

Lio-Mapping compilation

1、 Install dependencies

ROS(ubuntu 18.04 melodic)

If Ceres solver is not installed, follow the steps below to install it.

```
#Find it under the lio_mapping/dependent software folder and extract it to the
motherboard for installation.
mkdir build && cd build
cmake ..
sudo make install
```

If GTSAM is not installed, follow the steps below to install it.

```
wget -O ~/Downloads/gtsam.zip https://github.com/borglab/gtsam/archive/4.0.0-
alpha2.zip
#If the download is slow, find it under the lio_mapping/dependent software folder
and extract it to the motherboard for installation.
cd ~/Downloads/ && unzip gtsam.zip -d ~/Downloads/
cd ~/Downloads/gtsam-4.0.0-alpha2/
mkdir build && cd build
cmake ..
sudo make install
```

ndt_omp

2、 Source code path

imuCalibEx:

```
src/imuCalibEx
```

Lio-Mapping:

```
src/lio-mapping
```

3、 Compile

Here we take Jetson nano as an example. The installation environment is ubuntu 18.04 and the ROS version is melodic. ROS (ubuntu 18.04 melodic), Ceres solver, GTSAM and other dependencies have been installed by default.

Compile

If it is the first time to compile, please enter the following command:

```

sudo mv /usr/include/flann/ext/lz4.h /usr/include/flann/ext/lz4.h.bak

sudo mv /usr/include/flann/ext/lz4hc.h /usr/include/flann/ext/lz4.h.bak

sudo ln -s /usr/include/lz4.h /usr/include/flann/ext/lz4.h

sudo ln -s /usr/include/lz4hc.h /usr/include/flann/ext/lz4hc.h

```

Then enter Lio_Sam's workspace to compile

```

cd ~/lio_mapping_ws

catkin build

```

Compilation completed effect:

```

cd /home/vidia/lio_mapping_ws/build/vanjee_lidar; catkin build --get-env vanjee_lidar | catkin env -si /usr/bin/make --jobserver-fds=6,7 -j; cd -
.....
Finished <<< vanjee_lidar [ 2 minutes and 36.3 seconds ]
[build] Summary: All 6 packages succeeded!
[build] Ignored: None.
[build] Warnings: 2 packages succeeded with warnings.
[build] Abandoned: None.
[build] Failed: None.
[build] Runtime: 5 minutes and 44.7 seconds total.
[build] Note: Workspace packages have changed, please re-source setup files to use them.
nvdi@ubuntu:~/lio_mapping_ws$

```

What you need to pay attention to is when compiling imuCalibEx, if it is on a Jetson nano class arm motherboard. You need to modify the location in the src/imuCalibEx/src/imu_lidar_calibration/ndt_omp/CMakeLists.txt file, as shown in the figure below.

```

SET(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS}")

```

```

CMakeLists.txt
cmake_minimum_required(VERSION 3.10)
project(ndt_omp)

add_definitions(-std=c++14)
set(CMAKE_CXX_FLAGS "-std=c++14")

if (BUILD_WITH_MARCH_NATIVE)
    add_compile_options(-march=native)
else()
    #add_definitions(-msse -msse2 -msse3 -msse4 -msse4.1 -msse4.2)
    #set(CMAKE_CXX_FLAGS "-msse -msse2 -msse3 -msse4 -msse4.1 -msse4.2")
    SET(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS}")
endif()

# pcl 1.7 causes a segfault when it is built with debug mode
set(CMAKE_BUILD_TYPE "RELEASE")

#find_package(PCL 1.7 REQUIRED)
#include_directories(${PCL_INCLUDE_DIRS})
#link_directories(${PCL_LIBRARY_DIRS})
#add_definitions(${PCL_DEFINITIONS})

message(STATUS "PCL INCLUDE DIRS:" ${PCL_INCLUDE_DIRS})
message(STATUS "PCL LIBRARY DIRS:" ${PCL_LIBRARY_DIRS})
message(STATUS "PCL DEFINITIONS:" ${PCL_DEFINITIONS})

find_package(OpenMP)
if (OPENMP_FOUND)
    set (CMAKE_C_FLAGS "${CMAKE_C_FLAGS} ${OpenMP_C_FLAGS}")
    set (CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} ${OpenMP_CXX_FLAGS}")

```

You also need to modify the file `src/imuCalibEx/src/imu_lidar_calibration/linkalibr/CMakeLists.txt`, as shown below:

```
set(LIB_TBB_DIR /usr/lib/aarch64-linux-gnu)
```

```
79 src/init/InertialInitializer.cpp
80 src/state/State.cpp
81 src/state/StateHelper.cpp
82 src/state/Propagator.cpp
83 src/update/UpdaterLidarOdometry.cpp
84 src/track/lidarOdometry.cpp
85 src/core/linkalibrManager.cpp
86 )
87 target_link_libraries(linkalibr_lib ${thirdparty_libraries})
88 target_include_directories(linkalibr_lib PUBLIC src)
89
90 #####
91 # Adding different executables
92 #####
93 add_executable(ros_test_node src/ros/linkalibr_test.cpp)
94 target_link_libraries(ros_test_node linkalibr_lib ${catkin_LIBRARIES})
95
96 add_executable(ros_calib_init src/ros_calib_init.cpp)
97 target_link_libraries(ros_calib_init linkalibr_lib ${catkin_LIBRARIES})
98
99 #set(LIB_TBB_DIR /usr/lib/x86_64-linux-gnu) If it is an X86 host system, you need to remove this comment
100
101 set(LIB_TBB_DIR /usr/lib/aarch64-linux-gnu)
102 When compiling the jetson nano system, you need to remove the comments on this end of the code.
103 add_library(libtbb SHARED IMPORTED)
104 set_target_properties(libtbb PROPERTIES IMPORTED_LOCATION ${LIB_TBB_DIR}/libtbb.so.2)
105
106 add_executable(ros_calib_init_optimizer src/ros_calib_init_optimizer.cpp)
107 target_link_libraries(ros_calib_init_optimizer
```