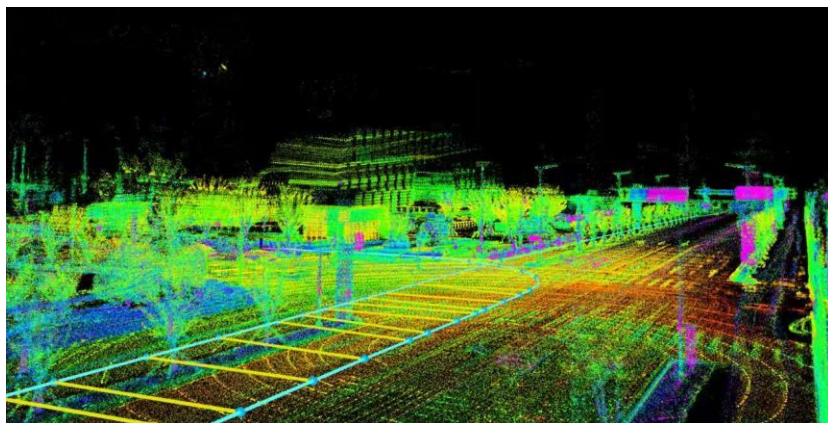




微信扫码加入星球

# 自动驾驶中实战课之Camera与IMU的融合实战

## Camera + LiDAR + Radar + IMU



主 讲 人：帅的丑小鸭

公 众 号：3D 视 觉 工 坊

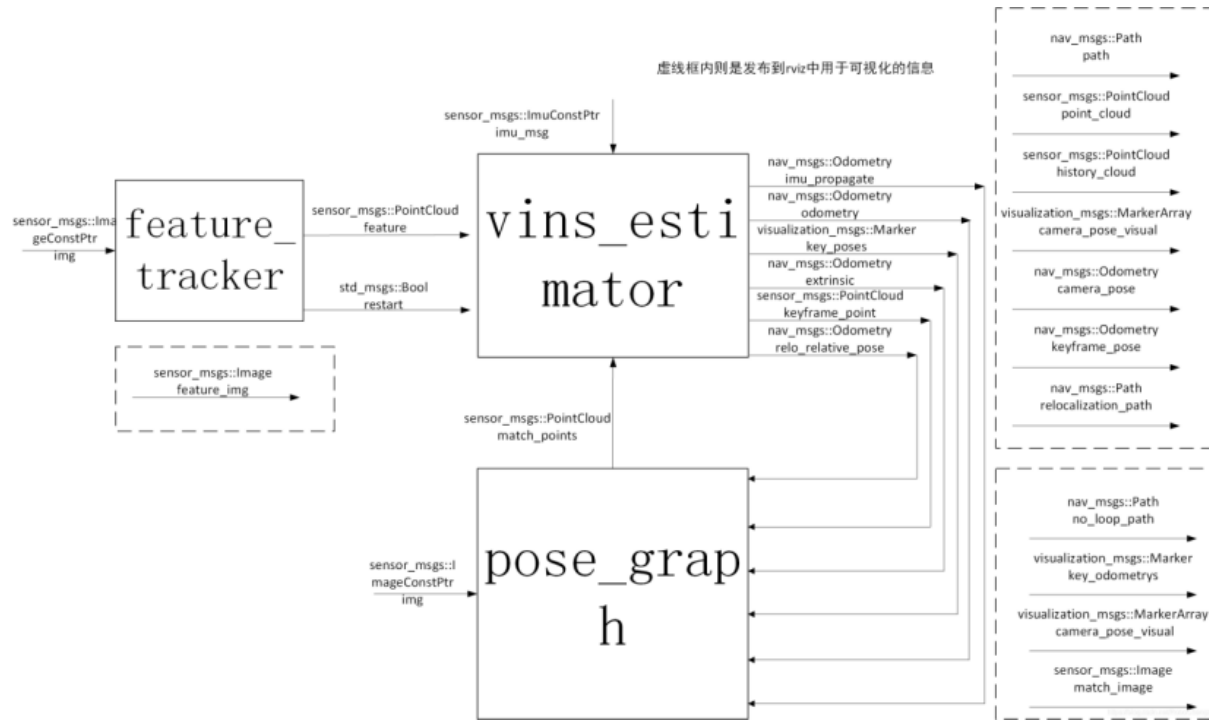
# 内容

一、Camera与IMU的融合方法

二、代码讲解



1. 采用VINS-Mono框架进行Camera-IMU的融合
  - a) VINS是基于优化和滑动窗口的VIO
  - b) 使用IMU预积分构建紧耦合框架
  - c) 自动初始化、在线外参标定、重定位、闭环检测、全局位姿优化
2. 视觉与IMU融合的经典框架
3. 离线数据工程中使用基于VINS修改的



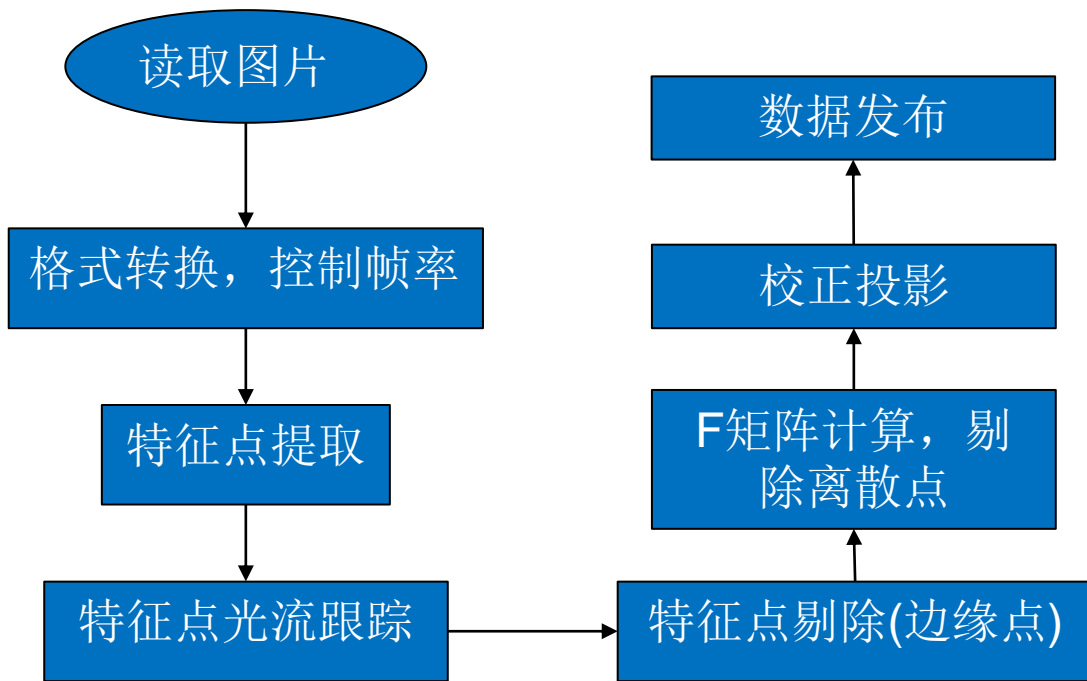
- feature\_tracker模块对图像数据进行处理，得到特征点跟踪信息
- vins\_estimator为主体程序处理跟踪得到的特征点信息和imu数据
- pose\_graph则为后端优化以及重定位功能



## FeatureTracker

核心思想为实现特征点跟踪，其流程如下：

- 获取一帧图像数据（需要按照opencv格式，并控制帧率）
- 使用goodFeaturesToTrack找到一定数量特征点
- 对特征点进行光流跟踪，在下一帧上进行匹配和校正
- 特征点发布
- 重复上述过程，前一帧到这一帧的重复

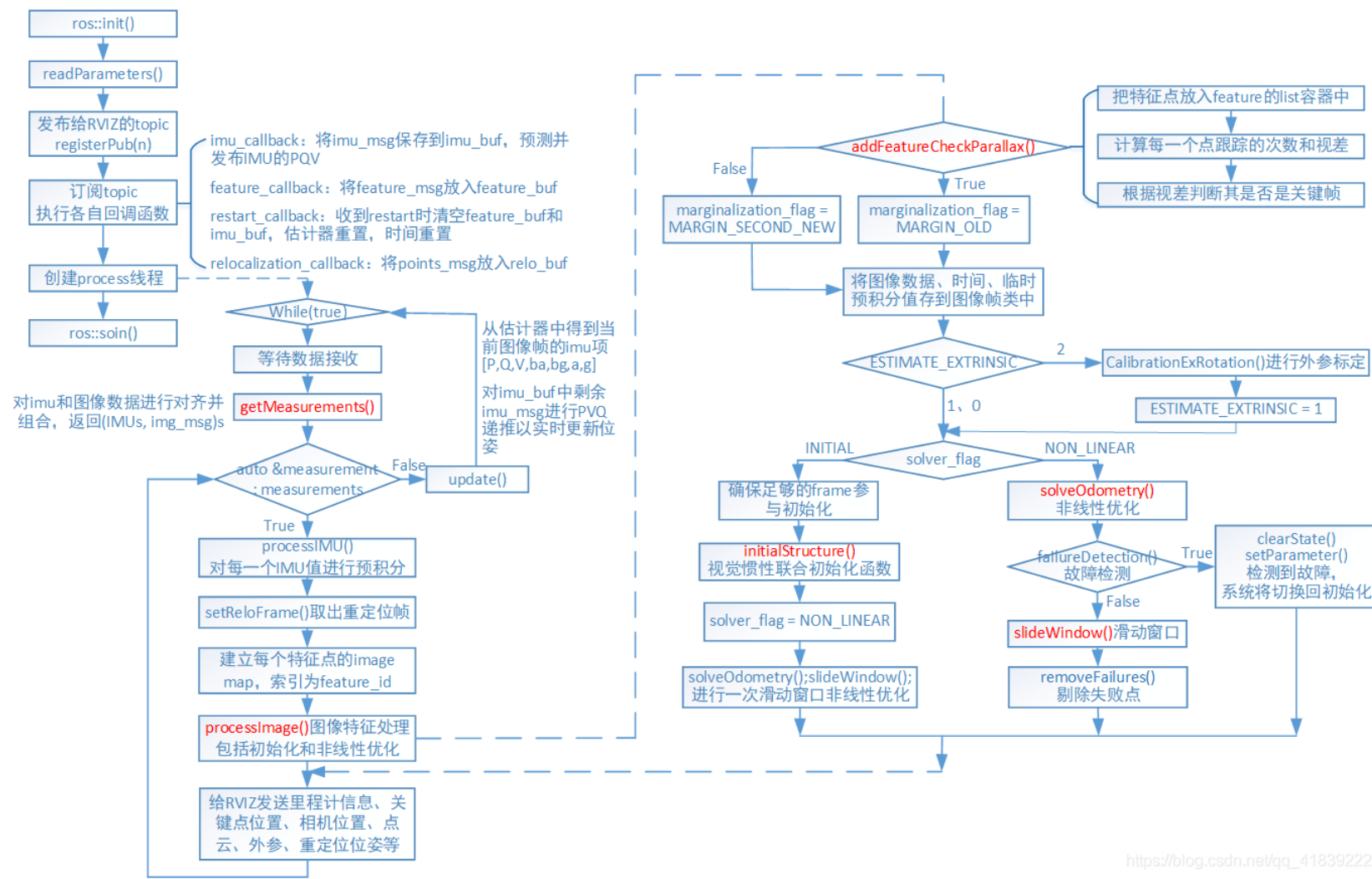


- 用`cv::goodFeaturesToTrack`在第一帧图像上面找最强的150个特征点，非极大值抑制半径为30。新的特征点都有自己的新的对应的id。
- 在下一帧过来时，对这些特征点用光流法进行跟踪，在下一帧上找匹配点。
- 对前后帧中这些匹配点进行校正。
- 先对特征点进行畸变校正，再投影到以原点为球心，半径为1的球面上，再延伸到深度归一化平面上，获得最终校正后的位置。
- 对于每对匹配点，基于校正后的位置，用F矩阵加ransac来筛选。然后再在匹配上的特征点之外的区域，用`cv::goodFeaturesToTrack`搜索最强的新的特征点，把特征点数量补上150个。
- 把剩下的这些特征点，把图像点投影回深度归一化平面上，再畸变校正，再投影到球面上，再延伸到深度归一化平面上，得到校正后的位置。
- 把校正后的位置发送出去。



## VINS\_estimator

核心功能是用来处理跟踪得到的特征点信息和IMU数据



主要函数：

- 单独线程 void process(): 主要处理模块
- IMU数据回调函数 imu\_callback: 用于IMU的接收、IMU积分更新、位置发布等
- 特征点数据回调函数 feature\_callback: 仅实现了保存数据功能
- 重定位匹配得到的数据 relocalization\_callback: 仅实现保存数据功能



## VINS\_estimator

## 函数process的处理流程

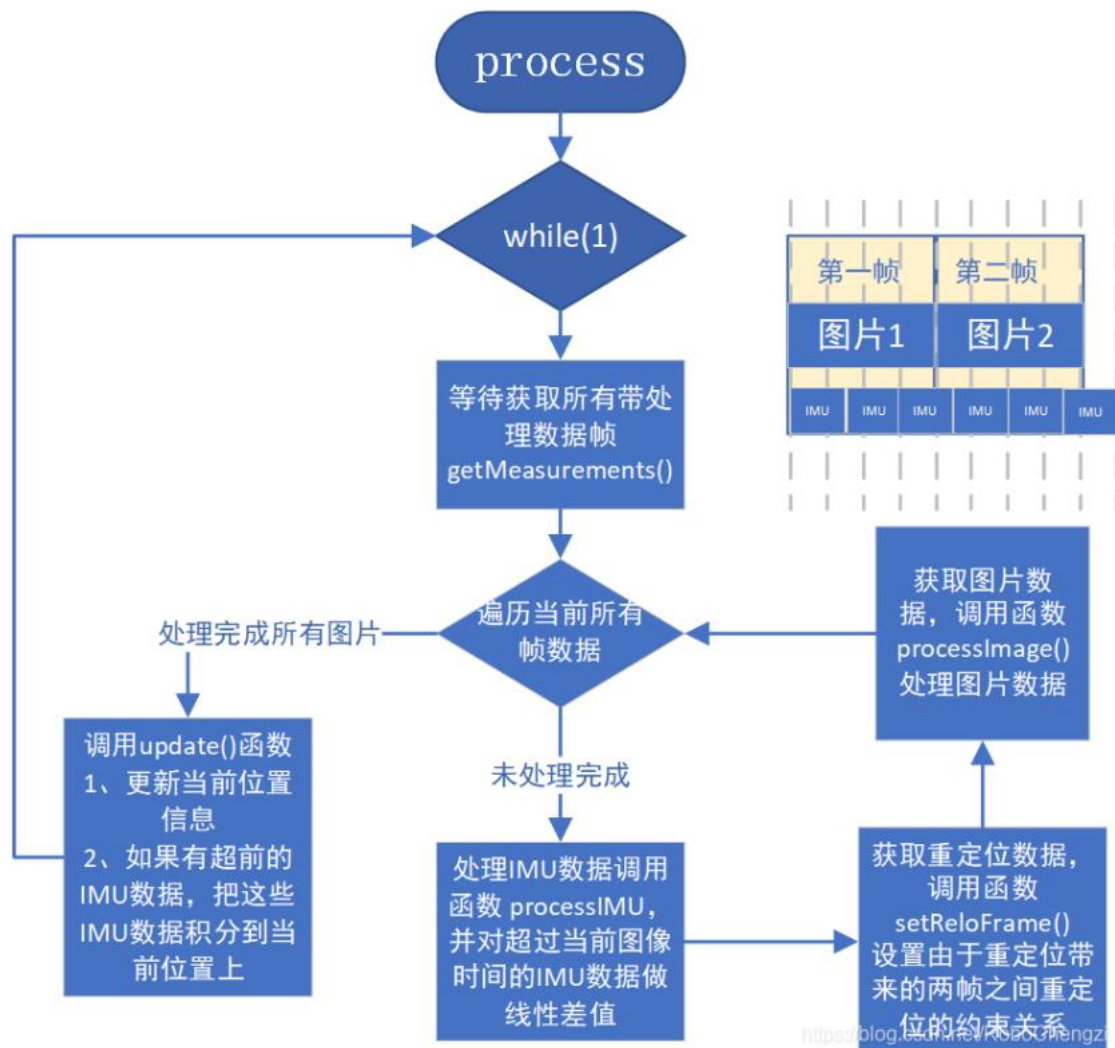
将数据按照采集时间打包成一帧图像特征点+多个IMU数据格式的多个测量

调用processIMU进行预积分计算（中值积分）

调用setReloFrame添加重定位约束（利用时间信息找到当前滑窗对应的帧信息，仅保存数据）

调用processImage处理图像特征点(是estimator中最核心、最复杂的内容之一)

结果发布

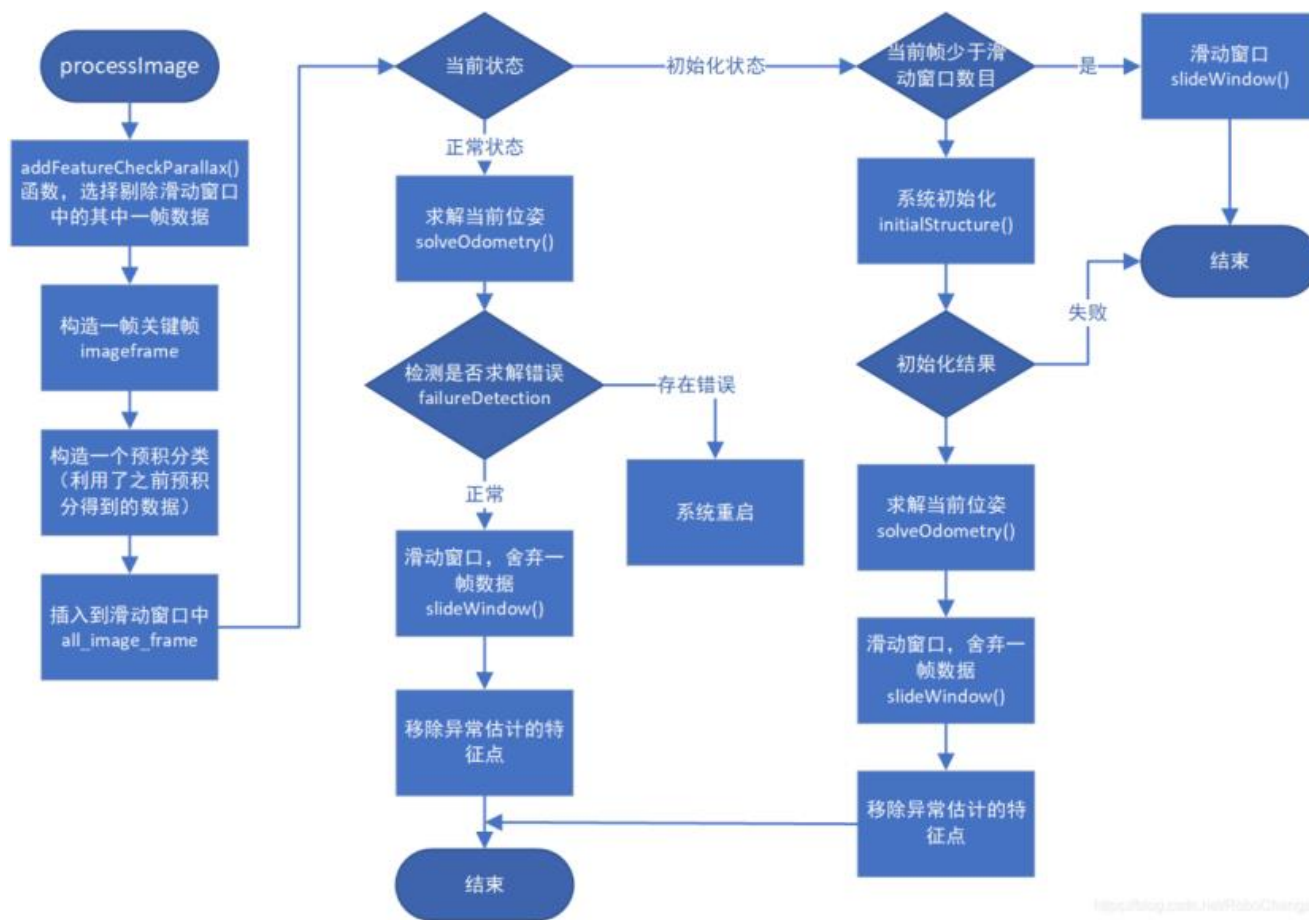


[https://blog.csdn.net/3D\\_Visual\\_Workshop](https://blog.csdn.net/3D_Visual_Workshop)



## VINS\_estimator

### 函数processImage流程图



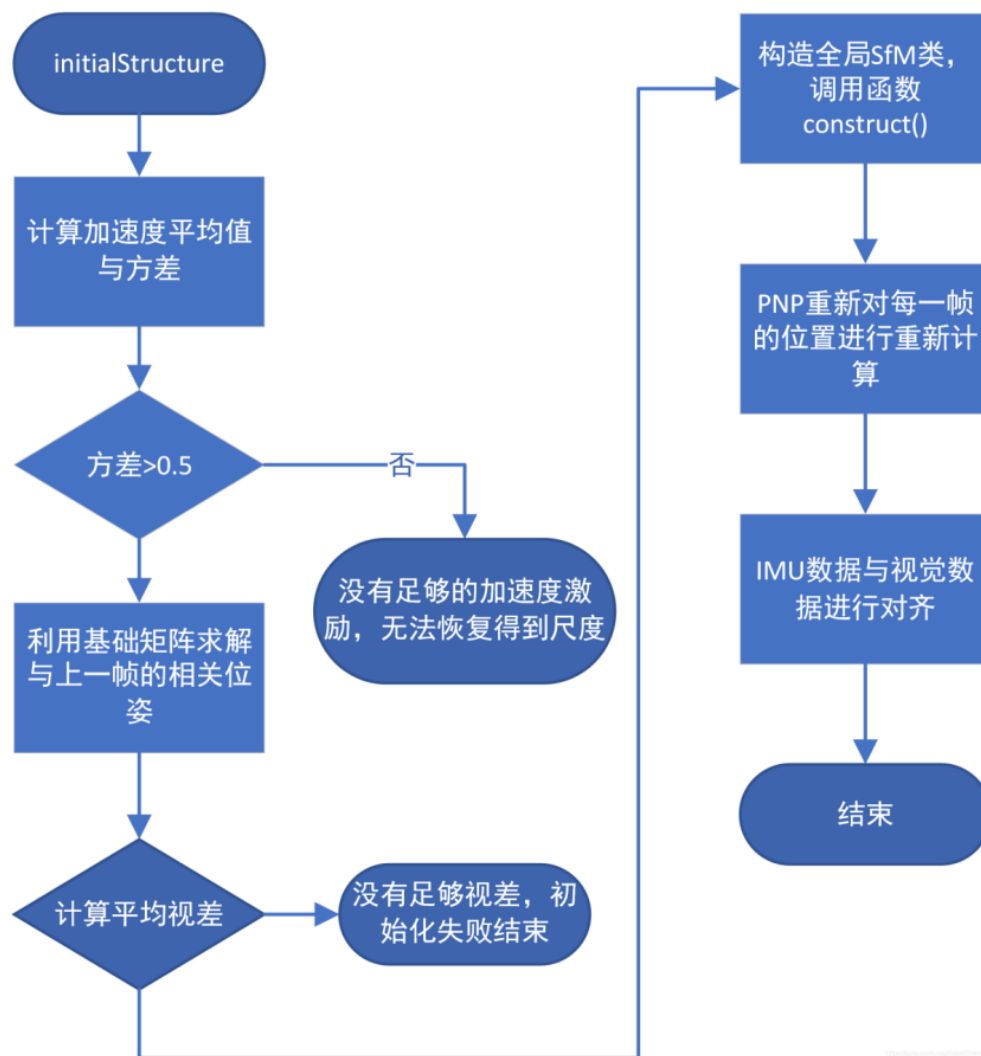
<http://blog.csdn.net/RobotChen94>





## VINS\_estimator

### 模块SfM流程图





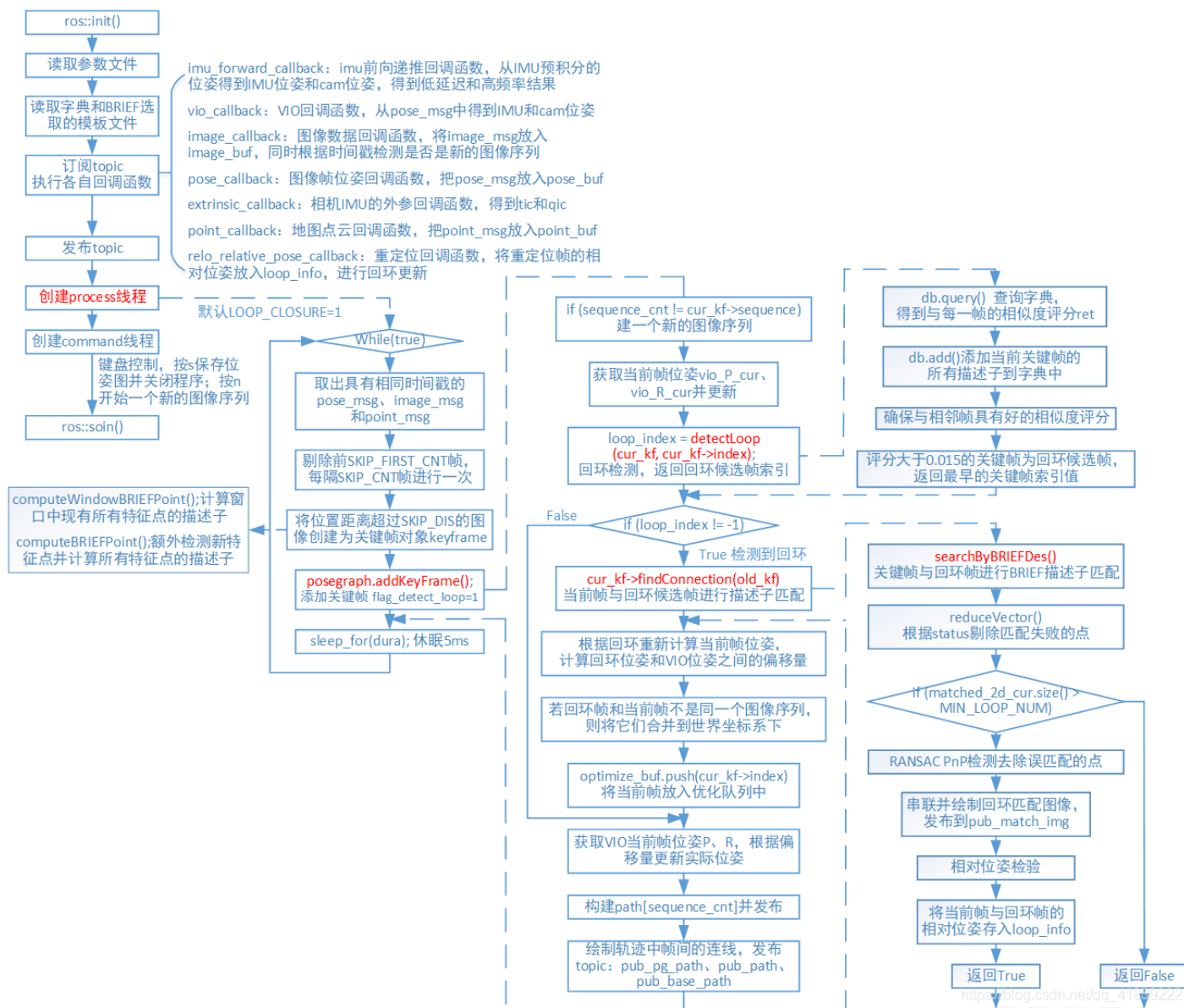


## Pose Graph

核心功能是后端优化以及重定位功能

keyframe构建关键帧类、描述子计算、匹配关键帧与回环帧

pose\_graph位姿图的建立与图优化、回环检测与闭环





代码讲解



购买该课程请扫描二维码



微信扫码加入星球



**感谢聆听**

Thanks for Listening