**图像拼接实验**

1. **实验目标**
2. 理解关键点检测算法DOG原理；
3. 理解尺度变化不变特征SIFT；
4. 采集一系列局部图像，自行设计拼接算法；
5. 使用python实现图像拼接算法。
6. **实验要求**
7. 不允许使用现成的图像拼接程序；
8. 在采图过程中可尽可能减少相机在垂直方向上的运动，但不能假设图像只在水平方向上平移；
9. 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”；
10. **实验步骤**
11. DOG原理

差分高斯（Difference of Gaussians，简称DOG）是一种图像处理方法，用于检测图像中的关键点。它是通过计算两个不同尺度的高斯模糊图像之间的差异来实现的。

高斯函数是一种在多个领域广泛应用的函数，其数学表达式为：

其中，（x，y）是坐标，是标准差，决定了高斯函数的宽度。

DOG是通过两个不同标准差和的高斯核对图像进行卷积，然后求取这两个结果的差值得到的，数学公式为：

通常，会大于，这样DOG函数就会在边缘区域产生明显的响应。

DOG在图像处理中一个典型应用是尺度不变特征变换（Scale-Invariant Feature Transform, SIFT），这是一种强大的局部图像特征描述子，广泛应用于目标识别、图像匹配、全景拼接等领域。DOG图像可以用来检测图像中的边缘和角点，这些点通常被认为是图像中的稳定特征，可以在不同的尺度下被可靠地检测到。

1. SIFT

尺度不变特征变换（Scale-Invariant Feature Transform，简称SIFT）是一种用于图像中关键点检测和描述的算法。SIFT由David Lowe在1999年提出，并在2004年进行了完善总结。SIFT算法因其在不同尺度空间上查找关键点并计算出关键点的方向的能力而闻名，这些关键点即使在光照、平移、旋转、仿射变换等变化下也能保持稳定，因此广泛应用于计算机视觉领域，如目标识别、图像匹配、全景拼接等。

SIFT的步骤如下：

①尺度空间极值检测：SIFT首先在不同尺度的尺度空间中检测关键点。这通过构建尺度空间，并对每个尺度的图像进行高斯差分（DOG）计算来实现，以发现潜在的角点和边缘；

②关键点定位：在尺度空间中精确定位关键点的位置和尺度，这通常涉及到对DOG图像进行泰勒展开，并求解极值点；

③方向赋值：为每个关键点分配一个或多个方向，通常是通过在关键点周围的局部区域内计算梯度方向的直方图，并选择主方向；

④关键点描述：生成关键点的描述子。描述子是关键点周围区域的梯度方向的统计，它对图像的旋转、尺度缩放和亮度变化具有不变性；

⑤匹配：使用关键点的描述子在不同图像之间进行匹配，以找到相同的特征。

SIFT算法具有以下特点：

①尺度不变性：由于在多个尺度上检测关键点，SIFT能够识别不同尺寸的特征；

②旋转不变性：通过为关键点分配方向，SIFT能够识别旋转后的特征；

③仿射不变性：SIFT能够识别即使在仿射变换下也保持不变的特征；

④光照和噪声的鲁棒性：SIFT描述子的设计使其对光照变化和图像噪声具有一定的鲁棒性。

1. 算法流程



**关键点检测（DoG）**：将彩色图像转换为灰度图像，使用SIFT算法创建SIFT对象，调用detectAndCompute方法来检测图像中的关键点并计算描述符。

**特征匹配：**使用Brute-Force匹配器进行特征匹配，应用比值测试来过滤不良匹配。

**图像配准与单应性矩阵计算：**根据匹配的关键点对计算单应性矩阵H，使用RANSAC算法进行鲁棒性估计。

**图像配准：**使用单应性矩阵H对image1进行透视变换，将变换后的image1和image2结合为warped\_image。

**平滑过渡：**在拼接接缝处进行平均值混合，减少接缝的可见性。

**去除黑边：**将拼接后的图像转换为灰度图像，使用阈值处理和轮廓检测来找到最小边界矩形，裁剪图像以去除黑边。

1. **结果与分析**
2. 结果

输入样例：

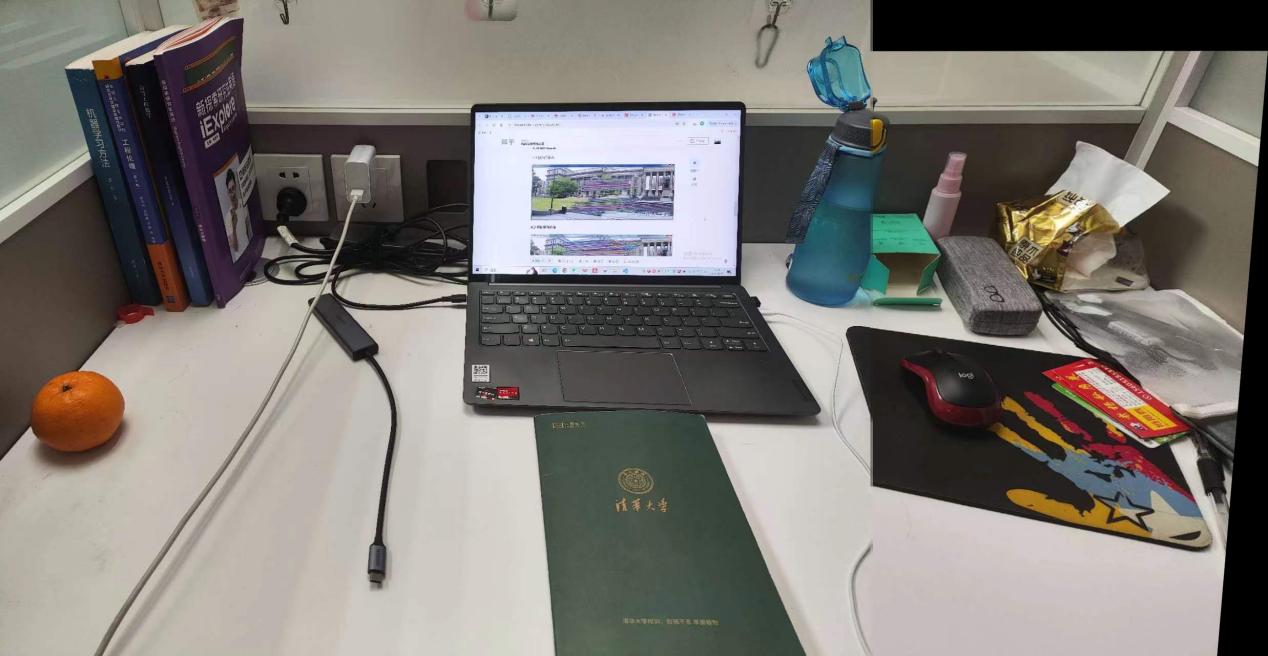
输出结果：



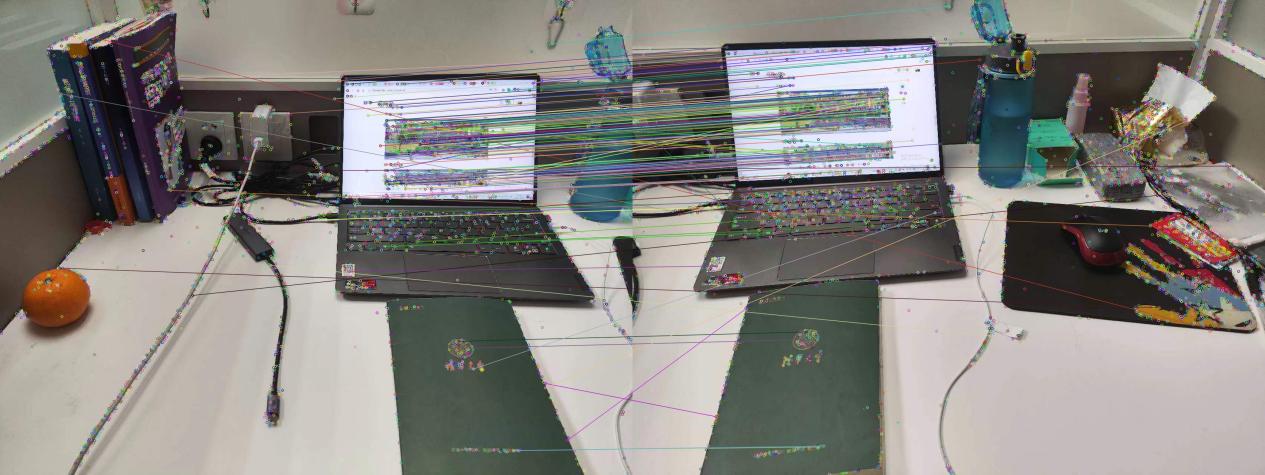
输入样例：

输出结果：



特征匹配图：



1. 分析

本次实验我采用了两种输入，第一种是图像没有垂直上的移动的，仅在水平方向上有位移，这种图像拼接的效果比较好；第二种是图像在垂直方向上有小范围的移动，不仅仅在水平方向上有位移，这种图像的拼接效果较差，边缘部分能找出特征点进行匹配，但是出现了不同程度的黑边。

对于第二种效果不好的原因可能如下：我并没有调整图像的几何形状，而是只通过透视变换进行图像拼接，后面可以使用仿射变换来调整图像的几何形状，以更好地匹配两张图像。

除此之外，对于接缝处的图像融合，我采用的方法是通过接缝处的平均值混合来实现平滑过渡，效果并不完美。除此之外，可以采用插值方法来填充和平滑变换后的像素值，减少因对齐不准确而产生的接缝，也可以使用泊松融合的方法来融合接缝处，使接缝处更加平滑。

1. **总结**

本次实验聚焦于图像拼接技术的研究和实现，目标是实现两张输入图像的无缝拼接，生成高质量的拼接图像。通过实验，我深入理解了图像拼接的核心算法和流程，以及在实际应用中所面临的挑战和解决方案。

首先，通过DoG（Difference of Gaussians）算法和SIFT（Scale-Invariant Feature Transform）算法检测和描述了图像中的关键点。这些关键点为后续的特征匹配和图像配准提供了基础。其次，特征匹配是图像拼接中的关键环节。通过使用Brute-Force匹配器和比值测试，能够高效地筛选出可靠的特征点匹配，为后续的图像配准提供准确的数据支持。在图像配准阶段，我利用匹配的关键点对计算了单应性矩阵，并使用RANSAC算法进行了鲁棒性估计，确保了配准的稳定性和准确性。接下来，在图像配准的基础上，我进一步实现了平滑过渡和黑边去除，通过这些后处理步骤，成功地提高了拼接图像的视觉质量，使得接缝处几乎不可见。

总的来说，本次实验不仅提供了一个全面的图像拼接解决方案，还涉及了计算机视觉中的多个核心概念和技术，如关键点检测、特征匹配、单应性矩阵计算、图像配准等。通过实践操作，我不仅加深了对这些概念的理解，还掌握了使用相应的图像处理库来实现复杂的计算机视觉任务。