# 计算机视觉实践-练习1实验报告

目录

[计算机视觉实践-练习1实验报告 1](#_Toc131931703)

[一、 实验目的 1](#_Toc131931704)

[二、 实验原理 1](#_Toc131931705)

[三、 实验步骤 3](#_Toc131931706)

[四、 数据集 5](#_Toc131931707)

[五、 代码程序 5](#_Toc131931708)

[六、 实验结果 10](#_Toc131931709)

[七、 实验分析与总结 12](#_Toc131931710)

1. 实验目的

* 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

1. 实验原理

**SIFT算法**

SIFT，即尺度不变特征变换（Scale-invariant feature transform，SIFT），是用于[图像处理](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86/294902)领域的一种描述。这种描述具有尺度不变性，可在图像中检测出关键点，是一种局部特征描述子。其特点如下：

* SIFT特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性；
* 区分性（Distinctiveness）好，信息量丰富，适用于在海量特征数据库中进行快速、准确的匹配；
* 多量性，即使少数的几个物体也可以产生大量的SIFT特征向量；
* 高速性，经优化的SIFT匹配算法甚至可以达到实时的要求；
* 可扩展性，可以很方便的与其他形式的特征向量进行联合。

**Lowe’s算法**

为了进一步筛选匹配点，来获取优秀的匹配点。一般会采用Lowe’s算法来进一步获取优秀匹配点。 为了排除因为图像遮挡和背景混乱而产生的无匹配关系的关键点，SIFT的作者Lowe提出了比较最近邻距离与次近邻距离的SIFT匹配方式：取一幅图像中的一个SIFT关键点，并找出其与另一幅图像中欧式距离最近的前两个关键点，在这两个关键点中，如果最近的距离除以次近的距离得到的比率ratio少于某个阈值T，则接受这一对匹配点。因为对于错误匹配，由于特征空间的高维性，相似的距离可能有大量其他的错误匹配，从而它的ratio值比较高。显然降低这个比例阈值T，SIFT匹配点数目会减少，但更加稳定，反之亦然。

**RANSAC算法**

随机抽样一致算法（RANdom SAmple Consensus,RANSAC）,采用迭代的方式从一组包含离群的被观测数据中估算出数学模型的参数。RANSAC算法假设数据中包含正确数据和异常数据（或称为噪声）。正确数据记为内点（inliers），异常数据记为外点（outliers）。同时RANSAC也假设，给定一组正确的数据，存在可以计算出符合这些数据的模型参数的方法。该算法核心思想就是随机性和假设性，随机性是根据正确数据出现概率去随机选取抽样数据，根据大数定律，随机性模拟可以近似得到正确结果。假设性是假设选取出的抽样数据都是正确数据，然后用这些正确数据通过问题满足的模型，去计算其他点，然后对这次结果进行一个评分。

1. 实验步骤

大体上可分为以下几个步骤：

1. 从输入的两张图片里检测关键点、提取局部不变特征。(sift)
2. 匹配的两幅图像之间的特征(Lowe’s算法)
3. 使用RANSAC算法利用匹配特征向量估计单映矩阵（homography matrix）。
4. 利用Step3得到的单映矩阵应用扭曲变换。
5. 除去生成的缝合图像的黑边并输出结果。

其中特征检测的基本步骤为：

**1. 尺度空间极值检测：**

搜索所有尺度上的图像位置。通过高斯微分函数来识别潜在的对于尺度和旋转不变的兴趣点。

**2. 关键点定位**

在每个候选的位置上，通过一个拟合精细的模型来确定位置和尺度。关键点的选择依据于它们的稳定程度。

**3. 方向确定**

基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向。所有后面的对图像数据的操作都相对于关键点的方向、尺度和位置进行变换，从而提供对于这些变换的不变性。

**4. 关键点描述**

在每个关键点周围的邻域内，在选定的尺度上测量图像局部的梯度。这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变化。

**SIFT特征匹配阶段：**

第一阶段：SIFT特征的生成，即从多幅图像中提取对尺度缩放、旋转、亮度变化无关的特征向量。

第二阶段：SIFT特征向量的匹配。

SIFT特征的生成一般包括以下几个步骤：

1. 构建[尺度空间](https://baike.baidu.com/item/%E5%B0%BA%E5%BA%A6%E7%A9%BA%E9%97%B4)，检测极值点，获得尺度不变性。

2. 特征点过滤并进行精确定位。

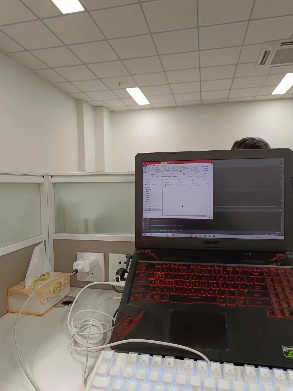
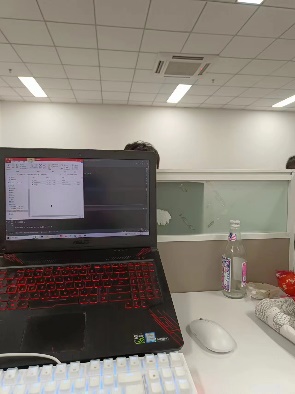
3. 为特征点分配方向值。

4. 生成特征描述子。

当两幅图像的SIFT特征向量生成以后，下一步就可以采用关键点特征向量的[欧式距离](https://baike.baidu.com/item/%E6%AC%A7%E5%BC%8F%E8%B7%9D%E7%A6%BB)来作为两幅图像中关键点的相似性判定度量。取图1的某个关键点，通过遍历找到图像2中的距离最近的两个关键点。在这两个关键点中，如果最近距离除以次近距离小于某个阈值，则判定为一对匹配点。

1. 数据集

这里的测试图片为自己拍摄的水平平移的图片。如下两图所示：

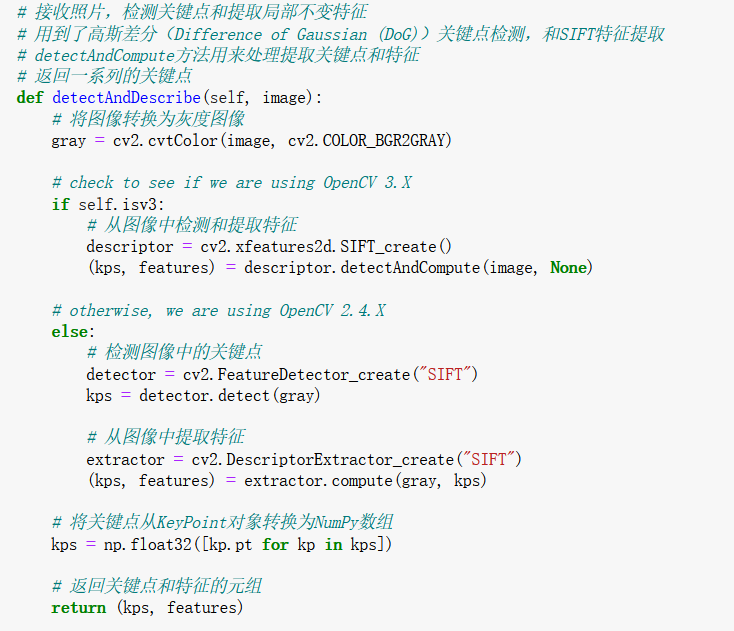
 

1. 代码程序

提取并匹配图像特征后进行缝合的总体调用部分如下：



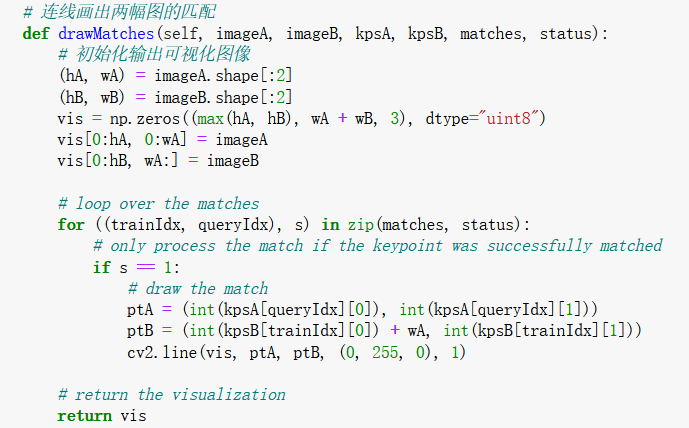
接收照片，检测关键点和提取局部不变特征部分如下：



将提取的特征进行匹配部分如下：

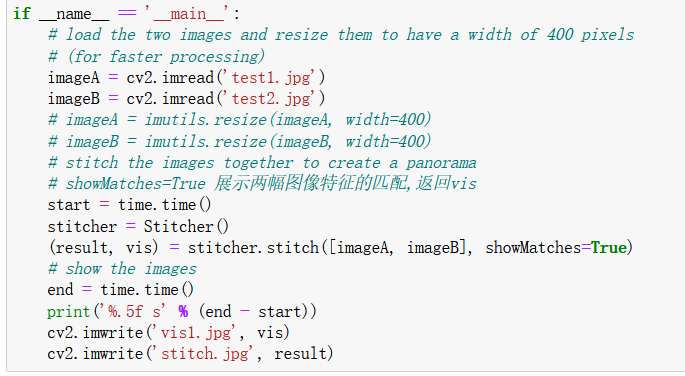


连线画出两幅图的匹配部分如下：



除去拼接后的图像黑边部分如下：

程序的输入输出部分如下：

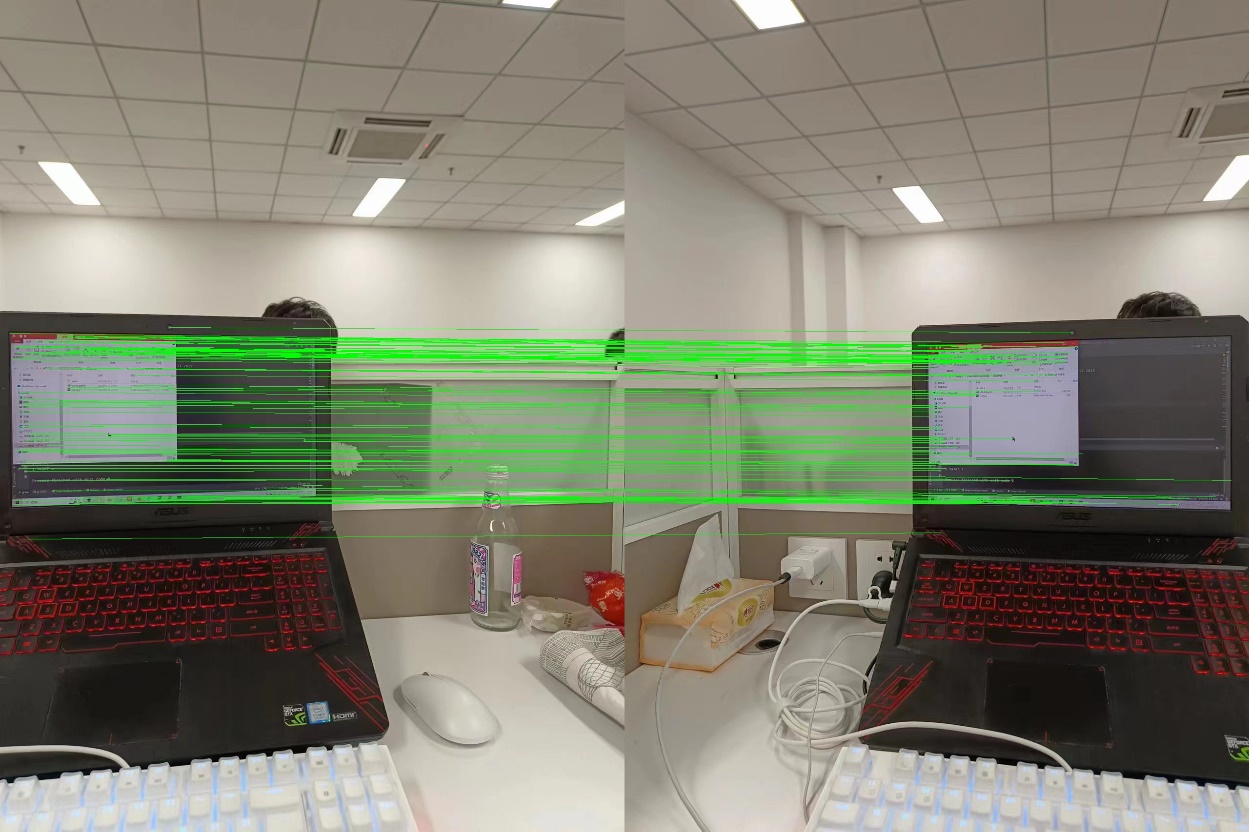


1. 实验结果

待拼接的两张测试图片，如下两图所示：

提取并匹配特征后输出结果如下：



拼接后的输出图像如下：



1. 实验分析与总结

Sitich调用detectAndDescribe，检测两张图片里的关键点、提取局部不变特征。

有了关键点和特征，sitich调用matchKeypoints方法来匹配两张图片里的特征。如果返回匹配的M为None，就是因为现有的关键点不足以匹配生成全景图。假设M不返回None，拆包返回元组，包含关键点匹配matches、从RANSAC算法中得到的最优单映射变换矩阵H以及最后的单映计算状态列表status，用来表明那些已经成功匹配的关键点。

有了最优单映射变换矩阵H后，就可将两张图片“缝合起来”。stitch调用cv2.warpPerspective进行缝合。这样，就返回一个拼接的图片。

最后sitich调用drawMatches函数用来将两张图片关键点的匹配可视化，去除拼接后产生的黑边后即为最后的输出结果。