

liosam imu预积分文件中我不理解为什么还需要点云配准的数据，数据起到什么作用



在 **LIO-SAM** 或你说的 *liosam imu* 预积分代码中，很多同学一开始都会疑惑：

IMU 预积分已经给了运动估计，为什么还需要点云配准（LO）数据？它在预积分中起什么作用？

下面我用最清晰的逻辑把关系讲清楚。

✓ 1. IMU 预积分 ≠ 完整位姿估计

IMU 预积分提供的只有：

- 相对旋转 ΔR
- 相对速度 Δv
- 相对位移 Δp

但这些都 依赖陀螺仪/加速度计偏差、漂移很快。

IMU 单独预积分是无法给出绝对位置的，只提供短时间相对运动。

✓ 2. 点云配准（LO）给的是低频但可靠的位姿约束

点云配准（Lidar Odometry）例如：

- NDT
- ICP
- LOAM 特色特征匹配

它提供的：

- 6 DoF 位姿（世界坐标）
 - 漂移非常慢
 - 频率低（10~20Hz）
-

★ 关键：IMU 提供短时间高精度相对约束，点云提供长时间全局约束

IMU 漂移快，但短时间很准

Lidar 不漂，但频率低、不够平滑

所以两者结合：

部件	特点	作用
IMU 预积分	高频、短时间准、会漂	用来约束相邻两个时刻之间的运动
点云配准	低频、但绝对位置稳定	用来提供全局的 pose 约束

二者是互补关系。

✓ 3. 在后端因子图中两类因子同时存在

LIO-SAM 的因子图包含：

1. IMU 预积分因子 (BetweenFactor)
 - 约束 $state_i \rightarrow state_{\{i+1\}}$
2. Lidar Odometry 因子 (GPS factor 类似)
 - 约束 $state_i$ 与真实地图的位置
3. Bias 先验因子

整个图优化的核心思路是：

 Copy code

点云提供位置基准（绝对位姿）

IMU 预积分维持连续性和高频约束

因子图优化使两者融合

✓ 为什么 IMU 预积分文件中会读取点云配准的数据？

因为：

预积分本身不需要点云，但后端对 IMU 预积分因子进行优化时，需要点云 LO 提供的位姿作为另一个因子。

预积分文件通常在：

- 更新 bias
- 判断是否需要重新积分
- 使用点云给的 pose 来重置预积分起点

更具体的几个功能：

① 点云估计用于修正 IMU 的状态

当后端优化得到新的 pose/bias 后，需要：

- 更新 imu 状态
- 重新积累预积分因子

② 点云时间戳用于确定关键帧

预积分文件里常常根据点云关键帧来：

- 结束上一次预积分
- 开始下一段预积分

③ 点云约束是优化的 anchor，IMU 只是辅助

IMU 的预积分结果本质只能作为约束，而不是位姿本身。

位姿最终必须由点云提供的约束来“钉住”。

★ 总结（一句话）

点云配准为后端因子图提供绝对位姿约束，而 IMU 预积分提供相对约束，两者必须一起参与优化，因此预积分模块必须读点云的结果用于初始化/更新。

如果你愿意，我可以进一步给你：

- 画一张 LIO-SAM 因子图的关键因素结构图
- 展示 IMU 预积分类（`PreintegratedImuMeasurements`）的代码流程
- 解释 在哪一行代码用到了点云配准的结果

- 帮你分析你正在看的具体文件（如果你贴出文件路径/代码片段）

你想继续看哪一部分？



点云配准的约束会影响imu优化时的残差块吗

简短回答：

会影响，但不是直接改残差块公式，而是通过后端优化后的状态（**pose**、**velocity**、**bias**）反向影响 IMU 预积分残差值，从而间接改变 IMU 残差块。

下面我详细讲“为什么 + 如何影响”。



关键问题

你问的是：

点云配准（Lidar Odometry）的约束会不会影响 IMU 优化的残差？

答案是：

会间接影响，但不是直接改 IMU 残差公式，而是改变残差所依赖的状态量（**pose**、**vel**、**bias**）。



1. 因子图中每个因子都有独立残差

在 LIO-SAM 的因子图里主要有：

- IMU 预积分因子（BetweenFactor）
- Lidar odometry 因子（PoseFactor）
- Bias 先验因子
- GPS / ground truth 因子（可选）

这些因子都是 独立的残差块。

点云配准不会直接改 IMU 残差公式。

IMU 的残差永远是：

$$r_{imu} = \begin{bmatrix} \text{\color{red}\textbackslash Log}(\Delta R^\top(R_i^\top R_j)) \\ R_i^\top(v_j - v_i - g\Delta t) - \Delta v \\ R_i^\top(p_j - p_i - v_i \Delta t - \frac{1}{2}g\Delta t^2) - \Delta p \end{bmatrix}$$

这是固定不变的。

📌 2. 但点云约束会影响优化后的状态 (pose、velocity、bias)

点云配准因子提供：

- 全局 pose 约束
- 低漂移信息

因子图优化时，会同时考虑所有因子：

[markdown](#)

[Copy code](#)

IMU 残差

- + Lidar 残差
- + Bias 先验
- + 其它约束

获得新的 (pose_i, vel_i, bias_i) 的最优解

因此，点云配准的约束会改变 i、j 时刻的优化后状态。

📌 3. 状态变化 ⇒ IMU 残差值变化

虽然 IMU 残差的“公式”不变，但残差是依赖 R_i, R_j, p_i, p_j, v_i, v_j 的。

点云约束改变了这些变量，IMU 残差自然跟着变化：

$$r_{imu}(new) \neq r_{imu}(old)$$

也就是说：

点云配准改变状态量 → 状态量被代入 IMU 残差计算 → IMU 残差值随之变化。

这就是间接影响。

📌 4. 还有一个重要机制：IMU 预积分还会因为点云更新重新计算 (repropagate)

在 LIO-SAM 中：

当优化改变了 bias：

- 预积分器会用新 bias 重新做一次预积分 (repropagate)
- 生成新的 ΔR , Δv , Δp
- 从而改变 IMU 残差

代码位置：

`imuPreintegration.cpp` → `repropagate()`

所以点云约束对 IMU 残差影响有两条路径：

路径 A：点云改变状态 → 改变 IMU 残差值（最主要）

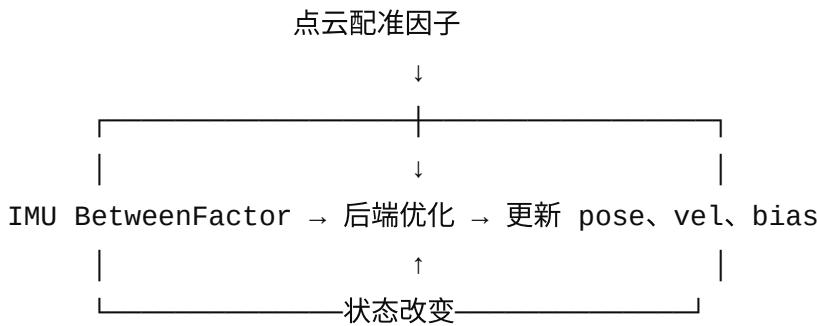
路径 B：点云改变 bias → 触发 IMU 预积分重新计算 → 改变 IMU 残差项 ΔR , Δv , Δp

两条路径都会更新 IMU 残差。

📌 5. 图形化理解

markdown

Copy code



状态改变导致 IMU 残差块重新计算（间接影响）

🧠 总结（最重要的 3 句话）

1. 点云配准不会直接修改 IMU 残差公式（残差表达式永远一样）。
2. 但点云约束会改变优化后的状态（**pose**、**vel**、**bias**）。
3. IMU 残差依赖状态量，因此残差值会随点云优化而改变（间接影响）。

因此：

👉 点云约束会影响 IMU 残差值，但通过后端优化间接发生。

如果你愿意，我可以继续给你：

- IMU 残差如何具体依赖点云优化的变量（逐项分析）
- 画一张 LIO-SAM 的因子图完整结构
- 分析你正在阅读的源码在哪一步体现了这种“间接影响”

你想继续哪一部分？



如果没有点云配准，是不是无法对imu残差进行优化呢

非常好的问题，而且这是 IMU-only SLAM / VIO 的核心本质。

🎯 结论先说：

不是不能优化，但会退化到「纯 IMU 惯导」：无法收敛、漂移无限大、残差无约束，优化会发散。

也就是说：

没有点云/视觉/GPS 等外部观测，IMU 残差无法单独支撑优化。

IMU 因子只提供 相对约束，不提供 绝对约束。
因子图没有 anchor (固定点)，优化就“浮”在空中。

下面详细解释为什么。

🚫 1. IMU 残差不能独立用于优化 —— 因为 IMU 预积分是“相对量”

IMU 预积分提供三类量：

- ΔR (相对旋转)
- Δv (相对速度)
- Δp (相对位移)

关键是：

IMU 给的是 state_i 与 state_j 之间的相对关系，而不是绝对状态。

例如：

$$R_j = R_i \Delta R$$

如果我把所有姿态同时旋转 10° ，IMU 残差不变。

如果我把所有位置整体平移 1 公里，IMU 残差不变。

这叫“不观测性”。

🚫 2. IMU-only 优化出现的问题

问题 ①：系统的绝对位置不可观测

所有平移方向都不受约束 → 解会漂移。

问题 ②：绝对旋转中的 yaw 不可观测

陀螺仪只能测角速度，yaw 与 global heading 没有锚点。

问题 ③：绝对速度不可观测

IMU 算相对速度，但零偏漂移会使速度越算越假。

没有外部观测就修不回来。

问题④：bias 无法估计

IMU bias 和 pose 都在残差里耦合，没有外部量，bias 也不能被唯一确定。

因此：

只有 IMU 因子的因子图 = 完全欠约束系统。

数学上，Jacobian 的秩不满 → 不能求唯一最优解。



3. 为什么加了点云配准（或视觉）就可以优化了？

点云配准因子给两个关键信息：

✓ 绝对位姿约束（非相对量）

例如：

“当前点云在地图中的位置估计为 Pose_k”。

这就提供了 anchor：

- 绝对位置
- 绝对旋转
- yaw 可观测

✓ 为 IMU 的 bias 提供全局修正

IMU 预积分严重依赖 bias，点云能使系统收敛。

因此，当点云 LO 因子加入因子图后：

- 位置受约束 → 不漂了
 - yaw 受约束 → 不绕轴乱转
 - bias 可估计 → 预积分稳定
 - IMU 因子有效传递运动信息 → 优化可收敛
-



4. 没点云时优化会怎样？（真实现象）

如果你尝试去掉 LIO-SAM 的点云配准，留下纯 IMU 预积分，会看到：

- 速度发散
- 位置往多项式爆炸（典型 IMU 漂移）
- bias 跑飞
- 后端矩阵变奇异，导致 Ceres/GTSAM 优化无法收敛
- 稍微转弯会使 yaw 漫无目的漂移

换句话说：

IMU-only 优化无法形成一个有效的因子图。

5. 类比理解（很形象）：

IMU 就像你一直走路时的“脚步感”，你知道自己走了几步、转了几次身，但：

- 没有 GPS（绝对位置），你不知道自己在哪
- 没有指南针（绝对方向），你的方向会漂
- 没有地标（点云），你越走越不准
- 最后你走成一条飞走的自由曲线

点云配准的作用就是提供：

- 地标
- 方向
- 重定位

这样脚步感（IMU）才有意义。

总结（关键 3 句）

- ① 只有 IMU 残差，优化是欠约束的，系统不可观测。
- ② 不能依靠 IMU-only 因子进行后端优化（会发散）。
- ③ 点云因子提供绝对约束，使残差有意义，系统可观测，优化可收敛。

回答你的问题：

如果没有点云配准，是不是无法对 IMU 残差优化？

是的：无法得到有意义的优化结果，数学上不可观测，系统会发散。

如果你愿意，我可以继续深入：

- 帮你推导 IMU-only 的可观测性矩阵，为何秩不满
- 演示在 GTSAM 中 IMU-only 因子图为什么奇异
- 解释为什么要引入各种外部观测（点云/GPS/视觉）