高的节点作为拼接的起始点，接着使用贪心的思想，逐步选择权重最大的边，以此确定拼接的顺序。

关于特征提取和匹配的部分，直接使用Task1已经实现好的接口即可。

到这里，这学期的CV课就结束了。总的来看，三次的实验收获还是蛮大的，至少可以不用pytorch框架写图像处理相关代码了。浅浅完结撒花一下，暑假完成关于composed image retrieval的综述。