**计算机视觉与应用实践**

**实验报告**

**课程名称： 计算机视觉与应用实践**

**学院： 计算机科学与工程学院 专业： 计算机技术**

**姓名： 杜立鑫 学号:**  123106222842

南京理工大学

一、实验目标

1. 理解关键点检测算法DOG原理。
2. 理解尺度变化不变特征SIFT。
3. 采集一系列局部图像，自行设计拼接算法。
4. 使用Python实现图像拼接算法。
5. 实验原理说明
6. 特征提取(Feature Extraction)：使用关键点检测算法DOG检测输入图像中的特征点。

DOG 是一种常用于特征点检测的算法，它通过在图像中应用不同尺度的高斯滤波器并计算其差异来检测图像中的局部极值点。

这种方法通过在不同尺度下检测图像的边缘和斑点来发现图像中的关键点，因为这些区域对于不同尺度的高斯滤波器会有不同的响应。

在图像拼接中，DOG 算法可以帮助定位图像中的显著特征点，从而进行图像的对齐和匹配。

1. 图像配准(Image Registration)：建立了图像之间的几何对应关系，使它们可以在一个共同的参照系中进行变换、比较和分析。

SIFT 算法通常用于图像配准的特征点检测和匹配，通过在图像中提取特征点并计算它们的描述子，可以实现对图像间的几何关系建模。

SIFT 算法首先在不同尺度下寻找图像中的关键点，然后为每个关键点计算描述子，描述子是对该关键点周围区域的局部特征的描述。

由于 SIFT 描述子具有尺度不变性和旋转不变性，因此它在不同图像之间的特征匹配中具有很高的鲁棒性，适用于图像拼接等应用场景。

1. 图像变形(Warping)：图像变形是指将其中一幅图像的图像重投影，并将图像放置在更大的画布上。
2. 图像融合(Blending)：图像融合是通过改变边界附近的图像灰度级，去除这些缝隙，创建混合图像，从而在图像之间实现平滑过渡。混合模式(Blend modes)用于将两层融合到一起。
3. 实验步骤说明
4. 导入必要的库：从 imutils 导入 paths，以及导入 numpy 和 cv2，这是OpenCV的Python接口。
5. 构建参数解析器并解析参数：使用 argparse 库构建了一个参数解析器，用于从命令行接收输入图像目录、输出图像路径和是否进行裁剪的信息。
6. 加载图像：使用 paths.list\_images 函数获取输入图像目录中的图像路径列表，并按路径顺序加载图像。
7. 图像拼接：使用 OpenCV 的 cv2.createStitcher() 或 cv2.Stitcher\_create() 函数创建一个图像拼接对象，并将加载的图像传递给该对象进行拼接。
8. 裁剪（可选）：如果指定了裁剪参数，则在拼接完成后对拼接图像进行裁剪。这个过程包括：

给拼接图像加上一个10像素的边框。

* 将拼接图像转换为灰度图像，并进行阈值化处理。
* 找到阈值图像中的所有轮廓，并找到最大的轮廓，即拼接图像的轮廓。
* 创建一个掩膜，该掩膜包含拼接图像区域的矩形边界框。
* 使用腐蚀操作和减法操作，找到最小的矩形区域。
* 根据最小矩形区域的边界框坐标提取最终的拼接图像。

1. 将输出的拼接图像写入磁盘，并在屏幕上显示结果（如果成功拼接）。
2. 如果拼接失败，则输出失败信息。
3. 实验结果分析

输入图像：

 

输出图像：

