知乎 首发于 C++、CV、SLAM小白入门

论文学习——VINS-Mono：一种鲁棒且通用的单目视觉惯性系统

yikang
THU硕

已关注

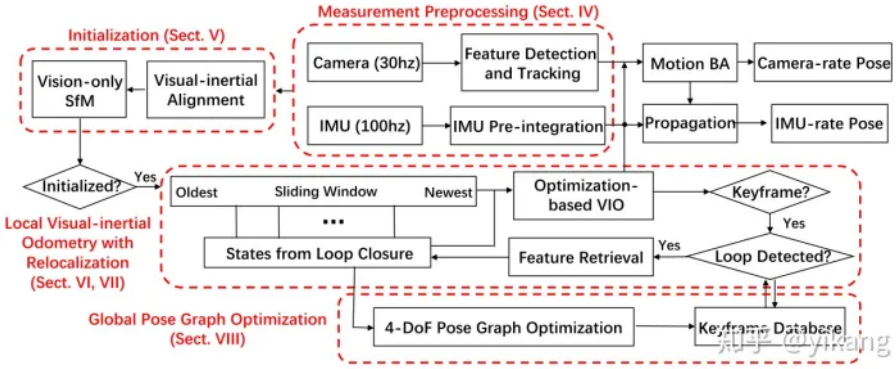
26 人赞同了该文章

[忙完毕设后的第一更，首发公众号“视觉部落”，未经允许请勿转载]

一、基本信息

本文提出了一种基于紧耦合滑动窗口非线性优化方法的单目视觉-惯性系统，来自港科大沈老师实验室。这篇**论文的亮点包括**提出了效果最佳的IMU预积分理论、估计器初始化机制、故障检测和复原机制、外参在线校订、基于优化的紧耦合VIO、重定位机制以及全局位姿图优化模块等内容。论文开源地址：HKUST-Aerial-Robotics/VINS-Mono 。

二、整体框架



VINS的整体结构框图如上所示。整体流程和各个模块的作用可以理解概括为：

1. 第一步是系统框图上方中间的Sect.4，这一模块中进行了相机和IMU测量数据的处理，包括**相机图像的特征提取和追踪，两帧图像时刻之间IMU数据的预积分以及选取关键帧。**可以看到图中注明了相机和IMU数据采集频率相差较大，因此下一步需要将相机和IMU数据进行对齐。
2. 第二步是框图上方右侧的Sect.5，这一模块进行了视觉惯性数据的对齐，通过对齐操作可以得到后面非线性优化部分需要的一些数值：相机位姿、速度、重力向量、陀螺仪偏差和路标点位置等。通过视觉惯性对齐，相机坐标系到世界坐标系(东北天坐标系)之间的关系就已知了，从而可以将相机坐标系中的轨迹(pose)对齐到世界坐标系，并且可以根据IMU的预积分值获得单目视觉不可观的尺度信息。**因此可以这样理解，在初始化部分IMU的作用包括：对齐世界坐标系、获得尺度信息。**

(marginalization)，被marg的数据将为下一时刻的窗口提供先验信息，也就是把信息向前进行传递而非直接丢弃。而marg旧帧的策略也有专门的部分进行介绍。VIO模块紧密融合了IMU预积分、图像中的特征观测以及闭合回路中的特征重检测。

4. 第四步也是最后一步是框图下方的全局位姿图优化模块，接收来自上一步的重定位结果，并且**执行4自由度（三维平移和偏航角yaw）的全局图优化**，目的是消除这四个自由度在长时间内存在的漂移。这一模块还同时维护了一个关键帧集合，这与VO的backend部分策略较为相似。
5. 值得总结的是上面提到的全局图优化可视为backend，第三步的VIO/重定位可视为frontend，因此就像之前笔者介绍VO的时候讲过的，可以使用多线程同时运行前端与后端，本文中也是这么做的。也就是**使用多线程同时运行第三步和第四步。**

三、各模块细节

本节将按照整体框架的结构分别介绍各个模块的内容。

1.测量预处理 measurement preprocessing

总的来说这一部分主要针对视觉测量和IMU测量进行处理：**对于视觉图像，在最新的图像帧中提取特征，在相邻两帧图像之间进行特征追踪；对于IMU数据，对两帧图像之间的时间内产生的数据进行预积分。**并且IMU测量数据是同时受到bias和高斯白噪声影响的，在进行预积分时会考虑bias的影响。

（1）视觉图像处理

- 特征检测采用角点特征，每帧图像有最低的特征数量(100个，过少将会影响后续的特征追踪)，相邻特征之间设置最小像素间隔(避免特征点集中)，使用基于基础矩阵F的RANSAC算法剔除异常点。
- 特征追踪使用KLT稀疏光流算法。

跟模块还负责选取关键帧，关**键帧选择标准有两个**：

- 平均视差标准：如果被跟踪特征的平均视差介于当前帧和最新关键帧之间，并且超过某个阈值，则将当前帧设为新的关键帧。由于纯旋转情况可以引起视差但是不能进行三角化，因此使用陀螺仪的短期积分进行旋转补偿以解决这个问题(仅用于关键帧选择)。
- 跟踪质量：如果跟踪的特征数低于某个阈值，将当前帧设为新的关键帧。这是为了避免特征轨迹由于特征点过少而丢失。

（2）IMU预积分

首先给出IMU测量值的数学模型：

$$\hat{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{b}_{a_t} + \mathbf{R}_w^t \mathbf{g}^w + \mathbf{n}_a$$
$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t + \mathbf{b}_{w_t} + \mathbf{n}_w.$$

其中加速度计和陀螺仪的噪声项都设为高斯白噪声，bias误差模型为随机游走，其导数为零均值高斯分布。

根据积分关系，对IMU的测量值在世界坐标系内进行积分可以得到位置、速度和旋转四元数在任意时间间隔内的值。末了避免IMU积分的重传递，采用了预积分协方差传播的策略，关于预积分部分的公式推导要去看原文中的对应部分，值得学习。注意预积分得到的结果都是以本体系/相机系为参考坐标系的。

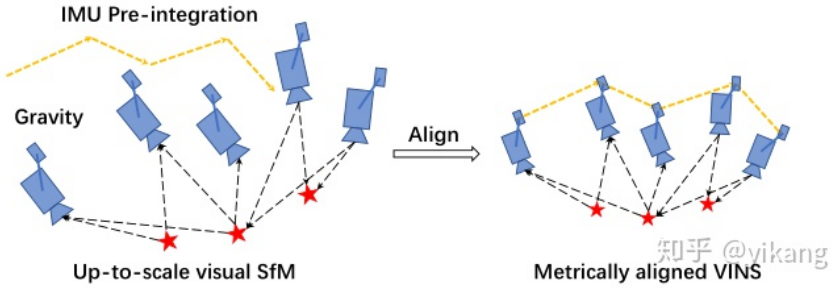
当对bias的估计发生变化时，本文提出使用两时刻间的协方差传递矩阵对预积分结果进行修正，而不是重新计算预积分的值，k时刻和k+1时刻的协方差矩阵为：

$$\begin{bmatrix} \hat{\boldsymbol{\beta}}^{b_k}_{b_{k+1}} \\ \hat{\boldsymbol{\gamma}}^{b_k}_{b_{k+1}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{b_k}_{w}\left(\mathbf{v}^{w}_{b_{k+1}} + \mathbf{g}^{w}\Delta t_k - \mathbf{v}^{w}_{b_k}\right) \\ \mathbf{q}^{w^{-1}}_{b_k} \otimes \mathbf{q}^{w}_{b_{k+1}} \\ \mathbf{b}_{a b_{k+1}} - \mathbf{b}_{a b_k} \\ \mathbf{b}_{w b_{k+1}} - \mathbf{b}_{w b_k} \end{bmatrix}.$$

2.估计器初始化 estimator initialization

初始化通常是单目VINS系统中最脆弱的一步。实验条件下当然可以在静止状态下进行初始化但是在实际使用中显然运动条件下的初始化才是更经常遇到的。因此**本部分解决的是如何在未知运动的条件下就对VINS系统进行初始化。**

首先**具体一点来说明通过坐标系对齐来获得初始值的思路**：在之前分享的文章中我们已经知道视觉SLAM有着良好的初始化特性，我们可以根据相邻帧间的相对运动获得相机位姿等参数，因此在SfM 的基础上通过**对齐IMU预积分结果和视觉SfM结果**，可以大致获得尺度、重力、速度甚至bias偏差。这种数据融合方式属于松耦合。该对齐过程的示意图如下所示：



本文在初始化阶段忽略了加速度计的bias偏差，这是因为和重力的量纲比起来bias偏差实在是太小了以至于可以融合在重力向量中。

（1）视觉 SfM 的滑动窗口算法 Sliding Window Vision-Only SfM

首先通过视觉SfM来获得去尺度化的位姿和路标点组成的图。过程大概如下：

1. 检查最新帧和之前所有帧之间的特征对应关系：如果能在滑动窗口中找到稳定的特征跟踪（超过30个被跟踪特征）和足够的视差（超过20个旋转补偿像素），就使用五点法恢复这两帧之间的相对运动；如果没有稳定的特征跟踪和足够的视差，就把当前帧留在滑动窗口中并等待下一帧进来。
2. 若运动恢复成功，就对两帧中共同观测到的所有特征进行三角化，尺度任意。
3. 有了第二步的三角化结果，就可以对窗口中的其他帧进行PnP求解（3D-2D）。
4. 最后使用全局BA对所有的观测到的特征的重投影误差，优化变量是相机位姿和路标点位置。（其实BA问题应该是我们非常熟悉的内容了）
5. 在上面的步骤中，我们已第一帧中的相机系为参考坐标系。这是为什么呢？因为我们还没有引入IMU来进行相机系和是世界系的对齐呗，这里也就体现了如果没有IMU来测量重力向量，那我们就没有办法把相机的轨迹对齐到世界坐标系（一般就是东北天坐标系）中。

在引出下一部分的视觉惯性对齐内容前，先提一嘴尺度因子 s，假设说我们获得了IMU和相机之间的外参 $\boldsymbol{p}_c^b$ 和 $\boldsymbol{q}_c^b$ (也就是相机系到本体系的平移和旋转)，那么我们就可以把相机系下的参数变换到本体系中（非常显然）：

$$\begin{aligned} \mathbf{q}^{c_0}_{b_k} &= \mathbf{q}^{c_0}_{c_k} \otimes \left(\mathbf{q}^{b}_{c}\right)^{-1} \\ s\bar{\mathbf{p}}^{c_0}_{b_k} &= s\bar{\mathbf{p}}^{c_0}_{c_k} - \mathbf{R}^{c_0}_{b_k}\mathbf{p}^{b}_{c}, \end{aligned}$$

而其中的尺度因子 s 就是下一部分对齐时获得的，解决这个参数可是能否成功初始化的关键（毕竟啊，引入IMU的一个出发点就是向解决尺度不可观的问题，而这个问题的解决就在初始化部分了啊（严格来说后面的优化中其实还有，但是初始化部分的尺度因子是非常关键的））

这一部分又分成了四个部分，分别概括一下进行的工作：

1. 校准陀螺仪bias偏差 b_w：通过对IMU预积分关于陀螺仪bias偏差的雅可比进行线性化，以及最小化损失函数，可以获得 b_w 的初始校准，并用此 b_w 来更新预积分中的平移、速度和旋转项。
2. 对速度、重力向量和尺度因子进行初始化：根据相机和IMU在物理上是**刚体连接的约束**，我们可以得到速度、旋转等几何约束，根据这些约束以及视觉SfM中恢复的相机运动，可以 得到窗口中每个帧的速度，视觉参考帧中的重力矢量，以及尺度因子。
3. 重力矢量细化：一般来说某个区域的重力大小是已知的，因此重力矢量的模长是已知的，这就构成了一个约束，利用这个约束可以是重力矢量的自由度变为2，由此就可以对重力矢量进行参数化（**这里其实是很巧妙的**，这个参数化在绝对重力矢量的基础上加入了两个正交的分量，然后通过迭代计算两个分量进而对重力矢量进行细化）。
4. 完成初始化：重点来了！利用上一步得到的精细化之后的重力向量和世界坐标系中的绝对重力矢量进行旋转重合，**由此得到相机系到世界系的变换关系，这就完成了对齐。**按照此对齐关系，把相机坐标系中的速度也变换到世界坐标系，同时把轨迹按照尺度因子向公制单位进行缩放。至此就完成了初始化，对齐后的量就可以喂给紧耦合的VIO部分了。

我觉得还是稍微总结一下上面这几个步骤的思路：首先在校准陀螺仪bias偏差 b_w后对预积分量进行更新（使之更精确了），然后利用IMU和相机的几何约束获得重力向量、速度和尺度因子的初始值，然后在对其中的重力向量进行精细化（**因为重力向量是连接本体系和世界系的桥梁**，被自己的这句总结文艺到了哈哈），然后再根据重力向量重合这样的条件使得相机系到世界系旋转对齐，把速度和尺度因子也用上，这就完成了初始化部分的工作。

3.基于优化的紧耦合单目VIO模块 tightly-couple monocular VIO

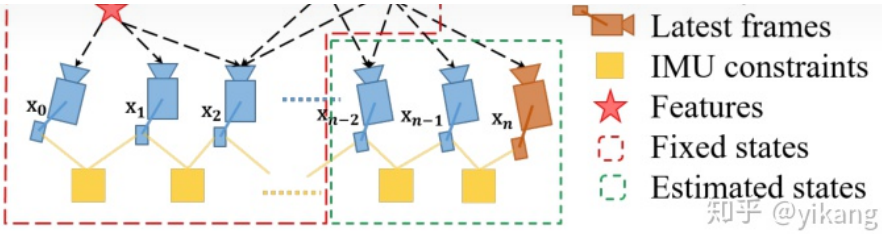
总的来看这一模块的核心是滑动窗口（含边缘化策略）以及视觉惯性BA，前者用于控制计算规模，后者用于实际优化。实现的功能就是初步优化得到了窗口中帧的位姿和速度（其实就是BA的作用）。这一节中同样分为了几个小的部分，大致分开讲一下。

（1）整体上使用视觉惯性BA公式（属于局部BA），**引入了Huber核函数和测量残差的先验信息，通过对构造的误差函数求最小化来获得最大的后验估计。其中测量残差包括IMU的残差和相机视觉的残差**（具体定义的方式在去看原文哈）。在这里提一下与传统的相机图像重投影误差定义方式不同，本文将图像测量残差定义在了单位球面上，将其向球面切平面的两个正交方向上进行分解。

（2）边缘化策略：这个还是比较有代表性的亮点了，这个算法的作用主要是维护滑动窗口的计算规模，新的一帧进来就要Marg掉旧的一帧，但是至于Marg哪一帧，以及如何把被Marg的信息转换为先验信息，这些都归属于**边缘化的策略**中了：

- 如果窗口中第二最新帧是关键帧，则将其保留在窗口中，并将最旧帧及其相应的视觉和惯性测量边缘化，被边缘化的测量将被转换为先验测量；
- 如果第二个最新的帧不是关键帧，只需删除该帧及其所有相应的视觉测量。但是对于非关键帧保留预积分惯性测量，并且预积分过程将继续进行到下一帧。
- 具体的边缘化操作通过舒尔补来进行，具体的实现建议去看一下原文，作者还提到边缘化会导致线性化点的更早确定，这可能会导致次优的估计结果，但是又由于VIO可以接受小的偏移因此认为边缘化带来的负面影响可以忽略不计。

（3）轻量级的视觉惯性BA算法。一般的视觉惯性(VI)-BA优化难以实现相机帧率同步的运行，作者提出了一种轻量级的VI-BA优化算法：不去优化滑动窗口中的所有状态，而是**只优化固定数量的最新IMU状态的姿态和速度。**将特征深度、外部参数、偏差和不希望优化的旧IMU状态**视为常量。**使用所有的视觉和惯性测量来进行仅运动的BA优化。与完全紧耦合的单目VIO（在最先进的嵌入式计算机上可能会超过50ms）相比，仅运动的VI-BA只需要大约5ms的计算时间。示意图如下所示：



（4）故障检测及复原策略。再鲁棒的算法也难免遇到故障或失效的状态，本文的故障检测标准可以概括如下：

- 最新帧中被跟踪的特征数小于某一阈值。这一点还是比较容易理解的，太少的特征点当然会让让运动跟踪以及优化出现问题（跟丢了）。
- 估计器最后两个输出之间的位置或旋转的显著不连续性。这一点也是容易理解的，位移或者角度不可能突然跳变，因此如果出现了较大的不连续性，那就说明出现了故障。
- bias偏差或者外参的估计值出现了较大的变化。理解同上一点，**外参发生大的跳变是不合理的，bias随机游走在短时间内是很小的**（前面我们说了其导数是零均值的高斯分布）。

如果检测到了故障怎么办呢？其实也很简单，那就是再把系统调到初始化阶段（十分简单直接）。

这个**VIO模块的一大作用是使得滚转角和俯仰角变得完全可观，整个系统只剩下三自由度平移和偏航角不可观**，于是后面步骤中会针对这四个自由度优化的误差进行消除。（哦对还得补充一下VIO是如何让roll和pitch变得可观的呢？在初始化部分我们也介绍过，通过旋转对齐重力矢量，我们将相机坐标系和世界坐标系进行了对齐。其实**对齐重力矢量的过程就使得滚转和俯仰可观**了，只剩下绕重力矢量方向的yaw角未知。）

4.重定位模块 relocalization

作者在这一部分里提到了**系统累计误差来源于滑动窗口和边缘化的操作**，更具体一点就是四自由度的优化估计（**三自由度平移以及绕重力方向的偏航角yaw**）。为了消除这一误差，作者提出了一种和VIO模块紧密耦合的重定位模块（就是这一部分的内容）。

概括的说这一模块的过程为：**闭合回环检测、建立当前帧与回环的特征关联、将这些特征关联集成到前面的VIO模块以消除漂移。**

（1）**回环检测。**这一部分使用的是DBoW2词袋算法来识别位置， DBoW2在时间和几何一致性检查之后返回循环闭合候选项，哦对查询时的单词使用的是特征的BREIF描述子（这个是老朋友了应该很熟悉它的计算过程）。

（2）**特征检索。**在上一步检测到闭合回环之后，通过BREIF描述子去检测当前滑动窗口和闭合回环之间的特征并建立关联。但是呢直接对BRIEF描述子进行匹配会有很多异常值（误匹配，这个之前还真是领教过，必须得进行异常值提出不然没法看啊），作者提出了**两步操作来剔除异常值**：

- 2D-2D的基础矩阵测试，测试对象是当前帧和回环候选帧中检索到的特征的二维观测。说的很绕口但其实就是把前面检索到的特征的2D观测拿出来进行基础矩阵的测试，具体怎么测试呢，去看看代码吧！
- 3D-2D的PnP测试，把检索到的特征在当前滑动窗口中对应的3D位置与回环候选帧中特征的2D观测进行测试。

经过上面两步检测，**如果正常特征点的数量大于一定的阈值，就认为回环中的候选帧是正确的**回环匹配，就进行下一步的紧耦合重定位。

（3）**紧耦合重定位。** 重定位过程有效地将当前由单目VIO模块维护的滑动窗口与过去的姿态图对齐（品一品这个对齐的意思）。在这一步重定位中会**使用所有IMU测量值、局部视觉测量值(也就是窗口内的视觉测量值)以及从回环检测中检索到的特征**来联合优化话滑动窗口。

5.全局位姿图优化 global pose graph optimization

键帧是在滑动窗口中被marg掉的帧，这些帧变成了“旧的帧”，进而在回环检测和重定位模块被用来消除滑动窗口中帧的累计误差，相当于是标尺的作用，所以这就意味着我们**要确保这些“旧的帧”得是“准确的”、被优化过的、具有全局一致性的**，这些帧从滑动窗口中被marg掉之前不是进行过重定位对齐了吗？难道还不具备这些条件？确实是不具备的，我们回头去看VIO模块中尽在滑动窗口中优化了固定数量的状态的位姿和速度，得到的结果其实是有误差的（不具备全局一致性），如果没有这个全局图优化模块，即便有前面的回环检测和重定位部分也不过是在有误的地图中进行了定位。

就像在前面两个部分中提到的一样，VIO模块使得位姿中的滚转和俯仰变得可观，只剩下了3自由度平移和偏航角（绕重力向量的旋转）不可观，因此**这里的全局图优化只用来估计这四个自由度，所有又称为4自由度全局位姿图优化。**

同样的这一部分还是将分为建立位姿图、执行优化、图管理等几个部分，分开看一下

（1）向位姿图中加入关键帧。被加入图优化的帧是从滑动窗口中被边缘化的关键帧，并且进入到图中之后会成为顶点，这些顶点有两中连接的边（请注意下面这两种边均只含有4自由度的位姿变换）：

- 连续运动边：这样的一条边代表着滑动窗口中相邻两个关键帧之间的相对运动，边的初始值就取自VIO中局部BA的求解结果。
- 回环匹配边：当位姿图中的一个关键帧成功检测到回环之后，就与回环中的对应帧之间建立起一条这样的边。

（2）4自由度的图优化。图优化想必是大家已经非常熟悉的事情了，我们先定义边的残差，然后引入鲁棒核函数（作者用的是Huber核函数，一般我们也这么用），就得到了损失函数，对损失函数进行最小二乘优化求解。作者使用了多线程来完成图优化。

（3）图优化规模管理。这一机制主要是防止因为行驶路程太长使得全局图优化的计算规模太大而影响实时性，于是作者限制了数据的规模：带有回环约束的关键帧将被保留，而其他太接近或者与相邻的关键帧非常相似的关键帧将被删除。（非常不错的想法，实现也值得一看）

四、实验以及总结

作者将VINS-Mono进行了多次实验和测试，复述结果没有什么意义，但是作者在分析结果时提到了几点问题感觉还是有助于我们理解VINS的原理的。

1. VINS-Mono在滚转roll和俯仰pitch的估计中效果不如OKVIS，原因可能是这两个量是根据IMU的预积分得到的，预积分其实是为了节省计算量而采用的一种一阶近似手段，因此可能造成了误差存在。
2. 关闭了回环检测和重定位功能的VINS-Mono表现大打折扣，和OKVIS一样在四自由度（三自由度平移和yaw角）上有漂移，但是打开回环检测后效果提升就很显著了。
3. 移植到IOS移动端的程序与商用的Google tango比起来丝毫不落下风，尤其是在全局的误差消除上，甚至比tango还要好，这得益于4自由度的全局位姿图优化。

论文的最后作者说了几点**VINS系统接下来的发展方向以及其团队的研究兴趣**，大概有这么几点吧：

1. 在线评价单目视觉系统可观性的方法，以及在若可观性下在线恢复。
2. 将单目VINS系统大规模部署在移动设备（消费电子产品）上。
3. 依托单目VINS系统进行稠密建图。

最后是自己的一点感受吧，整个系统确实非常复杂精妙，这样的成果肯定是作者积淀了很长很长时间所得来的，VIO或者VINS系统的理论可能剖析开来没有多么复杂，但是在实际实现中，为了一个准确性、鲁棒性、实时性要解决的细节和问题太多了。只能说作者“**不愧是拿了华为两百万年薪的人**”。（嗯甚至还觉得两百万给少了……）

发布于 2020-05-29 23:10

写评论 | 吴小奇 关注了作者

还没有评论，发表第一个评论吧

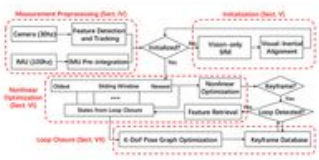
文章被以下专栏收录

C++、CV、SLAM小白入门
记录小白入门之路

推荐阅读

VINS-Mono 代码详细解读1——视觉跟踪...

晓伟Liu 发表于视觉、激光...



VINS-Mono源码解读(三)：视觉惯性联合初始化

任乾 发表于SLAM与...

【flomo 布道师】古典 - 你不是读书少，是记不住和用不上

flomo... 发表于flomo...

VINS-——视

晓伟Liu