图像拼接实验报告

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.具体要求

不允许使用现成的图像拼接程序

在采图过程中可尽可能减少相机在垂直方向的运动,但不能假设图像只存在水平方向平移

需包含图像融合部分，从而减少拼接图像中局部图像的“接缝”

1. 实验原理

**3.1 SIFT描述子定义**

SIFT的全称是Scale Invariant Feature Transform，尺度不变特征变换，由加拿大教授David G.Lowe提出的。SIFT特征对旋转、尺度缩放、亮度变化等保持不变性，是一种非常稳定的局部特征。

**3.2 SIFT特征**

1、图像的局部特征，对旋转、尺度缩放、亮度变化保持不变，对视角变化、仿射变换、噪声也保持一定程度的稳定性。

2、独特性好，信息量丰富，适用于海量特征库进行快速、准确的匹配。

3、 多量性，即使是很少几个物体也可以产生大量的SIFT特征

4、高速性，经优化的SIFT匹配算法甚至可以达到实时性

5、 扩招性，可以很方便的与其他的特征向量进行联合。

**3.3 SIFT检测过程**

1.构造高斯差分空间图像。

2.寻找极大极小值点

3.精确定位极值点

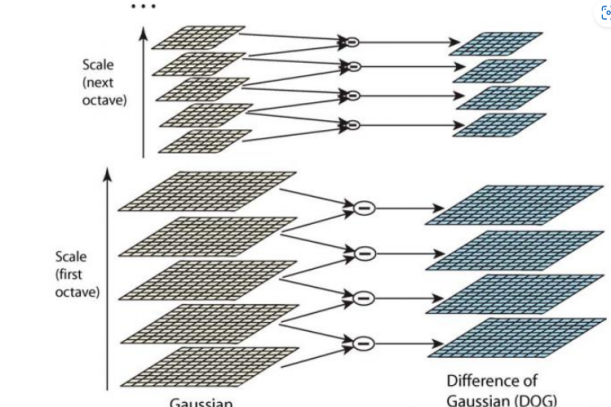
4.选取特征点主方向

5. 构造特征点描述算子。

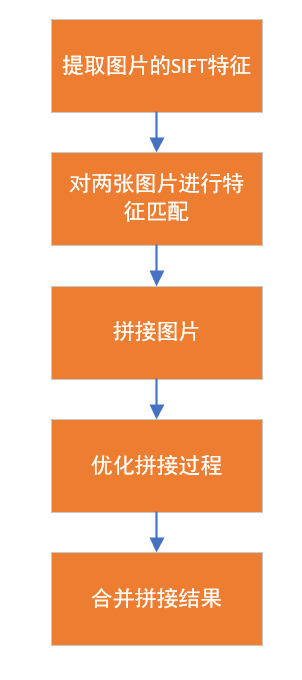
**3.4 DOG高斯差分金字塔**

对于二维图像I(x，y)，建立图像的DOG （difference of guassians，即高斯差分）金字塔，DOG尺度空间含义为：可以用一个尺度空间的高斯函数和图像的卷积来表示。

为确定特征点所在的位置，首先需要建立一个高斯金字塔。得到高斯金字塔后，再通过两个相邻的高斯尺度空间做差，得到高斯差分DOC金字塔。高斯差分金字塔建立后，特征点就是DOG尺度空间众多极值点。



1. 代码流程



代码的主要流程如上图所示，首先提取图片中的SIFT特征点，设置好合适的阈值，进行接下来的特征匹配，特征匹配中的配对点需要进行筛选，本文使用了KNN匹配和RANSAC的方法进行筛选，最终得到合适的匹配点。之后就可以以一张图片作为参考，计算另一张图片的单应性矩阵，获得第二张图在第一张图片下的位置分布。

之后直接将两者进行拼接，会产生明显的接缝，为了优化接缝的效果，采用重叠部分加权平均的方法，优化了接缝，使得图像看起来更加自然。最后将图像进行合并。

1. 结果展示

示例2：

输入图片：



没有处理拼接处的图片：



经过改进，处理拼接处的图片：



最终的结果，可以看出效果还是很不错的：



最后，再把上面的两张图片继续进行拼接，获得最终的输出。可以看出。整张图片是比较连贯的。但是这张图片是固定在一个地方，旋转相机拍摄的，这会导致如果以第一张图片为基准的话，之后拼接的图片会有些许模糊，不过整体效果还是不错的。

