

Bundle Adjustment Homework

本次 Bundle Adjustment 的作业中，为大家提供了一个简单的SFM算法框架，程序的入口在test.cpp文件中，SFM框架的system管理模块放在了visual_sfm.cpp文件中。

图像处理的核心执行模块在frontend.h和frontend.cpp中，提供了三个核心的模块

```
初始化部分
tracking 部分
特征提取和匹配部分
```

如果想要了解整体的算法流程，主要可以参考的几个函数是：

```
Init // 这个函数中，初始化了最初两张图像的相对位姿
BuildInitMap // 初始化了最初的地图点
TrackLastFrame // tracking过程中，实现了当前帧位姿求解和新的三维点的三角化
```

作业安排

本次作业有3个题目需要完成：

1. 完成 'DecomposeEssentialMatrix' 函数的功能，实现 Essential 矩阵的分解，然后根据 DecomposeEssentialMatrix求出的R和t，还有图像中的points1和points2，求解两张图像的相对位姿。

注：全局搜索 homework1 即可找到作业所在的位置。

2. 实现 Bundle Adjustment的 CostFunction 函数。

过程提示：costfunction输入是图像的位姿和对应的三维点，costfunction的类输入是图像观测的二维点，因此，残差的求解就是用三维点结合位姿投影到图像的二维平面上，然后根据观测的二维点求残差。ceres中对点求旋转可以使用ceres::QuaternionRotatePoint函数。

注：全局搜索 homework2 即可找到作业所在的位置。

3. 参考TwoFrameBundleAdjuster的 Optimize 函数，实现 BundleAdjuster的Optimize函数，实现对新加入的帧的位姿和新加入的三维点的位置进行优化。

注：全局搜索 homework3 即可找到作业所在的位置。

作业调试注意事项

作业调试过程需要注意的是：

1. 目前只给出了四张图像作为数据集，为了调试过程中方便调试，当调试homework1和homework2时，请把visual_sfm.cpp的第45行改为：

```
if(idx == 2)
```

调试homework3时改为：

```
if(idx == 4)
```

2. 编译和运行：

```
mkdir build  
cd build  
cmake ..  
make -j
```

```
./ba .../data/default.yaml
```

参考效果

