

一、 实验目的

将一系列水平重叠的照片合成成一张全景图

二、 实验环境及分工

1. 环境: python 3.7.4
2. 提交文件中的图片的 blendwidth 均为 200, campus 结果为 360° 全景图。
3. 分工: 全部独立完成

三、 实验内容

1. 首先通过 ORB 进行特征检测, 得到特征匹配集。

2. 通过特征匹配集来估计单应映射, 其中使用了 SVD 来计算最佳拟合单应映射。通过查阅资料, 了解到 SVD 算法是在矩阵 A 不为方阵时的对特征值分解的一种拓展。目前要解决的最小二乘问题是要 $\min ||Ax||$ s.t. $||x||=1$ 。而 $||Ax||^2 = x'A'A x = x' \lambda \text{dax} = \lambda \text{dax} = \lambda \text{dax} ||x||^2 = \lambda \text{dax}$, λdax 为 $A'A$ 的特征值, 因此只要得到 $A'A$ 的最小特征值, 其对应的特征向量就是最优解。现在, 对矩阵 M 进行 SVD 分解, 有 $M*M^T = V \Sigma^* U^* \Sigma V^* = V(\Sigma^* \Sigma) V^*$; $MM^T = U \Sigma V^* V \Sigma^* U^* = U(\Sigma \Sigma^*) U^*$, 因此 $M*M^T$ 的特征向量就是 V 的列向量。

3. 利用 RANSAC 求一个最佳匹配。对于平移变换, 每次取一对点得到变换, 并通过 getInliers 获得内点数, 如果他并将内点加到列表里。对于仿射变换, 每次要取 4 对点(因为有 8 个自由度)通过 2 中实现的函数计算单应矩阵, 然后计算内点数, 选择具有最多内点数对应的单应矩阵。最终通过 LeastSquaresFit 估计一个最好变换。

4. 对图像进行融合。首先计算出变换之后图像的边框, 以确定背景的大小。之后将每一张图片通过前面得到的变换矩阵反向卷绕到背景上, 这里我使用了网格点矩阵来优化。之后将累加器 acc 和图像进行羽化之后叠加。羽化的实现是通过一个生成一个和背景一样宽的等差数列来实现权重的分配。注意到此时的 acc 有权重通道。

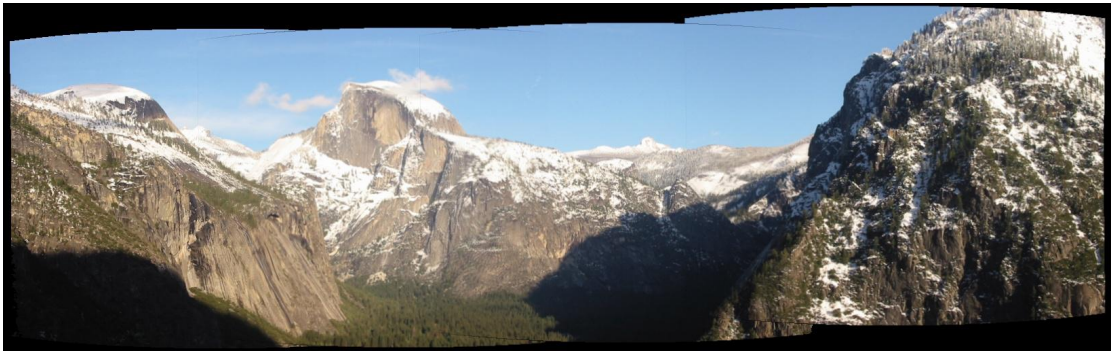
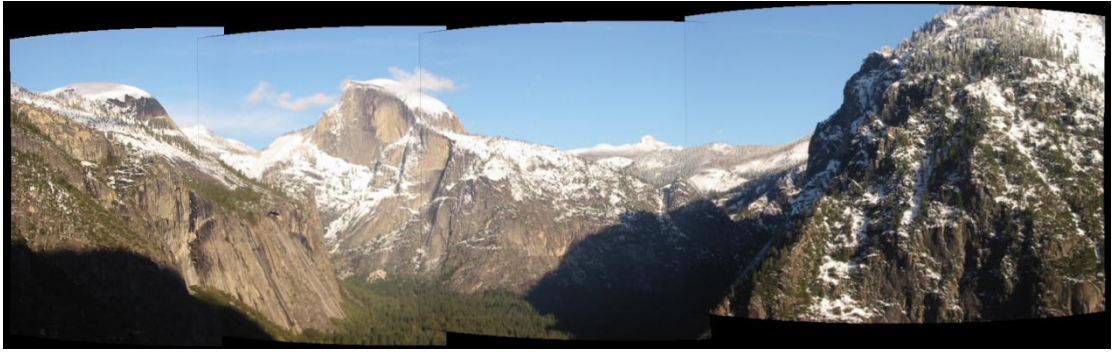
5. 将权重通道除掉, 这里注意要避免除 0 错误。

6. 实现 360 度全景图, 源文件中给了, 主要是要处理漂移问题, 方法是给每张图片增加一个 $(y_1 - y_n)/(n-1)$ 的偏移量。

四、 加分项

实现了拉普拉斯融合。高斯金字塔和拉普拉斯金字塔分别对应图像的下采样和上采样操作。将本层的高斯图像与上一层的高斯图像的上采样结果做差可得到拉普拉斯图像。图像拉普拉斯金字塔分解的目的是将源图像分别分解到不同的空间频带上, 融合过程是在各空间频率层上分别进行的, 这样就有可能将来自不同图像的特征与细节融合在一起(貌似可以在不同层上采用针对性的融合算法, 但是实现时不会做)。

通过拉普拉斯融合效果确实有提高



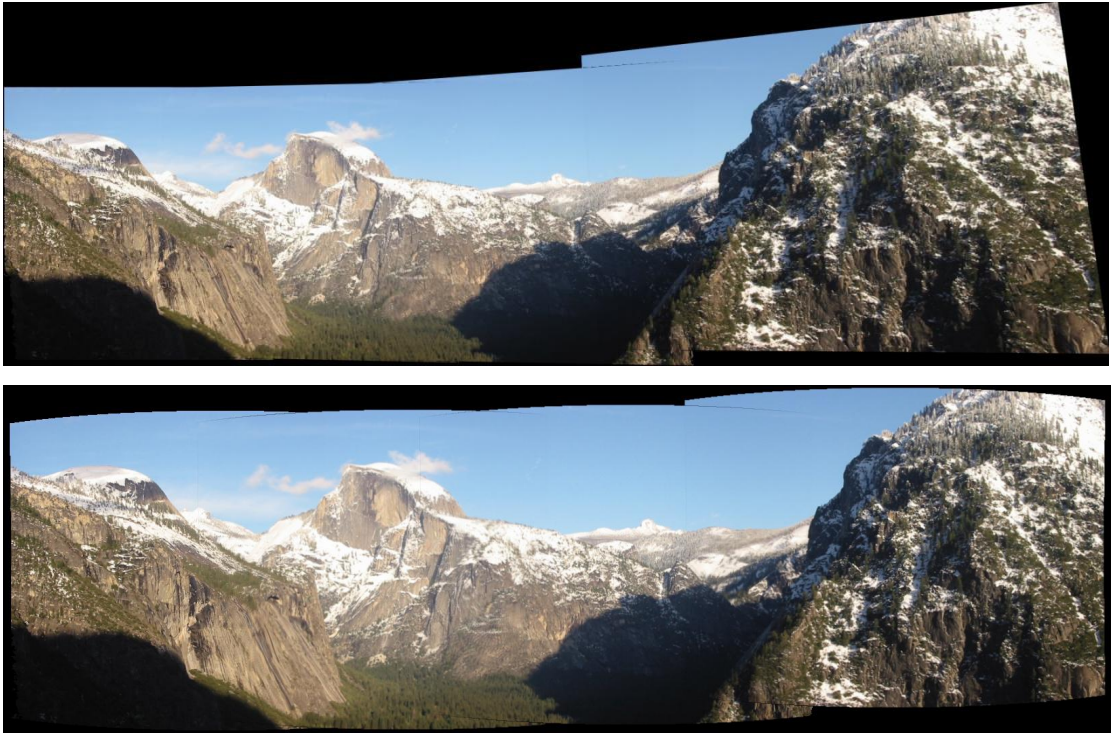
五、 实验结果

```
Unittests in test.py ×
✓ Tests passed: 8 of 8 tests – 16 ms
Testing started at 21:48 ...
C:\Users\崔恩博\AppData\Local\F
Launching unittests with argum

Ran 8 tests in 0.026s

OK
```





六、 实验心得

通过本次实验，将最近 4，5 讲内容融合在一起，加深了对只是的理解，尤其是仿射变换。实验中也遇到了很多问题，比如在使用拉普拉斯融合的时候，周围黑色的部分会变成汽油那种彩色，后来通过对金字塔融合的结果图过滤掉黑色像色就解决了。对于 SVD 还有矫正径向畸变的算法不太理解，会继续抽时间学明白。

总的来说，本次实验令我受益匪浅。

17069130005

崔恩博