

用户需要准备正确格式的点云数据进行云倾斜，这主要在“imageProjection.cpp”中完成。这两个要求是：

- 提供点时间戳。io-sam使用IMU数据执行点云桌面。因此，需要知道扫描中的相对点时间。最新的Velodyne ROS驱动程序应该直接输出此信息。这里，我们假设点时间通道称为“时间”。点类型的定义位于“imageProjection.cpp”的顶部。“deskewPoint ()”函数利用这个相对时间来获得这个点相对于扫描起始点的变换。当激光雷达以10Hz的速度旋转时，一个点的时间戳应该在0到0.1秒之间变化。如果您正在使用其他激光雷达传感器，则可能需要更改此时间通道的名称，并确保它是扫描中的相对时间。
- 提供点环号。LIO-SAM使用这些信息在矩阵中正确地组织点。环号表示该点属于传感器的哪个通道。点类型的定义位于“imageProjection.cpp”的顶部。最新的Velodyne ROS驱动程序应该直接输出此信息。同样，如果您正在使用其他激光雷达传感器，则可能需要重命名此信息。请注意，目前该软件包只支持机械激光雷达。

IMU 精度和频率 至少 200HZ 9 轴 改进版可用 6 轴

IMU对齐。LIO-SAM将IMU原始数据从IMU帧转换为Lidar帧，遵循ROS REP-105惯例（x-forward, y - left, z - up）。为了使系统正常工作，需要在“参数”中提供正确的外部转换。
“yaml文件”。之所以有两个外接，是因为我的IMU（Microstrain 3DMGX5-25）的加速度和姿态坐标不同。取决于您的IMU制造商，您的IMU的两个外部可能是相同的，也可能不是。以我们的设置为例：

the correct extrinsic transformation needs to be provided in "params.yaml" file.
check whether the readings correspond to the sensor's movement

- 我们需要设置x-z加速度和陀螺仪负的读数来转换激光雷达框架中的IMU数据，这是由“params.yaml”中的“extrinsicRot”表示的。

- 态度读数的转变可能略有不同。IMU的姿态测量 q_{wb} 通常表示IMU坐标系中点到世界坐标系（如ENU）的旋转。

然而，该算法需要 q_{wl} ，即从激光雷达到世界的旋转。因此，我们需要从lidar到IMU q_{bl} 的旋转，其中 $q_{wl} = q_{wb} * q_{bl}$ 。为方便起见，用户只需要在“params”中将 q_{1b} 作为“extrinsicRPY”提供。（如果加速度和姿态具有相同的坐标，则与“extrinsicRot”相同）。

坐标转换 从激光——IMU——世界

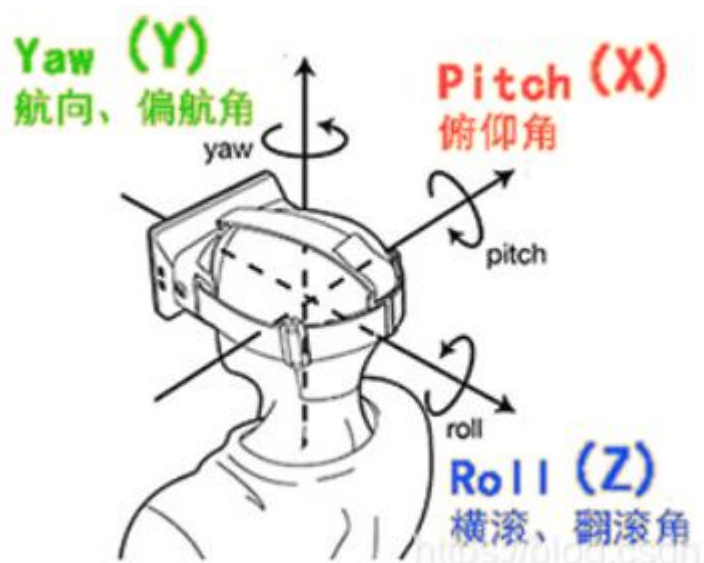
- 循环闭包：这里的循环函数给出了一个概念证明的例子。它直接改编自乐高-loam循环闭合。有关更高级的循环闭包实现，请参阅ScanContext。在params中设置loopClosureEnableFlag。将Yaml “” 设置为“true”以测试循环闭包函数。在Rviz中，取消选中“地图（云）”，选中“地图（全局）”。这是因为可视化地图——“地图（云）”-在Rviz中只是一堆点云。姿态校正后，它们的位置不会更新。这里的循环关闭功能只是改编自LeGO-LOAM，这是一种基于icp的方法。因为ICP运行得很慢，所以建议将播放速度设置为“-r 1”。您可以尝试使用Garden数据集进行测试。

- **Loop closure:** The loop function here gives an example of proof of concept. It is directly adapted from LeGO-LOAM loop closure. For more advanced loop closure implementation, please refer to [ScanContext](#). Set the "loopClosureEnableFlag" in "params.yaml" to "true" to test the loop closure function. In Rviz, uncheck "Map (cloud)" and check "Map (global)". This is because the visualized map - "Map (cloud)" - is simply a stack of point clouds in Rviz. Their position will not be updated after pose correction. The loop closure function here is simply adapted from LeGO-LOAM, which is an ICP-based method. Because ICP runs pretty slow, it is suggested that the playback speed is set to be "-r 1". You can try the Garden dataset for testing.

GPS 添加到因子图中，可以改变权重

LOOPCLOSURE 是否开启回环

体素滤波降采样



Question

点云偏斜导致漂移

松耦合： 例子： IMU 在扫描匹配前进行运动

imu:

$$\hat{\omega}_t = \omega_t + \mathbf{b}_t^\omega + \mathbf{n}_t^\omega \quad (2)$$

$$\hat{\mathbf{a}}_t = \mathbf{R}_t^{\mathbf{B}\mathbf{W}}(\mathbf{a}_t - \mathbf{g}) + \mathbf{b}_t^{\mathbf{a}} + \mathbf{n}_t^{\mathbf{a}}, \quad (3)$$

where $\hat{\omega}_t$ and $\hat{\mathbf{a}}_t$ are the raw IMU measurements in \mathbf{B} at time t . $\hat{\omega}_t$ and $\hat{\mathbf{a}}_t$ are affected by a slowly varying bias \mathbf{b}_t and white noise \mathbf{n}_t . $\mathbf{R}_t^{\mathbf{B}\mathbf{W}}$ is the rotation matrix from \mathbf{W} to \mathbf{B} . \mathbf{g} is the constant gravity vector in \mathbf{W} .

零偏、白噪声、旋转矩阵

论文计算使用二阶 T

然后，我们应用[20]中提出的IMU预积分方法来获得两者之间的相对物体运动

步伐。预积分测量值 $\Delta \mathbf{v}_{ij}$, $\Delta \mathbf{p}_{ij}$, 和

$\Delta \mathbf{R}_{ij}$ 时间 i 和 j 之间的 \mathbf{R}_{ij} 可以用下面的公式计算：

during the above integration.

We then apply the IMU preintegration method proposed in [20] to obtain the relative body motion between two timesteps. The preintegrated measurements $\Delta \mathbf{v}_{ij}$, $\Delta \mathbf{p}_{ij}$, and $\Delta \mathbf{R}_{ij}$ between time i and j can be computed using:

We can now use the measurements from the IMU to infer the motion of the robot. The velocity, position and rotation of the robot at time $t + \Delta t$ can be computed as follows:

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \mathbf{g}\Delta t + \mathbf{R}_t(\hat{\mathbf{a}}_t - \mathbf{b}_t^{\mathbf{a}} - \mathbf{n}_t^{\mathbf{a}})\Delta t \quad (4)$$

$$\begin{aligned} \mathbf{p}_{t+\Delta t} = \mathbf{p}_t + \mathbf{v}_t\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 \\ + \frac{1}{2}\mathbf{R}_t(\hat{\mathbf{a}}_t - \mathbf{b}_t^{\mathbf{a}} - \mathbf{n}_t^{\mathbf{a}})\Delta t^2 \end{aligned} \quad (5)$$

$$\mathbf{R}_{t+\Delta t} = \mathbf{R}_t \exp((\hat{\boldsymbol{\omega}}_t - \mathbf{b}_t^{\boldsymbol{\omega}} - \mathbf{n}_t^{\boldsymbol{\omega}})\Delta t), \quad (6)$$

where $\mathbf{R}_t = \mathbf{R}_t^{\mathbf{WB}} = \mathbf{R}_t^{\mathbf{BW}^T}$. Here we assume that the angular velocity and the acceleration of \mathbf{B} remain constant during the above integration.

We then apply the IMU preintegration method proposed in [20] to obtain the relative body motion between two timesteps. The preintegrated measurements $\Delta \mathbf{v}_{ij}$, $\Delta \mathbf{p}_{ij}$, and $\Delta \mathbf{R}_{ij}$ between time i and j can be computed using:

$$\Delta \mathbf{v}_{ij} = \mathbf{R}_i^T(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) \quad (7)$$

$$\Delta \mathbf{p}_{ij} = \mathbf{R}_i^T(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2) \quad (8)$$

$$\Delta \mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j. \quad (9)$$

<https://blog.csdn.net/tiancanink>

激光雷达:

提取边缘和平面特征

关键帧选择

变化超出设定的阈值时选取为关键帧作为状态节点添加到因子图中 1m 10°

体素贴图: 滑动窗口提取 N 个最新的关键帧进行估算, 将其使用相关的变换 转化为 帧 \mathbf{w} , 合并到一个体素贴图中 边缘特征/平面特征体素贴图

从 IMU 预测中获得初始转换

高斯牛顿法 最优化

F和FP是w中变换后的边缘和平面特征，然后对M和M进行下采样，以消除落在同一体素单元中的重复特征。在本文中，n取25。M和M的下样分辨率分别为0.2m和0.4m。

$$M_i^e = F_i^e \cup F_{i-1}^e \cup \dots \cup F_{i-n}^e$$

F_i^e and F_i^p are the transformed edge and planar features in W . M_i^e and M_i^p are then downsampled to eliminate the duplicated features that fall in the same voxel cell. In this paper, n is chosen to be 25. The downsample resolutions for M_i^e and M_i^p are $0.2m$ and $0.4m$, respectively.

GPS:

转为局部笛卡尔坐标 与激光雷达是否线性同步？

由于漂移缓慢，只在估计位置的协方差大于接收到的 GPS 信号的协方差时添加 GPS 因子

回环:

基于欧氏距离检测

对于海拔检测非常有用