## 图像拼接实验

## 实验说明

本实验使用python语言编写，在实现过程中使用了OpenCV、Numpy等核心库实现图像的拼接和融合功能。实验中用到的核心算法是关键点检测算法DOG原理和尺度变化不变特征SIFT。

实验设计了一个图像拼接类Stitcher，该类定义了图像拼接和图像融合函数，完成了本实验中涉及的所有核心算法。

## 代码实现说明

本次实验设计了一个Stitcher类用以实现图像拼接和图像融合，整体的处理步骤是：首先从输入的两张图片中检测关键点，提取局部不变特征，接着匹配两幅图像之间的特征，然后将两幅图像中的匹配点连线，最后根据匹配出的关键点进行特征融合以得到融合的图像。

提取关键点和匹配特征部分是本实验的重点和核心，分为两个阶段，一是SIFT特征的生成，即从图像中提取对尺度缩放、旋转、亮度变化无关的特征向量；二是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。取图像1的某个关键点并通过遍历找到图像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。

在Stitcher类中定义了实现以上所有功能的函数，其中detectAndDescribe用于检测图片的关键点并提取局部不变特征；matchKeypoints用于匹配特征；drawMatches将两幅图像中匹配的关键点连线，返回可视化结果；stitch读取输入的图片，串联上述函数完成最后的拼接和融合功能。

## 结果截图

本次实验使用两张水平方向平移的图像，并尽可能减少垂直方向的移动。原图像如下图所示：



两幅图像的匹配关键点连线如下：



最终拼接结果如下：



## 运行说明

在运行前，需要准备用于拼接照片，将其分别命名为image1和image2，并且放到项目文件的同级目录下。此外，由于项目中使用了OpenCV、Numpy、pillow和matplotlib库，需要先下载安装这些库。之后运行项目文件即可在文件中查看拼接后的图片以及在拼接时程序匹配到的特征点。