**Ubuntu20.04 ROS1 Noetic安装LIO\_SAM等Slam包流程**

<https://github.com/YJZLuckyBoy/liorf> LIORF修改后的LIO-SAM

<https://github.com/KomasaQi/liorf_localization> 基于LIORF的定位算法

<https://github.com/TixiaoShan/LIO-SAM> 最基本的LIO-SAM，上面的方法是以其为基础进行修改的分支算法

首先需要做的是配置环境

从 GTSAM 官方仓库下载 4.0.3 版本源码（通过 git 或直接下载压缩包）：

# 进入用户目录（避免权限问题）

cd ~

# 克隆GTSAM仓库并切换到4.0.3版本

git clone https://github.com/borglab/gtsam.git

cd gtsam

git checkout 4.0.3  **# 切换到4.0.3稳定版，否则后续程序会报错！！！！！！！！！！**

# 创建编译目录（规范操作，避免污染源码）

mkdir build && cd build

# 配置CMake（重点：关闭Python绑定以加速编译，保留Boost支持）

cmake -DGTSAM\_BUILD\_EXAMPLES=OFF -DGTSAM\_BUILD\_TESTS=OFF -DGTSAM\_BUILD\_PYTHON=OFF -DGTSAM\_WITH\_EIGEN=ON -DGTSAM\_USE\_BOOST=ON -DCMAKE\_INSTALL\_PREFIX=/usr/local ..

#安装到系统目录

sudo make install

# ROS 依赖

sudo apt update

sudo apt install -y ros-noetic-ros-base \

ros-noetic-tf ros-noetic-pcl-conversions ros-noetic-cv-bridge \

ros-noetic-visualization-msgs ros-noetic-geometry-msgs ros-noetic-nav-msgs \

libpcl-dev libopencv-dev libgeographic-dev ros-noetic-robot-localization ros-noetic-robot-state-publisher ros-noetic-navigation

下面是下载仓库

cd ~ && mkdir lio\_ws && cd lio\_ws && mkdir src && cd src

git clone <https://github.com/YJZLuckyBoy/liorf>

git clone <https://github.com/KomasaQi/liorf_localization>

git clone <https://github.com/TixiaoShan/LIO-SAM>

cd ~/lio\_ws

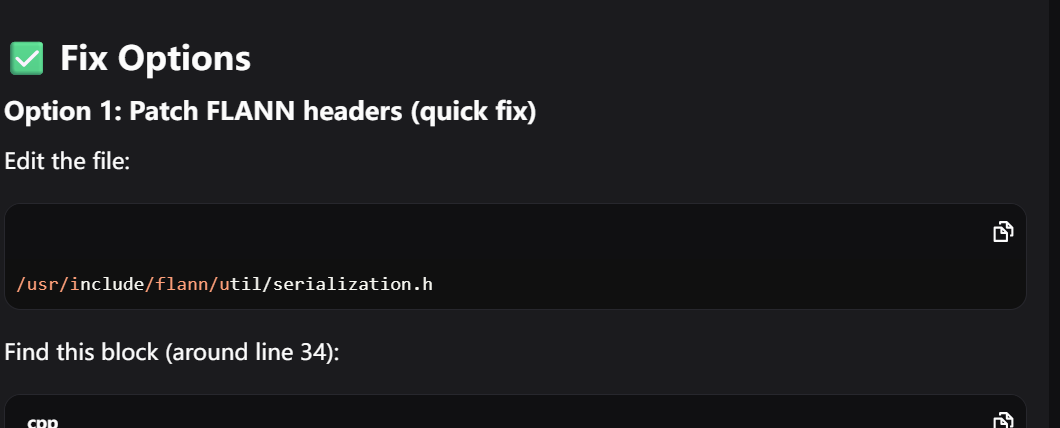
catkin\_make

如果编译没有报错，就可以参考上述仓库的官方文档进行建图测试与定位测试，如果有报错，请复制编译所有输出尽可能多给AI，问一下大概都能解决，估计是依赖项缺失之类的。比如opencv.h的问题， C++11改C++14的问题，FLANN的问题等，按AI给出的解决方案就ok。

修改LIO-SAM/src/include/Utility.h当中的#include <opencv/cv.h>修改为#include <opencv2/opencv.hpp>

将LIO-SAM/CMakeList.txt前几行里的c++11改成c++14

然后 sudo gedit /usr/include/flann/util/serialization.h,按照下面的指示将





修改一下第34行，注意前面的type删除了，挪到了括号里面，这一点要尤其注意仔细修改

编译成功以后要运行我们的建图节点

数据：链接: https://pan.baidu.com/s/1-sAB\_cNlYPqTjDuaFgz9pg 提取码: ejmu （walk不需要改配置文件，其他两个需要<https://blog.csdn.net/unlimitedai/article/details/107378759#t1> 下文有）

播放下载的bag：

cd ~/lio\_ws

# 启动 LIO-SAM

source devel/setup.bash #（建议把这个指令写到~/.bashrc中）

roslaunch lio\_sam run.launch

#打开另一个终端

在终端 3 播放 casual\_walk.bag 数据包：

cd ~/下载 # 数据包所在目录

rosbag play casual\_walk.bag -r 1

放bag的同时liosam可以进行建图，bag播放完并算法建图完后需要通过命令保存建好的地图。根据lio-sam的github说明运行对应的rosservice存储建好的地图

播放结束可以执行下面的命令来存储pcd地图。

cd ~/lio\_ws

source devel/setup.bash

rosservice call /lio\_sam/save\_map 0.2 "/下载/LOAM"

注意上面的这个地址是默认在~文件夹下的，不需要加前缀，同时程序会将原有的这个文件夹删掉并重新创建。

* 注意：服务名称是 /lio\_sam/save\_map（固定名称，与代码路径无关）。

，确保路径填写正确，不要重复包含用户目录（如 /home/car）。