浮动主题

SC-LIO-SAM_Relo代码框架

两者各有自己独立的isam2图优化系统，两者紧耦合imuPreintegration.cpp中的isam2利用imu里程计和激光里程计在不断优化imu增量位姿和临偏bias，为scan2map提供最好的初始位姿；mapOptmization.cpp中的isam2利用imu增量里程计提供的初始位姿进行scan2map优化得到激光里程计位姿，再结合gps因子和回环因子，得到最好的激光里程计

featureExtraction.cpp 提取线面特征

- Subscriber
  - subLaserCloudInfo 订阅去畸变后的点云
- Publisher
  - pubCornerPoints 发布线点点云
  - pubSurfacePoints 发布面点点云
  - pubLaserCloudInfo 发布cloud_info包含特征点云
- laserCloudInfoHandler() 订阅去畸变点云的回调函数，来自imageProjection.cpp
  - calculateSmoothness() 根据前后五个点计算当前点的曲率
  - markOccludedPoints() 遮挡或平行的点不提取特征
  - extractFeatures() 提取线面特征，遍历扫描线，每根扫描线扫描一周的点云划分为6段，针对每段提取20个角点、不限数量的平面点，加入角点集合、平面点集合
  - publishFeatureCloud() 发布特征点云

mapOptmization.cpp scan2map地图优化

- struct PointXYZIRPYT 注册点云格式
- Subscriber
  - subCloud 订阅点云信息cloud_info
  - subGPS 订阅GPS消息(可选)
  - **subLoop 订阅外部回环消息(可添加自己设计的回环检测方式)**
- Publisher
  - pubKeyPoses 发布关键帧
  - pubLaserCloudSurround 发布纯激光里程计
  - pubLaserOdometryIncremental 发布与imu数据融合(对z限制)后的激光里程计
  - pubPath 发布激光里程计轨迹(path)
  - pubHistoryKeyFrames 发布闭环所对应的历史submap
  - pubIcpKeyFrames 发布闭环icp变换后的特征点云
  - pubLoopConstraintEdge 发布闭环连线
  - pubRecentKeyFrames 发布局部降采样关键帧平面点合集
  - pubRecentKeyFrame 发布所有历史帧的线点、平面点集合
  - pubCloudRegisteredRaw 发布原始点云配准结果
- laserCloudInfoHandler() 激光点云的回调函数
  - 满足地图优化频率 mappingProcessInterval = 0.15s
  - updateInitialGuess() 为当前激光帧scan2map优化提供初始位姿。如果是第一帧，则使用imu原始数据仅初始化rpy，xyz设为0；如果不是第一帧则使用imu增量里程计 cloud_Info.initialGuess*，左乘上一帧imu增量里程计的逆，得到imu相对于上一激光帧时刻时的imu增量变化，然后再左乘上一激光帧位姿，得到当前激光帧在map系下的初始位姿
  - extractSurroundingKeyFrames() **时空提取**相邻关键帧
    - extractNearby()
      - 空间上提取相邻关键帧，并对得到的位姿点云降采样；时间上10s内提取相邻关键帧，针对机器人原地维持不动的情况
      - extractCloud() 将相邻关键帧集合对应的角点、平面点，加入到局部map中，作为scan2map优化的submap特征点云(有个重复筛选操作，毕竟是时间和空间上同时操作，难免重复)
  - downsampleCurrentScan() 降采样。一是SC点云输入；二是特征稀疏化
  - scan2MapOptimization() scan2map优化
    - 检测特征是否足够多
    - cornerOptimization() 线特征优化准备
      - updatePointAssociateToMap() 变换矩阵形式转换
      - pointAssociateToMap() 将某个特征点进行变换
      - 在submap中kd搜索5个最近的特征；根据协方差矩阵特征值判断是否可以进行直线拟合；若可以则进行直线拟合；计算该线特征到拟合直线的距离
    - surfOptimization() 面特征优化准备
      - updatePointAssociateToMap() 变换矩阵形式转换
      - pointAssociateToMap() 将某个特征点进行变换
      - 在submap中kd搜索5个最近的特征；根据协方差矩阵特征值判断是否可以进行平面拟合；若可以则进行平面拟合；计算该面特征到拟合平面的距离
    - combineOptimizationCoeffs() 组合待优化参数
    - LMOptimization() 高斯牛顿优化，更新transformTobeMapped位姿(不熟悉)
    - transformUpdate() 使用9轴imu的orientation与做transformTobeMapped插值，并且roll和pitch收到常量阈值约束（权重）
  - saveKeyFramesAndFactor() 存储关键帧并进行isam2优化
    - saveFrame() 检测是否为关键帧
    - addOdomFactor() 在isam2中加入激光里程计因子
    - addGPSFactor() 在isam2中加入gps因子(可选，那么我们是否可以改成加入相机因子呢，同时用相机提供语义信息)
    - addLoopFactor() 在isam2中加入回环因子
    - 执行isam2优化，更新thisPose3D、thisPose6D、transformTobeMapped
    - makeAndSaveScancontextAndKeys() 计算当前关键帧的SC描述子保存为scd格式，并将关键帧点云保存为pcd格式
    - updatePath() 更新rviz显示轨迹(白色)
  - correctPoses() 修正位姿(有点多余，但确保位姿修正)
  - publishOdometry() 发布最终的激光里程计位姿，包括全局位姿轨迹和tf消息(涉及加权)
  - publishFrames() 发布局部地图、特征点云、全局轨迹等
- std::mutex mtx
- loopClosureThread() 多线程并发之回环检测线程
  - performRSLoopClosure() RS回环检测(两次锁)
    - 备份cloudKeyPoses3D，对z轴方向进行约束为1.1，有利于RS仅在水平xy方向进行回环检测
    - detectLoopClosureExternal() 外部回环输入(未使用)
    - detectLoopClosureDistance() 寻找时间戳最早的激光帧作为回环匹配候选帧
    - loopFindNearKeyframes() 根据回环匹配候选帧组成submap并发布
    - scan2map的icp配准检测，并保存回环因子
  - performSCLoopClosure() SC回环检测(一次锁)
    - scManager 进行sc回环检测(提及没有使用得到的yawdiff进行icp初始化..但是在RS回环中也并没有进行icp初始化)
    - loopFindNearKeyframes() 根据闭环匹配候选帧组成submap并发布
    - scan2map的icp配准检测，并保存回环因子
  - visualizeLoopClosure() 回环连线可视化(无锁)
- visualizeGlobalMapThread() 多线程并发之局部地图发布并保存全局地图
  - publishGlobalMap() 发布局部地图(一次锁)，两次降采样之后才发布
  - ctrl c 结束ros进程，保存全局地图
- std::mutex mtxLoopInfo
  - loopInfoHandler() 存储回环消息
  - detectLoopClosureExternal() 检测外部回环(此函数未使用，只是提供外部回环API)
- saveOptimizedVerticesKITTIformat() 保存isam2中的节点信息，并保存为kitti格式，便于evo评估
- writeVertex() & writeEdge() 打印保存isam2节点和边信息
- transformPointCloud() 点云变换(并行操作)

Scancontext.cpp SC回环检测

- makeAndSaveScancontextAndKeys() 生成sc描述子信息
  - makeScancontext() 制作sc
  - makeRingkeyFromScancontext() 二维sc矩阵row取平均变一维
  - makeSectorkeyFromScancontext() 二维sc矩阵column取平均变一维
  - eig2stdvec() eigen格式变vector格式
- detectLoopClosureID() 寻找sc回环
  - 生成搜索树，用于执行knn搜索，每10个sc添加后更新一次
  - distanceBtnScanContext() 计算sc差异，找到相似度最高的sc回环匹配候选帧ID和前3名得分的sc描述子ID，并计算yawdiff

relocationVisual.cpp 基于SC的重定位

- 无 Subscriber
- Publisher
  - publishGlobalMap 发布全局先验地图
  - publishGlobalTraj 发布全局先验轨迹
  - publishsubMap 发布submap点云
  - publishReloCloud 发布定位成功后的点云
  - publishReloPose 发布定位成功后的位姿
  - publishPath 发布定位轨迹
- read() 读取先验地图、轨迹的pcd
- publishCloud() 多功能发布点云格式消息，包括点云、位姿等
- PathPlaner() 生成全局一致性的地图，用于实现导航功能(导航功能暂未实现)
- relocationShow() 重定位可视化
  - SCDataBase() 生成sc数据库
  - PathPlaner() 生成全局一致性地图
  - cloud2Relo() 当前扫描帧定位
    - curCloud2SC() 生成当前扫描帧的sc描述子
    - scManagerRL.detectLoopClosureID() 寻找sc匹配候选帧，得到1个高置信度候选和2个低置信度候选
    - 利用scManagerRL提供的yawdiff作为初始只，进行scan2map的icol匹配检测，得分最低者为最优回环匹配ID
  - 遍历pcd数据库，依次定位并发布定位结果

imuPreintegration.cpp imu预积分

- *.yaml文件 各种参数配置
  - params_mulran.yaml 针对Mulran数据集在extRot上特殊处理
  - params_liosam.yaml
- run_*.lanuch文件 发布多个节点
  - rosparam 导入yaml文件
  - 发布多个launch文件
- utility.h 数据预处理
  - 点云类型 typedef pcl::PointXYZI
  - 传感器类型 SensorType { MULRAN, VELODYNE, OUSTER }
  - class ParamServer 参数类
    - nh.param参数赋值与yaml中的rosparam互应
    - publishCloud 将pcl点云消息转化为ros点云消息并发布
    - imuConverter 将imu数据旋转变换到"中间系"，与lidar系差一个平移
    - ROS_TIME() 把ros时间变成标准时间
    - imuAngular2rosAngular() 读取imu角速度
    - imuAccel2rosAccel() 读取imu线速度
    - imuRPY2rosRPY() 读取imu位姿rpy(使用tf格式技巧)
    - pointDistance() 计算点之间的距离
    - saveSCD() 将每个关键帧的SC描述子保存为scd文本格式(保留3位有效数字)
    - rankS2L() std::sort排序函数
- TransformFusion 变换融合类
  - Subscriber
    - subImuOdometry 订阅imu增量里程计odometry/imu_incremental，来自IMUPreintegration。它是一个关于相邻两帧激光帧之间的imu增量集，上一帧激光帧直到当前激光帧的**每个相邻imu的里程计增量**
    - subLaserOdometry 订阅激光里程计mapping/odometry，来自mapOptimization。它是scan2map优化后的里程计结果，上一帧激光里程计的优化结果
  - Publisher
    - pubImuOdometry 发布imu先验里程计odometry/imu，用于rviz显示(粉红色)。它是"subLaserOdometry订阅"+"subImuOdometry订阅的**首尾相对变换**"="可显示的imu里程计先验"
    - pubImuPath 发布imu里程计轨迹imu/path，单纯发布subImuOdometry订阅的最后最后一个imu增量里程计位姿(感觉没什么用)
  - lidarOdometryHandler() 激光里程计回调函数
  - std::mutex mtx
    - imuOdometryHandler() imu增量里程计回调函数。它刷除早于上一帧激光帧的imu增量里程计数据，计算imu增量里程计首尾的相对变换，再左乘上一帧激光里程计的逆，得到基于上一优化激光帧下的当前时刻imu的里程计位姿，并pubImuOdometry发布。同时，pubImuPath 也发布了最后一个imu增量里程计位姿。最后，发布了baselink的tf消息便于rviz中的显示
  - odom2affine() 订阅odom消息，并将其转化为变换矩阵
- IMUPreintegration imu预积分类
  - Subscriber
    - subImu 订阅imu原始数据
      - imu预积分优化组
        - imuIntegratorOpt_ 预积分器
        - imuQueOpt[] 对当前激光帧时刻之前的imu预积分
      - imu预积分传播组
        - imuIntegratorImu_ 预积分器
        - imuQueImu[] 对当前激光帧时刻之后的imu预积分
    - subOdometry 订阅激光增量里程计mapping/odometry_incremental(与mapping/odometry，只是与imu融合了一下)，来自mapOptimization。作为因子用来isam2优化
  - Publisher
    - pubImuOdometry 发布imu增量里程计，用来在TransformFusion中pubImuOdometry发布imu里程计先验(在rviz中显示)
  - std::mutex mtx
    - odometryHandler() 激光增量里程计回调函数。具体过程为：1)若未初始化，去除imuQueOpt[]中当前激光帧之前的imu数据，结合当前激光帧位姿作为位姿先验信息，另外初始化速度、imu偏置先验。然后跳出函数；2)若初始化成功，则将imuQueOpt[]在当前激光帧之前的imu数据全部预积分，得到一个imu积分器预测状态，再结合当前激光帧的增量位姿，共同加入isam2约束。得到一个优化状态(注意isam2中的约束信息大于100时会重置)；3)再得到当前时刻的优化状态(主要考虑imu优化后的偏置bias)后，删除imuQueImu[]当前激光帧之前的imu数据，对剩余的所有imu进行预积分，这是为了防止imuHandler()中对imuQueImu[]中每个imu预积分时，产生前面imu数据的遗漏
    - imuHandler() 订阅imu原始数据的回调函数。它会将imu原始数据逐个存储imuQueOpt[]到和imuQueImu[]中，并将每一个imuQueImu[]**当前接收到的**imu数据预积分一次，然后pubImuOdometry在前一帧isam2优化基础上发布imu增量里程计
  - failureDetection() 检测imu数据是否震荡太大导致无效

imageProjection.cpp 点云映射

- 根据激光传感器类型向PCL库申请点云类型
- Subscriber
  - subImu 订阅imu原始数据
  - subOdom 订阅imu增量里程计，来自IMUPreintegration
  - subLaserCloud 订阅原始点云数据
- Publisher
  - pubExtractedCloud 发布去畸变且提取后的点云数据，用于后面提取线面特征
  - pubLaserCloudInfo 发布当前帧点云的各种信息cloud_info，来自自定义消息类型cloud_msg
- std::mutex imuLock
  - imuHandler() 订阅原始imu数据的回调函数
  - deskewInfo() 去畸变处理
- std::mutex odoLock
  - odometryHandler() 订阅imu增量里程计的回调函数
  - deskewInfo() 去畸变处理
- cloudHandler() 订阅原始点云的回调函数(核心函数)
  - cachePointCloud() 至少缓存3帧点云原始数据；取队首为当前激光帧；根据传感器类型修改点云信息；检查点云信息是否包含ring和time；得到当前激光帧的扫描起止时间
  - deskewInfo() 计算畸变参数，保证imu数据覆盖当前帧扫描起止时间
    - imuDeskewInfo() 去除当前激光帧前0.01s之前的imu数据；遍历留下的imu，直到最接近当前激光帧开始时刻的imu位姿并记录；假设每个imu数据间一致性，在初始化第一个imu增量为0下，存储每一时刻imu相对于初始时刻的旋转增量；设置当前激光帧的Imu数据可用标志
    - odomDeskewInfo() 去除当前激光帧前0.01s之前的imu增量里程计；遍历留下的imu增量里程计，直到找到最接近当前激光帧开始时刻的imu增量里程计，记录为当前激光帧的初始位姿cloud_info.initialGuess*；根据imu增量里程计，计算当前帧扫描起止时刻间的相对变换
  - projectPointCloud() 根据传感器range信息映射至rangeimage；对当前帧点云中的每个点逐个去畸变，fullCloud存储为一维点云向量；但是貌似两者之间没有什么关系
    - 坐标变换 velodyne的坐标系里，x轴向前，y向左，扫描方向是顺时针，从y轴开始；在xy坐标系内，扫描顺序是90度->0度->-180度(180度)->90度；一系列坐标变换，使得y轴(90度)的扫描起始时刻的横向索引为0。十分巧妙
    - deskewPoint() 对某个点去畸变，变换到扫描起始第一个点的坐标系下
      - findRotation() 旋转插值
      - findPosition() 平移插值
  - cloudExtraction() 根据rangeimage提取有效的点，存储在extractedCloud一维点云内，并记录每个ring扫描起止位置前5个点的索引
  - publishClouds() 发布extractedCloud；发布cloud_info消息
  - resetParameters()
- record_initialguesss() 记录imu里程计给出的初始位姿initial_guess*(感觉不太准)