

Task3 SFM

3-0 配置与说明

从 github 网站 <https://github.com/weisui-ad/ImageBasedModellingEdu.git> 上拉取最新的代码, 注意仔细处理冲突, 不要覆盖自己本地的已有代码

3-1 运行 `examples/task3/class3_test_incremental_sfm.cc` 工程文件, 测试 `examples/data/sequence` 中的数据, 观察中间打印结果, 查看生成的稀疏点云文件 `points.ply`, 该文件位于 `~/ImageBasedModellingEud/` 目录下。

3-2 单步调试代码, 结合图 3-1 理解并掌握 Incremental sfm 的流程:

- 1) 写出增量 BA 的伪代码; `sfm`
- 2) 思考全局的 SFM 会不会有误差累计和漂移;
- 3) 增量 BA 的计算复杂度是? 写出你能想到针对增量 BA 的加速方法; `sfm`
- 4) 结合代码 `sfm::bundler::Intrinsics::compute()` 给出内参矩阵中焦距的初始化方法。

需要着重查看一下函数:

`add_exif_to_view` -- 从图像交换文件中读取图像的焦距作为初始值;
`sfm::bundler::Intrinsics::compute()` -- 进行图像之间的两两匹配;
`sfm::bundler::Tracks::compute()` -- 将所有特征匹配对转化成 Tracks;
`sfm::bundler::InitialPair::compute()` — 计算初始的匹配对;

类 `sfm::bundler::Incremental` 中主要是以下类成员函数:

`triangulate_new_tracks()`: 当有新的相机姿态被重建之后, 将会产生新的 track;
`invalidate_large_error_tracks()`: 根据重投影误差去掉质量差的 track
`bundle_adjustment_full()`: 全局的捆绑调整
`find_next_views()`: 找到新的相机视角用于重建
`reconstruct_next_view()`: 重建新找到的相机姿态(ransac_based_p3p)
`bundle_adjustment_single_cam()`: 对单个视角进行 BA

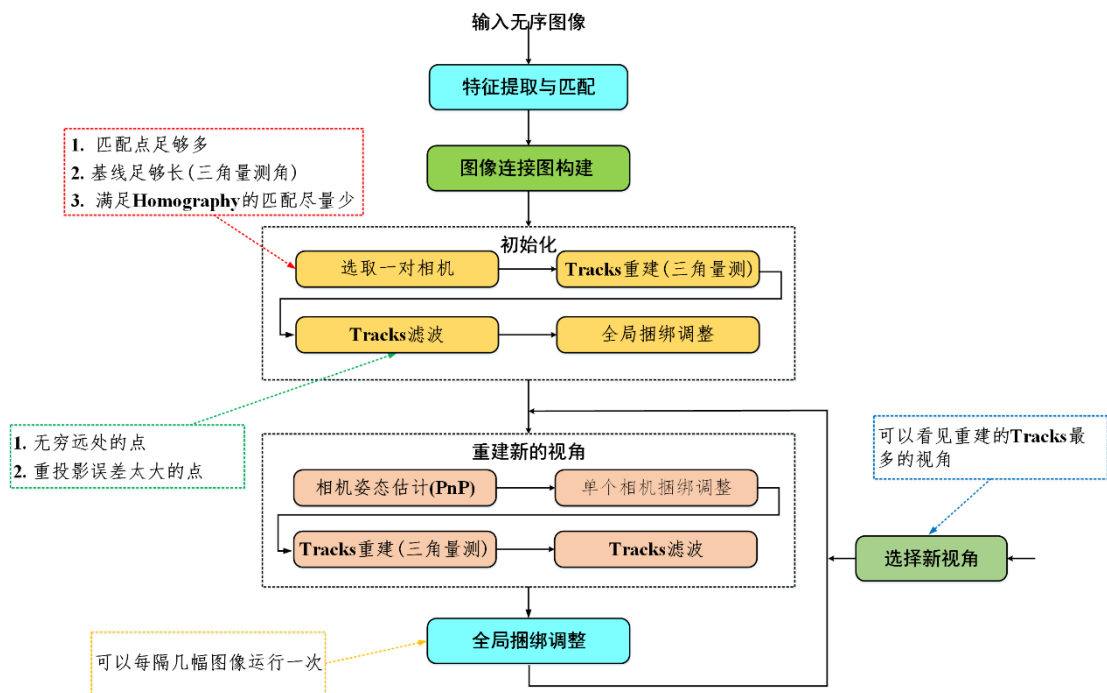


图 3-1 增量捆绑调整算法流程图（与代码/examples/task3/class3_test_incremeantal_sfm.cc 相对应）

理解这些函数，你将会对增量捆绑调整有更深刻的认识！希望大家有所收获！