链接：https://zhuanlan.zhihu.com/p/34995102

VIO中，目前多采用流形空间上预积分的方法对IMU数据进行预处理，核心思路是在两帧之间计算IMU的帧间运动增量，在迭代优化时直接使用运动增量，提高计算效率。这部分参考论文《On-Manifold Preintegration for Real-Time   
Visual-Inertial Odometry》，整篇论文推导的思路是先推导出流形空间上的运动状态的表达公式，然后为了能够整合到最大后验估计的优化框架里，分离出运动表达式中的噪声项，使其近似满足高斯分布，之后推导了在bias变化时的运动估计值的更新方法。

前面有提到VIO用到了以IMU作为控制输入的运动方程，Joan Sola 写的《Quaternion kinematics for the error-state Kalman filter》对以IMU测量值为运动输入的运动方程的误差递进形式进行了详细推导，如果在vinsmono和okvis的运动递进方程的雅克比矩阵推导遇到困难，或者对四元数方法进行运动描述不了解，可以仔细阅读下这篇文章。文章对坐标系的locally和globally的解释也让人印象深刻。

VIO的初始化是系统工作非常关键的部分，这部分可以参考vinsmono以及ORB作者写的的VIO文章《Visual-Inertial Monocular SLAM with Map Reuse》。两篇文章思路比较相似，先是通过单目运动估计的方法获取多帧图像的位姿，然后以此为运动参考估计其他参数，整个过程和相机IMU标定比较相似。初始化主要完成三部分工作：1. 为非线性估计系统提供一个运动初值 2. 估计尺度、IMUbias 3. 估计重力向量在视觉坐标系下的投影向量，以此将视觉坐标系对齐到世界坐标系下。

vinsmono的代码中协方差矩阵递进公式采用的是中值积分的方法，推导思路和Joan Sola文章中的一致，具体推导可以参考<https://www.zhihu.com/question/64381223/answer/255818747>，此外有篇博客也对vinsmono的整体框架做了整理总结[https://blog.csdn.net/u012871872/article/details/78128087](http://link.zhihu.com/?target=https%3A//blog.csdn.net/u012871872/article/details/78128087" \t "_blank)。

vonsmono代码中的融合部分是非常值得学习参考的，但是代码中的视觉处理部分多是直接使用OPENCV的函数，而且代码风格是C++/C 混合的。所以推荐看下ORBSLAM的代码，首先编程实现非常规范，对编程学习有很大的参考价值，而且整个代码对opencv的依赖较低，视觉部分的特征提取，视觉运动估计，都是作者自己编程实现的，对理解视觉几何的实现有很好的帮助，泡泡有篇公开课对ORBSLAM的代码进行了梳理([https://pan.baidu.com/s/1c1QOoHM](http://link.zhihu.com/?target=https%3A//pan.baidu.com/s/1c1QOoHM) 密码: xfjd)。网上相关的博客也很多。

3.3 相机-IMU标定

相机IMU标定的目的是获取两个传感器坐标系之间的空间关系和数据延迟，是VIO系统工作的前提工作。相机-IMU标定可以看成状态估计的逆过程，标定是通过标定板获取每个时刻的精确运动状态，计算出模型参数（坐标系间旋转位移、时间延迟、IMUbias），而运动估计则是在已知两个传感器坐标系间的模型参数，估计每个时刻的运动状态。

目前有现成的标定库可以使用(/kalibr/wiki/Camera-IMU-calibration )，我写了一个使用方法总结，在资源链接里。

此外，相机-IMU标定需要事先知道相机和IMU的内参；做SLAM中我们也经常需要对新的相机进行内参标定。推荐ROS自带相机标定工具，可以实现在线标定。使用方法我写了一个文档，在资源链接里。IMU内参即陀螺仪加速度计的噪声参数，Kalibr 也给了一些说明。

数据集

定位误差有ATE和RPE两种评测指标，TUM给了评测估计值和基准值误差Python脚本，而且对两种误差的解释，TUM的评测脚本是假设估计轨迹和真实轨迹都是实际物理尺度的。ORB作者改写了一个自动缩放尺度的脚本，可以用来评测单目SLAM运动估计效果。