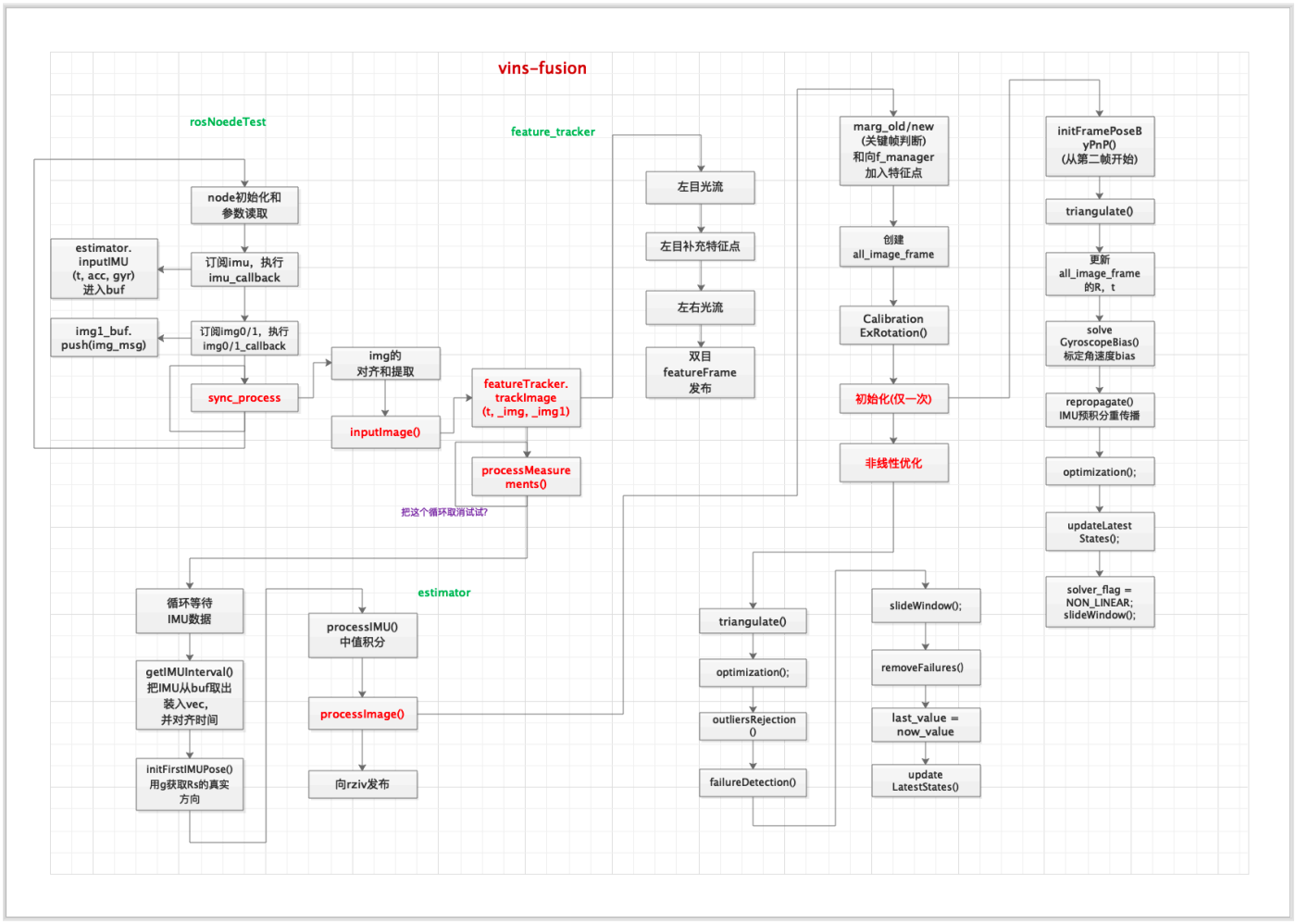
VINS-Fusion 分析

因为时间原因，没有像 vins-mono 看的和写的那么具体。
vins-fusion 不像 mono 那样有三个 node，它只有一个 node，在 rosNodeTest.cpp 里。我推测之所以这样做，是为了方便非 ros 版的 c++项目使用。先上流程图。



基本上所有的内容都在 estimator.cpp。
几个比较印象深刻的的地方。

feature_tracker 里面，
并没有对 imageConstPtr 的时间/频率校准工作，而只是放在 img_buf 里面；
单目光流跟踪加上了反向光流追踪。

数据预处理，
没有采用 messurements 这个大的数据结构，而是用了几个小的 vec 来存放着对齐的数据；
对于边界位置的 IMU 采用，放弃了被相邻两帧共享的线性插值处理；

processMeasurements：
对于状态量 Rs，mono 里面初始化是 Identity，而在 fusion 里面，初始化是用 acc 的平均值与 gw 对比就获得了相对于先验 w 系的旋转，通过这个获得 Rs[0]；

然后初始化区别非常大，没有 construct 和 IMUvisual align 那一些复杂的操作。此时 PVQ 都是 w 系的，那么左右双目的位姿也都是 w 系的，可以直接三角化双目的特征点获得特征点在 w 系上的坐标和逆深度；
之后就可以通过 PnP 进一步优化 Ps 和 Rs。

最后非线性优化一下，就结束了。

TODO 而 mono 里面，能不能也采用一样的方式呢？一开始 PVQ 都是 w 系的，scaler 也是，然后三角化不同 pose，得到一些 3D 点，然后 PnP 优化？这样是不是精度就不行了？这是不是 fusion 双目版比 mono 版精度低的理由？

optimization:
重投影误差用的是不同 camera 之间的。

```
                //左相机在 i 时刻和 j 时刻分别观测到路标点
                  ProjectionTwoFrameOneCamFactor *f_td = new ProjectionTwoFrameOneCamFactor(pts_i,
 pts_j, it_per_id.feature_per_frame[0].velocity, it_per_frame.velocity,
                                                                        it_per_id.feature_per_frame[0].
cur_td, it_per_frame.cur_td);
                  problem.AddResidualBlock(f_td, loss_function, para_Pose[imu_i], para_Pose[imu_j]
, para_Ex_Pose[0], para_Feature[feature_index], para_Td[0]);
            }

            if(STEREO && it_per_frame.is_stereo)
            {
                Vector3d pts_j_right = it_per_frame.pointRight;
                if(imu_i != imu_j)
                {   //左相机在 i 时刻、右相机在 j 时刻分别观测到路标点
                    ProjectionTwoFrameTwoCamFactor *f = new ProjectionTwoFrameTwoCamFactor(pts_i
, pts_j_right, it_per_id.feature_per_frame[0].velocity, it_per_frame.velocityRight,
                                                                        it_per_id.feature_per_frame[0].
cur_td, it_per_frame.cur_td);
                    problem.AddResidualBlock(f, loss_function, para_Pose[imu_i], para_Pose[imu_j
], para_Ex_Pose[0], para_Ex_Pose[1], para_Feature[feature_index], para_Td[0]);
                }
                else
                {   //左相机和右相机在 i 时刻分别观测到路标点
                    ProjectionOneFrameTwoCamFactor *f = new ProjectionOneFrameTwoCamFactor(pts_i
, pts_j_right, it_per_id.feature_per_frame[0].velocity, it_per_frame.velocityRight,
                                                                        it_per_id.feature_per_frame[0].
cur_td, it_per_frame.cur_td);
                    problem.AddResidualBlock(f, loss_function, para_Ex_Pose[0], para_Ex_Pose[1],
 para_Feature[feature_index], para_Td[0]);
                }

            }
```