

1.关于传感器标定

(1) . 汽车的传感器标定是不是还涉及到车身和组合导航之间的外参标定？车身的原点选在哪里？标定的方法有哪些？

车身原点一般选择在后轴中心，标定方法我会给出几篇参考文献

(2) . eth 的 lidar align 要求 imu 进行充分的旋转平移操作，这对无人机很方便，但是车辆却很麻烦。请问如何克服？或者有没有其他方法？

找一个有上下坡的地方，上下坡完成水平旋转，转向完成航向旋转

2. 多传感器标定的时候，比较平常的做法是：a 和 b 标，b 和 c 标。但实际上也可以实现 a 和 c 标，形成一个圈。这种性质能不能在多传感器标定时利用起来？有没有意义？有没有应用？

分情况来说：

情况 1：a 和 c 是要直接融合使用的。这种情况下，尽量不要通过 b 传递来标，因为标定都是有误差的，通过 b 传递一次，就等于误差被多叠加了一次。举例：a 是雷达，b 是 imu，c 是相机，目标是把雷达点云要投影到相机上，那么最好直接标定雷达和相机（前提是二者有共视）。而不是先标定雷达-imu，再标定相机-imu，最后算出雷达和相机之间的外参。

情况 2：a、b、c 均要参与融合，而且是在一个融合系统里，比如 a 是雷达，b 是 imu，c 是相机，目标是要做一个融合点云特征、视觉特征、imu 的一个紧耦合系统（雷达与相机有共视），那么外参就最

好是 a、b、c 的外参一起放在一个大的优化系统里去估计，而不是先估 ab，再估 bc，最后用 ab 和 bc 算出 ac.

情况 3: a 是雷达、b 是 imu、c 是相机，雷达和相机无共视，这种可以先标 ab，再标 bc，最后算出 ac

3.

(1) . 在【代码框架】中 IMU 预积分因子中的预积分测量值，没有像建图过程那样进行 IMU 零偏修正，这是出于哪种考虑呢。

没有零偏修正是不合理的，应该是代码框架不完善

(2) . 在 LIO-Mapping 边缘化过程中考虑了上一次的边缘化先验因子，而在课程【代码框架】中并没有添加上一次边缘化得到的因子，这是松耦合的通用做法吗。不加边缘化因子也是不对的，现在框架里问题比较多，一直在改，但一致比较忙，没改完，所以还遗留了一些问题。

另外，在学习不同开源代码时该怎么理解和应用这些细节上的差别，谢谢老师。基于上面的解释，排除不合理来不及改的地方，这些差别就不存在了。

4.老师您好，我的问题如下：

(1) 在重温代码框架过程中，我发现本课程框架读取数据时候设置了缓存的 buff 队列，使其能够不遗漏每一帧数据，但是这样做也造成了系统不能实时的处理当前数据，那么在实际系统中，如果仅考虑

缓存最后一帧（而不是第一帧）以保证系统实时性，这种做法和本课程相比优缺点是什么样的？密集的数据采样能否带来精度上的提升？只保留最后一帧是不正确的做法，等于丢弃了一些数据，会有精度损失。根据经验，在运动系统中，提高数据密集度对精度带来的收益很明显。数据密集以后，不实时问题有该问题的解决方法，比如开两个线程 A 和 B，A 线程用于在上一次观测融合后位姿的基础上做预测，对外输出位姿，B 线程用观测做融合，由于 imu 的数据实时性较好，A 线程一般是没有滞后的，所以对外输出的位姿一直是实时的，虽然 B 线程有滞后，即它输出的融合结果已经不是当前时刻的结果了，但是 B 线程可以把后验给到 A 线程去修正 A 的历史姿态，A 在此基础上重新解算，仍然能保证实时性。

(2) 在视觉 imu 系统中如果无法硬件同步，那么需要对时间戳进行同步补偿，在激光 imu 系统中是否存在同样的问题？是的 这方面的补偿思路是否和视觉近似？原理上类似 补偿后的效果是否有明显差异？效果上我还没有实测过，给不出结论

5. 第十章代码全部填完之后，运行 Rviz 发现只有红线、黄线、蓝线，没有看到绿线轨迹，可能是哪里的问题？

通过这个简单描述，我无法给出问题的原因，可以给一些调试的建议：看绿线轨迹是谁输出的，输出之前做了哪些操作，在每一步操作后面输出 log，通过 log 信息来定位是哪一步的问题。

6. 闭环检测中，两次到达的点，除了车辆位置相近，是不是对位姿也有什么要求，如果两次位姿成 90° 甚至航向是反的*(即 180°)，是不是点云匹配的残差优化中难以找到最优解。闭环的检测和修正与航向无关，可以考虑从其他方面找找原因。仍然建议通过打 log 的方式定位问题。

7. 老师您好，我是一个直博的二年级新生，在论文方向上有问题想请教任佬。导师想让我第一篇论文从组合导航 (GNSS+IMU) 入手，后续再融合视觉的内容，但是我在查文献的时候感觉 GNSS+IMU 这个方向目前做的人并不多，开源代码好像也不如激光和视觉这么多。想请教一下任佬目前 GNSS+IMU 这个方向怎么样，有什么可以写文章的点吗？另外在实际工作中会需要自己写 GNSS+IMU 吗？而且还有点担心 gnss 和视觉都搞的话，最后哪个也做不深...

- 1) GNSS+IMU 这个方向做的人非常多，论文也非常多，但是可创新的点已经不多了，属于很成熟的领域了(在工程上，虽然国内厂商在这一点上跟国外还有差距，但多数是工程问题，不是理论创新问题)
- 2) 基于以上原因，不建议做单独的 GNSS+IMU 二者融合的方案，不容易出成果，在实际工程中，确实有部分厂家在自己写 GNSS+IMU 融合模块，但这属于工程实现问题（为了成本或自助可控性），对学校学生来讲，意义不大
- 3) 建议直接搞 Camera+GNSS+IMU 的方案，不过 GVINS 已经做

了，创新点需要再考虑考虑，比如酌情加更多传感器(lidar、论速计之类的)，目前包含 lidar、camera、imu 三者的紧耦合方案在学术上还有可挖掘空间，可以考虑。

8.

(1) 最近遇到基于 tcp 分布式 ros 的一个问题是 rosmaster 所在树莓派可以自启动一些程序但是其他非 rosmaster 的树莓派不能启动一些程序，会报错没有找到 rosmaster，增加了延时也没有效果。对于分布式 ros 有没有什么资料学习呢？或者有没有相关的论坛呢？

(2) 关于 ros info 的一个问题是，用 ssh 登陆树莓派启动程序后拔掉网线，ros info 会阻塞（主要表现在里面有一个记录时间的 txt，拔掉网线后的时间没有记录），插上网线又恢复。

这方面没有任何经验，说也是瞎说，就不误导人了，抱歉！

9.能否请老师分享一些工作心得、项目 debug 的经验呢？

做项目最最重要的核心原则是“问题导向”，具体是指

- 1) 要加入什么方法取决于发现什么问题，而问题来自于实验。常见误区是看到或者想到什么方法，觉得可能是好的，就往里面加。
- 2) 基于 1)，工程的首要问题是多多的测试，而且是有量化评测的测试。比如，测一个 slam 的效果，正确的做法是找一个真值(比

如一个组合导航), 去定量评测一个量化的指标出来, 以该指标为导向, 去进行效果的迭代。常见误区是, 只肉眼看它的效果, 虽然能发现一些问题, 但是不严谨, 容易受各种主观因素影响。

关于 **debug**, 我目前唯一使用的方法就是写 **log**(从以前回答程序调试问题时的建议也能看得出来), 而且平时写代码就要养成在代码中关键步骤处加 **log** 的习惯, 这样有问题可以很容易发现。(不建议单步调试, 在工作中, 多是在远程服务器上跑, 而且很多事多线程, 单步调试的作用及其有限)

如果时间充裕, 就多些 **UnitTest**, 可在根源上大量减少 **bug** 的产生, 但是写这个很费时间, 一般在学校没精力弄, 在工程领域好弄, 堆人就行。