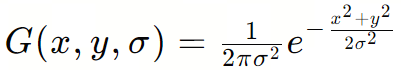
1. 实现说明

1.1关键点检测算法DOG原理

1、高斯模糊：首先，对输入图像应用不同尺度的高斯模糊。高斯模糊通过高斯核对图像进行卷积，从而模糊图像细节，保留图像的低频成分。这个过程可以用下面的高斯函数表示：



其中，

G(x,y,σ) 是高斯核函数，σ是高斯核的尺度参数。

1. 差分操作：接着，对不同尺度的高斯模糊图像进行差分操作。具体来说，对于每个尺度 σ，计算两个相邻尺度的高斯图像的差，即 G(x,y,σ) 与 G(x,y,σ/ ) 的差。这个差分图像能够突出显示在该尺度上具有明显变化的区域。

3、关键点检测：差分图像中的局部最大值或最小值点被认为是潜在的关键点。这些点在图像的局部区域内相对于其邻域具有显著的亮度变化，这使得它们对图像的几何变换（如旋转、缩放等）具有鲁棒性。

4、非极大值抑制：为了进一步精炼关键点，通常使用非极大值抑制（Non-maximum Suppression, NMS）技术。NMS移除那些不是其邻域内的局部极值点的点，从而保留最显著的特征点。

1. 尺度和方向分配：最后，为每个关键点分配一个尺度和方向，以生成SIFT特征描述符。这使得特征点对尺度缩放和方向变化具有不变性。

1.2 SIFT（尺度不变特征变换）算法

SIFT算法是一种尺度不变特征变换算法，主要包括尺度空间极值点检测、关键点定位、关键点主方向确定和关键点描述子生成四个步骤。其中，尺度空间极值点检测是通过构建高斯金字塔来检测尺度空间中的极值点；关键点定位是通过拟合高斯模型来精确定位关键点；关键点主方向确定是通过统计关键点周围的像素梯度方向来计算关键点的主方向；关键点描述子生成则是通过计算关键点周围的像素梯度方向直方图来生成描述子。

1.3代码说明

这段代码的核心是使用OpenCV提供的图像拼接器对象stitcher来实现图像拼接功能。它通过在输入图像上应用SIFT算法提取特征点，然后使用KNN算法找到匹配的特征点对，最后使用RANSAC算法估计图像间的变换矩阵。拼接过程中，它会自动根据图像间的重叠区域进行对齐和融合，以生成一个全景图像。

如果指定了裁剪选项，代码会自动裁剪拼接后的图像，去除重叠区域之外的部分，以得到一个无缝的全景图像。裁剪过程中，代码使用边界框和轮廓检测来确定拼接图像的有效区域，并将其提取出来。

最后，代码将拼接后的图像保存到指定的输出路径，并在屏幕上显示拼接结果。如果图像拼接失败，代码会输出相应的失败信息。

总结起来，该代码实现了一种基于特征点匹配和变换矩阵估计的图像拼接算法，能够自动将多张图像拼接成一个宽阔无缝的全景图像。

1. 结果截图

输入1：



输出1：



输入2：



输出2：



输入3：



输出3：



输入4：



输出4：



输入5：



输出5：



1. 运行说明

Terminal输入：

python image\_stitching.py --images images/datasets\_name --output output\_name.png --crop 1

其中`images/datasets\_name`为待拼接图像所在文件夹，`output\_name.png`为处理拼接保存后的图像；这里使用了相对路径，因为在项目根目录下运行了终端,不确定在根目录最好使用完整的绝对路径。` --crop 1`为是否裁剪黑色边框，缺省则不裁剪。