

计算机视觉 Ex6 实验文档

16340129 梁俊华 数据科学与计算机学院

一、实验要求

1. Take pictures on a tripod (or handheld)
2. Warp images to spherical coordinates
3. Extract SIFT features and Match features(by KNN or Hashing)
4. Align neighboring pairs using RANSAC
5. Write out list of neighboring translations
6. Correct for drift
7. Read in warped images and blend them
8. Crop the result and import into a viewer

二、实验原理

实验原理主要依据有柱状投影变换原理，SIFT 对图像的关键点进行提取，RANSAC 算法提取局内点最多的关键对，最后进行图像的拼接。

首先是柱状投影变换的原理，在平面视图上进行球面投影变换与进行煮面投影变换是相似的，因此本实验采用柱状投影变换进行，实验时，对原图像 G 上的点 (x, y) 按照下面的公式以图像的中间点 (a, b) 为中心进行投影变换，即可得到相应柱状坐标 (x', y') ：

$$\begin{cases} f = \frac{a}{2 \cdot \tan \frac{\alpha}{2}} \\ x' = f \cdot \arcsin \frac{x}{f} \\ y' = y \cos \arcsin \frac{x}{f} \end{cases}$$

进行相应的投影变换以后，就能够保持图片中空间约束与视觉的一致性。

其次要进行的是 SIFT 对图像进行关键点的提取。SIFT 的特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声

也保持一定程度的稳定性。其中用 SIFT 特征提取算法主要有以下五个步骤：

1. 尺度空间的生成；
2. 检测尺度空间极值点；
3. 精确定位极值点；
4. 为每个关键点指定方向参数；
5. 关键点描述子的生成。

本实验中没有具体实现 SIFT 的关键步骤，只是使用了开源的 **vlfeat** 库进行特征提取，生成图像间的对应关键点集。

SIFT 完成图像的特征提取以后，就需要用 RANSAC 提取局内点最多的关键点集，以用于图像的拼接。RANSAC 的主要步骤包括以下 5 个方面：

1. 随机选择 4 对关键点集
2. 根据生成的关键点集，计算单应性变换矩阵 H
3. 选定阈值 ε ，统计满足 $SSD(p'_i, Hp_i) < \varepsilon$ 的点
4. 记录从计算开始，局内点最多的关键点集
5. 对生成拥有最多局内点的最大关键点集 H ，进行最小二乘估计

在上述算法中，根据概率统计的知识，选定 RANSAC 迭代的次数为：

$$k = \frac{\log(1-p)}{\log(1-\omega^n)}$$

其中 k 表示迭代次数， p 表示算法产生有用结果的概率， ω 表示每次从数据集中选取一个局内点的概率。

在选取 4 对关键点集后，就需要计算相应的单应性变换矩阵 H ，矩阵 H 能够将第一幅图中的关键点映射到第二幅图中相应的点，它是一个 3×3 的矩阵，写成方程组的形式如下所示：

$$\begin{cases} x_1 = h_{11}x + h_{12}y + h_{13} \\ y_1 = h_{21}x + h_{22}y + h_{23} \\ 1 = h_{31}x + h_{32}y + h_{33} \end{cases}$$

若令 $h = [h_{11} \ h_{12} \ \dots \ h_{32} \ h_{33}]$ ，则经过一定的线性变换后可以得到等价求解的方程

$Ah = 0$ ，只需要对 A 进行 SVD 分解，取 V 的最后一行，即可得到 h 的结果。

在得到变换矩阵后，就进行局内点的统计，最后取局内点最多的一组作为关

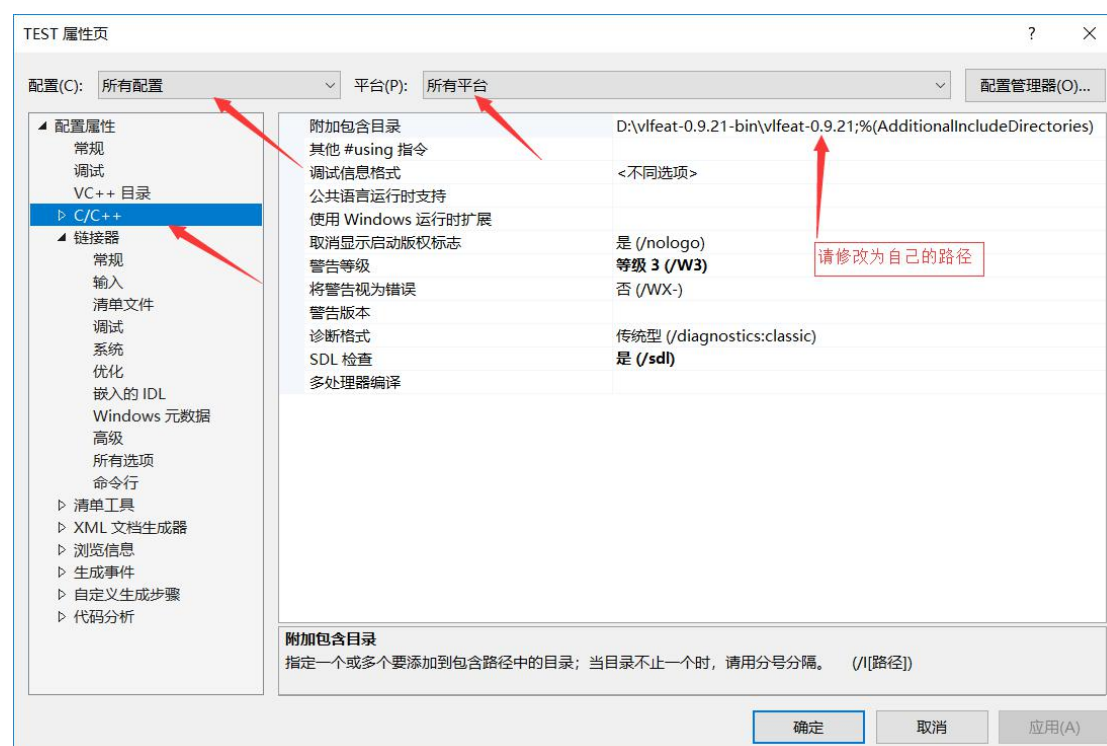
键对，并进行最小二乘法估计，即可完成 RANSAC 的过程。

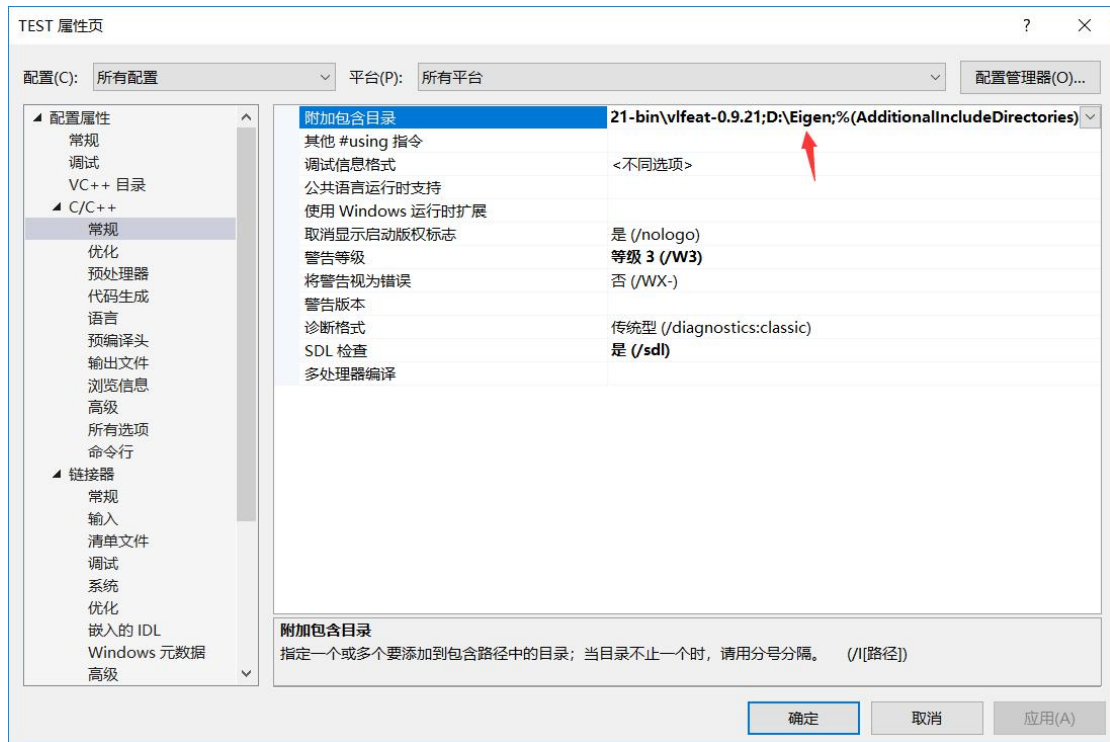
在完成 RANSAC 过程后，就对输入的相邻图像进行拼接，拼接的方法是对 RANSAC 过程中得到的局内点最多的关键点对，计算平均位移，并利用该平均位移的大小求解拼接过程中的相关参数，即可完成全景图的拼接。

三、测试说明

本实验的测试环境为 Windows10 操作系统，使用 Visual Studio 2017 进行编译，需要对 Visual Studio 2017 进行 Vlfate 库和 Eigen 库的配置，Vlfate 的版本为 0.9.21，Eigen 的版本为 3.35，测试数据分别存放在文件 TEST1、TEST2、TEST3 中。

下面说明 Visual Studio 2017 中 Vlfate 库的配置。在官网 <http://www.vlfate.org/> 中下载 Vlfate 库，然后在 Visual Studio 2017 中打开项目->属性，进行如下配置：





配置完成后即可进行运行源代码。

在执行代码时，需要手动输入两个参数，第一个参数为文件的根，例如图片 `pano1_1` 的根为 `pano1_`，第二个参数为图片的数目，若要测试 `TEST1`，`TEST2`，`TEST3`，则相应的根应当输入参考如下（需要注意文件的具体路径）：

测试 `TEST1`: `./TEST1/pano1_ 4`

测试 `TEST2`: `./TEST2/pano2_ 18`

测试 `TEST3`: `./TEST3/pano3_ 5`

四、实验结果

测试结果如下图所示：

TEST1 数据集：



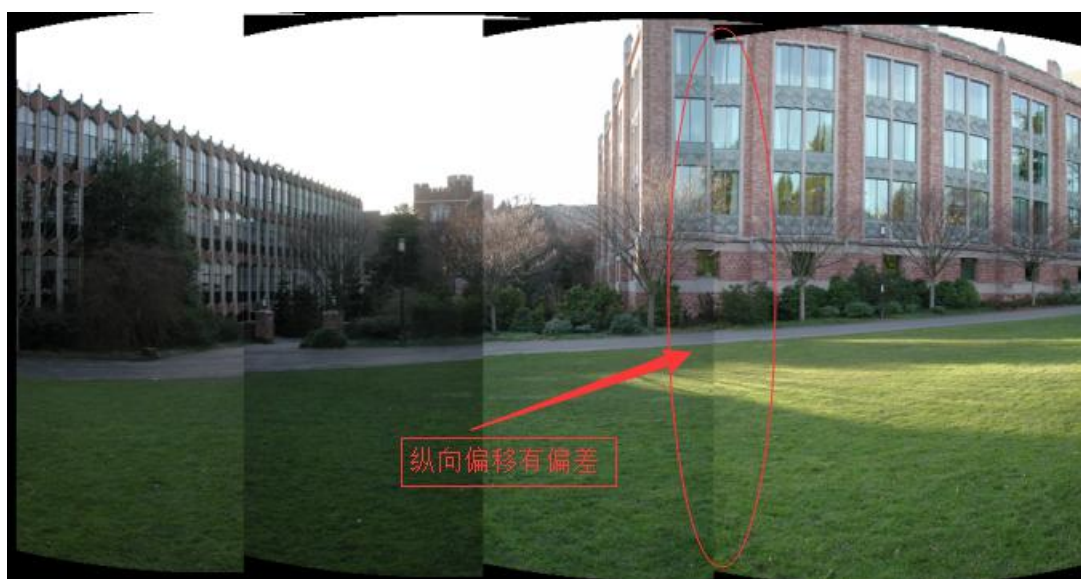
TEST2 数据集：



五、实验分析

实验结果中可以很明显的看出，拼接的效果其实只是一般，不少地方都存在缝隙，但是总体上能够完成图像的拼接。还有就是融合的结果并不是很令人满意，不同的图像之间拼接之后会存在一些融合上面的差距，这一点可以通过对两幅图像之间的像素值做一些插值处理来改善。

在进行投影变换后，会多出一些黑色的边框，这些边框其实也是挺令人讨厌的，其实也可以通过一些修补过程将其补充为完整的图像。



纵向偏移有偏差