



《计算机视觉工程实践》报告

学号：823104010011

姓名：高双猛

图像拼接

实验要求

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。

实验原理

DOG 原理

差分高斯算子 (Difference of Gaussians, DOG) 是一种在图像处理和计算机视觉中用于特征检测和描述的方法，它在 SIFT (Scale-Invariant Feature Transform, 尺度不变特征变换) 算法中扮演着重要的角色。下面是 DOG 算法的原理。

高斯模糊 (Gaussian Blur): 高斯模糊是一种常见的图像处理技术，它利用高斯函数的性质来对图像进行模糊处理。高斯模糊的作用是消除图像中的噪声，并使图像变得更加平滑。在 SIFT 算法中，首先对输入的图像进行一系列不同尺度的高斯模糊处理，产生一组高斯金字塔 (Gaussian Pyramid)。高斯金字塔是一种多尺度表示，通过在不同尺度上对图像进行模糊处理而得到的一系列图像。

差分图像 (Difference Image): 在构建了高斯金字塔后，接下来对相邻尺度的图像进行减法运算，得到一组差分图像，也就是相邻尺度的高斯模糊图像之间的差异。这些差分图像突出了图像中的高频信息，有助于寻找图像中的显著特征。

关键点检测: 在差分图像中，通过检测局部最大值和最小值，可以找到图像中的关键点，这些关键点通常对应着图像中的角点、边缘或其他显著特征。在 DOG 算法中，关键点的检测通常通过对图像中的像素点进行比较来实现，寻找局部极值点。

关键点筛选: 为了保证关键点的稳定性和可靠性，在检测到的关键点中通常会进行一定的筛选，例如通过设置阈值来剔除低对比度的关键点，或者通过非极大值抑制来保留局部极值点，剔除冗余的特征点。

方向赋值: 对于每个关键点，还需要确定其主方向。在 SIFT 算法中，主方向通常是通过在关键点周围的局部区域计算梯度方向来确定的，选择梯度方向最强的方向作为关键点的主方向。

SIFT

SIFT (Scale-Invariant Feature Transform) 是一种用于图像处理和计算机视觉中特征检测与描述的经典算法，由 David Lowe 在 1999 年提出，并在其后的论文中进一步完善和推广。SIFT 算法具有尺度不变性和旋转不变性，可以在不同尺度和旋转角度下稳定地检测出图像中的关键点，并生成对这些关键点的描述子，用于图像匹配、目标识别等应用。

以下是 SIFT 算法的主要步骤：

1. 尺度空间极值检测：

首先，SIFT 算法利用高斯模糊构建高斯金字塔。这一过程通过对原始图像应用不同尺度的高斯卷积核，生成一系列图像，每个图像在尺度上相对于前一图像具有一定程度的模糊。这种多尺度表示可以有效地检测不同尺度下的特征点。

在高斯金字塔的每个尺度上，通过比较像素点与其周围像素点及相邻尺度的像素点进行极值检测，以找到图像中的极值点（关键点）。这些关键点对应于图像中的角点、边缘等显著特征。

2. 关键点定位：

通过对检测到的极值点进行精确定位，通常使用插值技术来确定关键点的亚像素级精确位

置，以提高关键点的稳定性和准确性。

3. 关键点方向分配：

对于每个关键点，SIFT 算法通过计算其周围区域的梯度方向直方图，选取最显著的方向作为关键点的主方向。这一步骤保证了关键点的旋转不变性。

4. 关键点描述子生成：

在关键点周围的邻域内，利用关键点的主方向，构建描述子 (Descriptor)。描述子通常由关键点周围的像素梯度信息组成，这些信息可以表征关键点周围的纹理和结构特征。

5. 特征匹配：

使用生成的关键点和描述子，进行特征匹配。常见的方法是通过计算描述子之间的距离或相似度，选择最佳的匹配关系。这些匹配可以用于图像拼接、目标识别、三维重建等任务。

实验步骤

1. 导入必要的库：

首先导入了需要使用的 OpenCV 库 (cv2) 和 NumPy 库 (numpy)。

2. Stitcher 类定义：

定义了一个 Stitcher 类，包含 stitch 方法用于图像拼接，detectAndDescribe 方法用于检测和描述图像特征点，matchKeypoints 方法用于特征点匹配和计算视角变换矩阵，drawMatches 方法用于可视化匹配结果。

3. stitch 方法：

接受两张输入图像 img1 和 img2，通过 detectAndDescribe 方法检测并描述这两幅图像的 SIFT 特征点，然后通过 matchKeypoints 方法进行特征点匹配和视角变换。最后，对拼接后的结果进行图像融合，生成最终的全景拼接图像。

4. detectAndDescribe 方法：

将彩色图像转换为灰度图像，然后使用 SIFT 算法检测图像特征点并计算描述子，返回特征点集和描述特征。

5. matchKeypoints 方法：

利用暴力匹配器进行特征点匹配，筛选出满足条件的匹配对，然后根据匹配对的坐标计算视角变换矩阵。

6. drawMatches 方法：

可视化匹配结果，将匹配成功的特征点连接起来，生成显示匹配效果的图片。

7. Main 函数：

主函数读取需要拼接的图像对，调用 Stitcher 类的 stitch 方法将两幅图像拼接成全景图，并可选择是否显示匹配结果。最后将拼接结果保存为文件，显示在窗口中。

实验结果

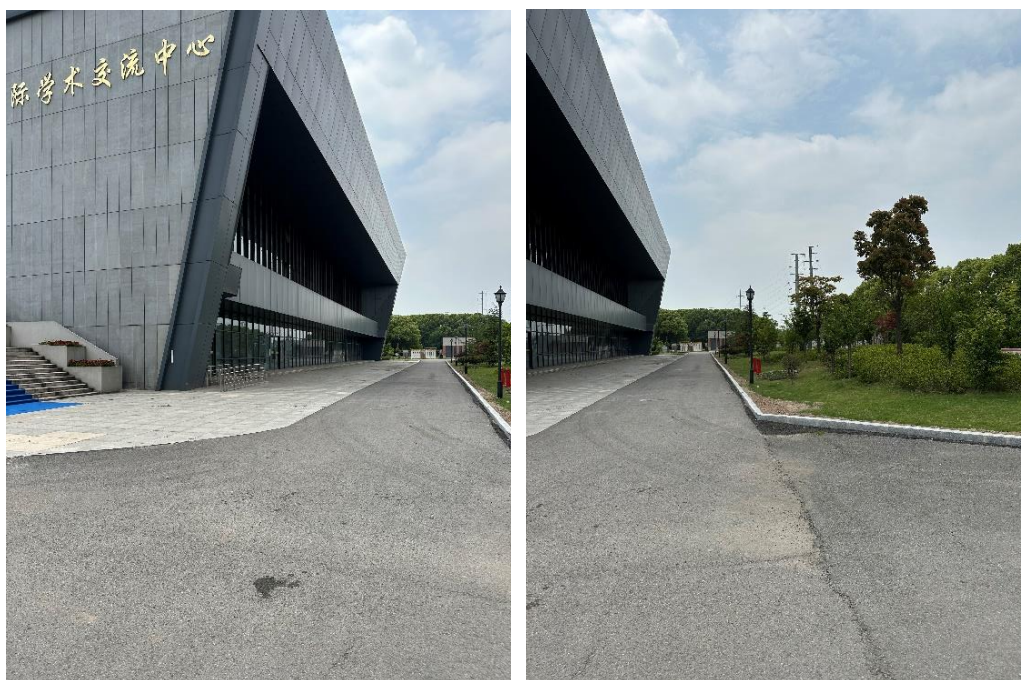


图 1. 场景 1 输入

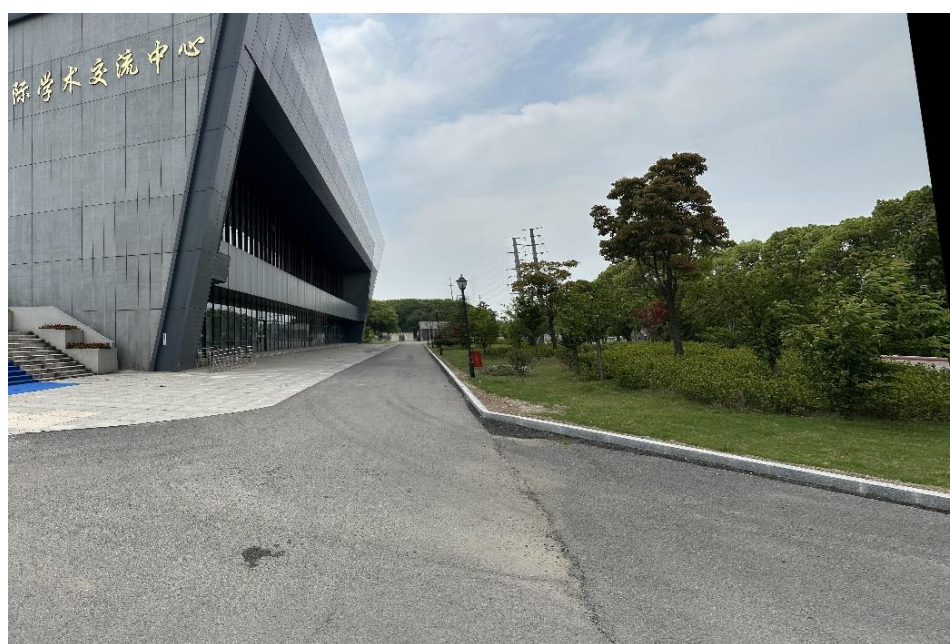


图 2 场景 1 输出



图 3. 场景 2 输入



图 4. 场景 2 输出



图5 场景3输入



图6 场景3输出



图 7. 场景 4 输入



图 8 场景 4 输出

结果分析

拼接后的图像清晰度和亮度均较好，无明显不连续的情况，细节部分的清晰度和完整，拼接处的过渡平滑，颜色、亮度等保持一致，呈现出良好的全景效果，展示出更广阔的视野，但出现了伪影和畸变后续可考虑从此优化。

实验总结

通过本次实验，深入理解了 SIFT 算法在图像拼接中的应用，提高了对特征点检测、描述与匹配的理解，为进一步学习和研究图像处理领域打下基础。

这样的报告结构能够清晰地呈现实验的目的、原理、步骤、结果与分析，以及对实验的总结与展望。