# 图片拼接与融合

## 实验说明

本实验基于Python编程语言，利用OpenCV库和Numpy库实现图片拼接和融合的操作。主要分为以下三个部分：

图片拼接函数stitch，用于将两张图片拼接在一起；

图片融合函数blend，用于对拼接后的图片进行混合处理；

主程序代码，包含图片路径、函数调用以及结果展示等。

## 代码实现说明

图片拼接函数stitch：首先，通过cv2.imread读取传入的两张图片path1和path2，并使用cv2.resize调整其大小。然后，利用cv2.copyMakeBorder对两张图片进行边界填充，以确保两张图片的大小相等。接下来，使用cv2.cvtColor将两张图片从BGR格式转换成灰度图像，使用cv2.SIFT\_create进行关键点检测和描述符提取，并利用cv2.FlannBasedMatcher进行特征点匹配。匹配结束后，筛选出好的匹配点，并使用cv2.findHomography进行单应性矩阵的估计。最后，利用cv2.warpPerspective进行图像变换，得到拼接后的结果。该函数返回两张拼接图片的变换结果。

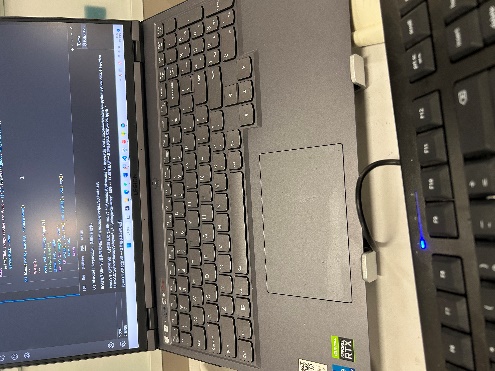
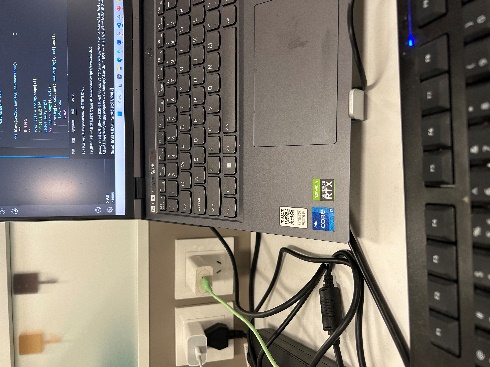
图片融合函数blend：该函数采用多频段融合的方法，对拼接后的图片进行混合处理。首先，通过cv2.pyrDown将原始图片进行下采样处理，得到一组不同尺度的图像金字塔。然后，对每一层图像金字塔，分别进行高斯模糊处理，并将拼接后的图片按照相同的方式进行切割。利用高斯金字塔上的拉普拉斯金字塔进行融合处理，最终得到混合后的图片。该函数返回混合后的图片结果。

主程序代码：主程序代码中，首先设置两张图片的路径，然后调用图片拼接函数stitch，得到拼接后的结果。接着，调用图片融合函数blend，得到混合后的图片结果。最后，展示融合后的图片结果。

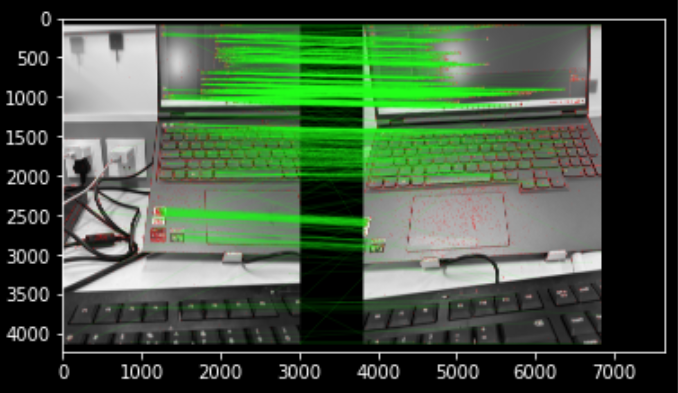
## 实验结果

本实验使用两张分辨率相同的图片进行拼接和融合，最终得到如下的结果。

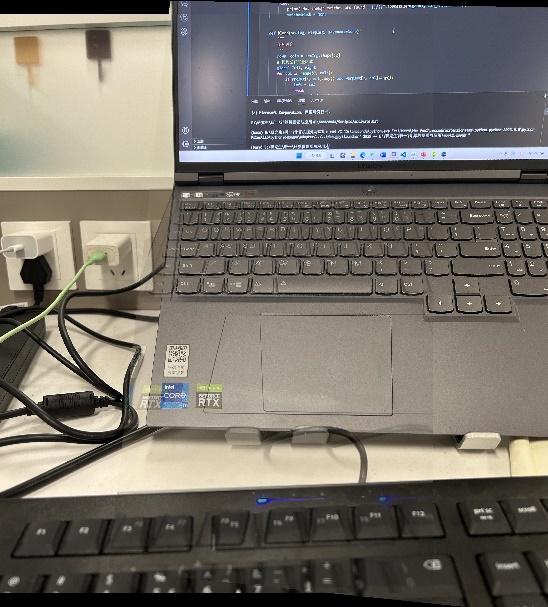
融合前图片：



图像特征点匹配：



融合后图片：



# 代码运行说明

本实验的代码使用Python编写，需要安装以下依赖库：

opencv-python , numpy , matplotlib