

计算机视觉 第二次作业汇报

Implementation

容易观察到，输入给我们的图片是一个彩色图片（坑 1），所以我们要先把这个图片变成黑白的再处理。又由于变成黑白的之后数据类型还是不方便后续处理，我们将其转化为 np.array，类型为 float32/float64.

梯度

直接对转化后的图像先做高斯模糊，再做 sobel 滤波即可。根据实验结果，第一步高斯模糊的意义有限，反而会使得很多本来就不清晰的角点响应函数彻底变成 0,于是在最后的实现时直接删去了这一步。

Harris Response Function

求出梯度后，计算得到 harris 响应函数需求的 I_x^2, I_y^2 和 $I_x I_y$.为了得到 M 矩阵，我们还要对每个窗口分别对窗口内的三个函数求和。这里，我们可以发现一个优化，即这个求和等价于对图像做一个 uniform filter。最后再做一个高斯滤波即可（题目要求）。

角点选择

首先先做非极大值抑制，只需要对所有点做一个 max_filter,size 取 min_dist，其原因是在 min_dist 距离下之可能有一个点，这个点一定是局部最大值。这个改动相较于将其设定为一个较小的固定值相比，运行时间从 30s 变成了 23s.

再做阈值筛选，只保留大于阈值的响应。

再将其按照 R 的大小排序，做 KDTree 查询，就可以得到要求的角点.

HOG

先对整个图像求梯度，用欧几里得量度计量梯度的长度，用 np.arctan2(grad_y , grad_x) 计量梯度方向。

对每个关键点所在的 32*32 区域做计算，划分为 4*4 的 cell, 每个 cell 大小为 8*8，方向投影到 9 个方向上。使用 np.histogram 统计每个 cell 的方向分布作为这个 cell 的特征。

对全部 cell 的特征构成的特征做归一化。将对应角的特征加起来，取这里的最大值作为主方向。使用 np.roll 将这个主方向旋转到水平方向即可。

计算 homography

按照 ppt 上的流程计算即可。需要指出的是，我们这里的 x 和 y 坐标是反的，所以要交换一下。

RANSAC

随机选取 4 个点，计算 H 矩阵。用 H 矩阵对第一张图片的点做变换，求得的坐标和第二张图上的对应点计算距离，小于阈值则认为是正确的点。迭代 5000 次，选取最多的点再计算 H 矩阵作为真实 H 矩阵。

其余无改动

结果展示



Figure 1: 图像匹配结果 1

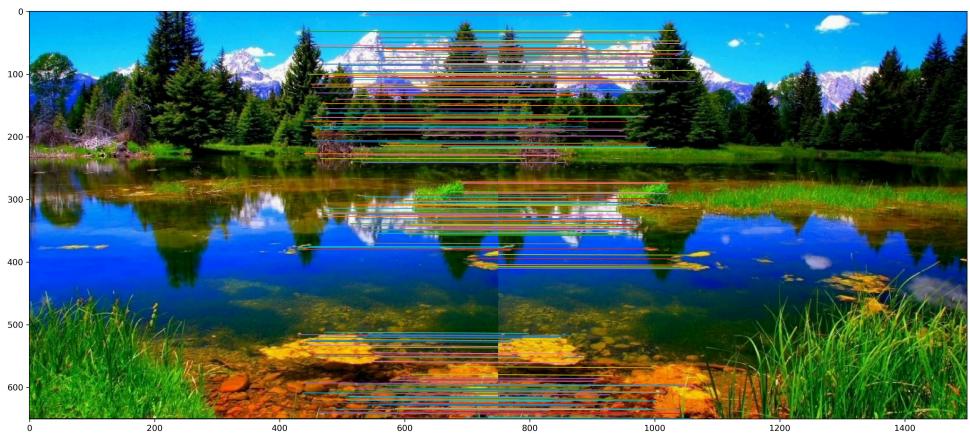


Figure 2: 图像匹配结果 2

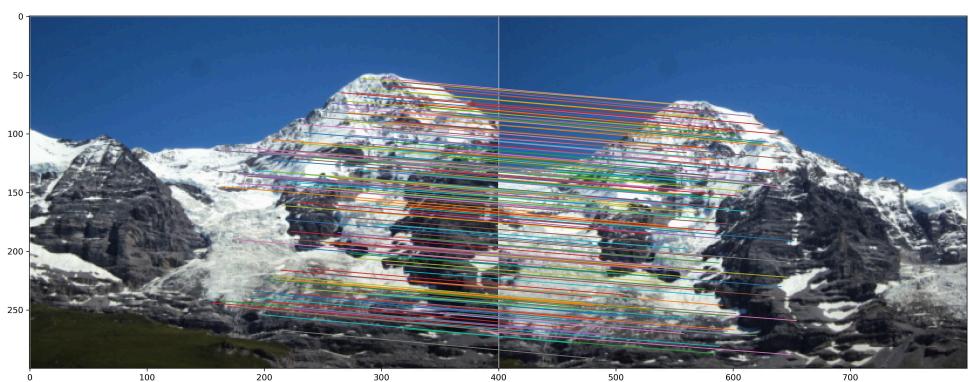


Figure 3: 图像匹配结果 3



Figure 4: 全景图 1

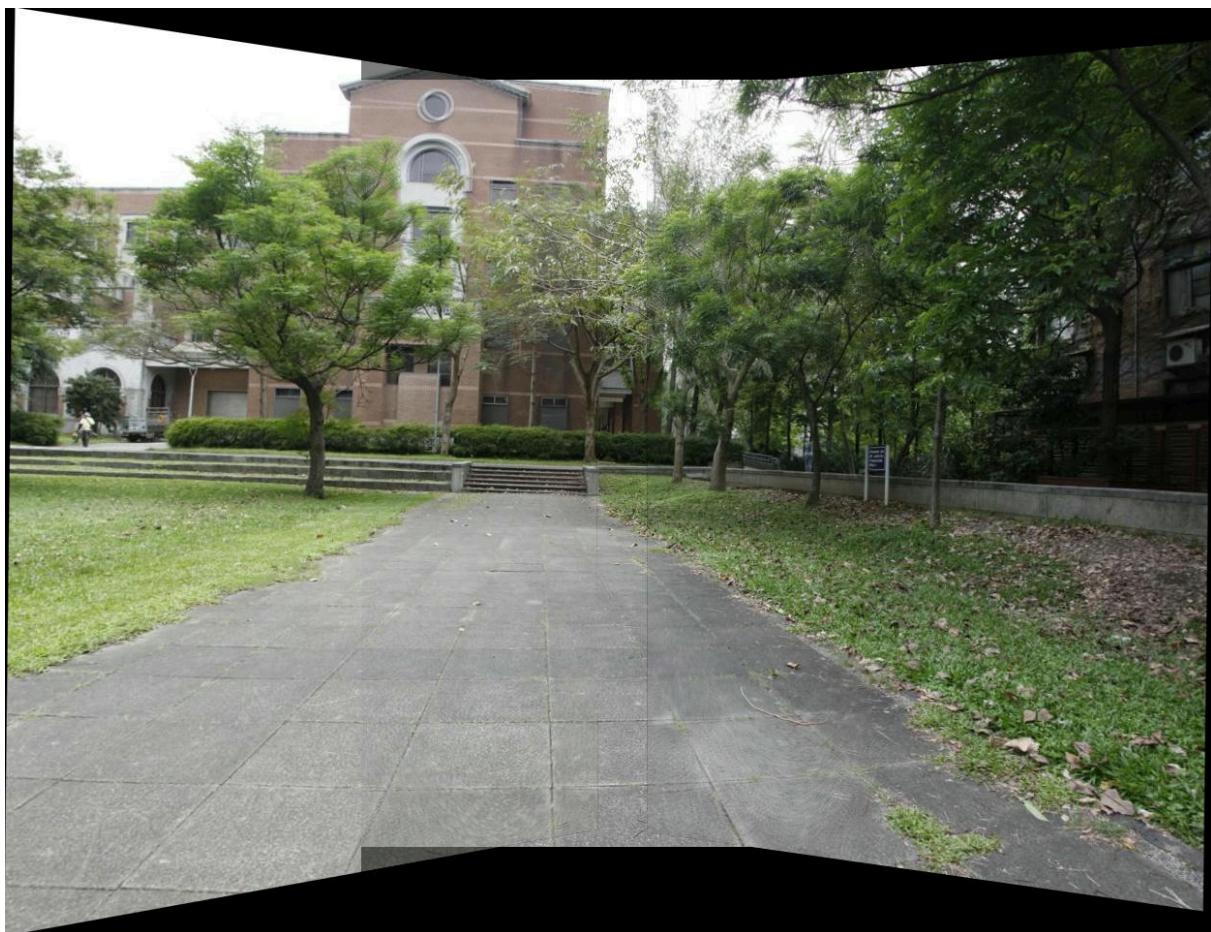


Figure 5: 全景图 2(3 张图)

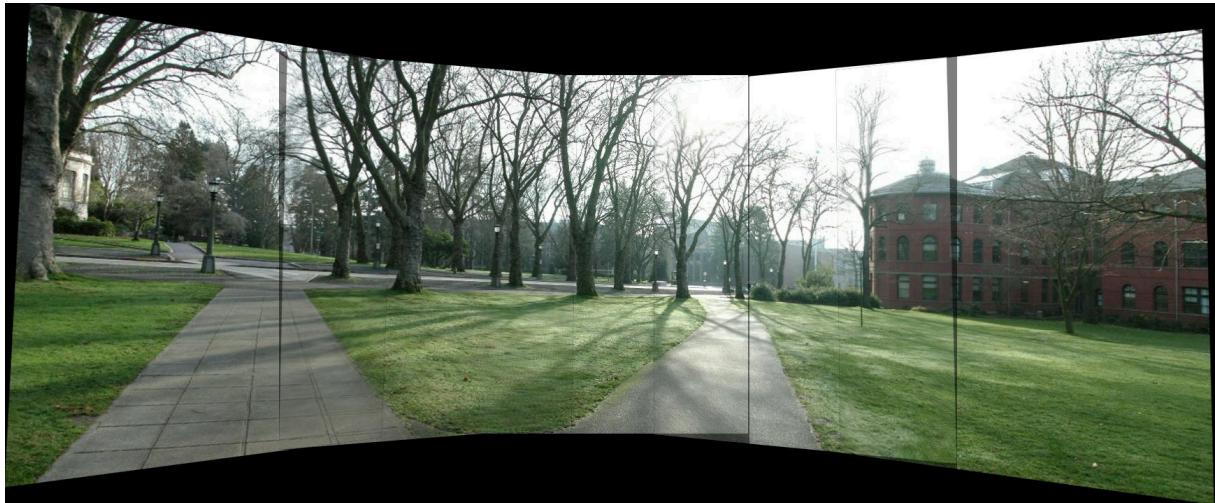


Figure 6: 全景图 3



Figure 7: 全景图 4(3 张图)

对于 library 和雪山入口，程序表现较差，可能是照相机移动时不平稳、角度变化很大，使得图像发生了很大的扭曲造成的，这导致特征匹配的失败。

分析

对于超参选择而言，我发现对结果影响最严重的超参就是在做 HOG 时选取的 cell 数量、大小。通过分别独立调整 cell size 和 block size，多次做图像拼接实验，可以发现，cell size,block size 二者较小的时候，容易出现 fail to match 错误，即无法找到足够的点做匹配，而二者很大的时候，拼图耗时过久。权衡后，我选择使用前文提及的参数进行计算。

对于相机移动而言，我通过观察图片拼接后变形的情况来判断。容易看出，library 和雪山两图在拍摄时移动不稳定，有上下移动和视角变化，这使得拼图失败。