深蓝学院
shenlonxueyuon.com

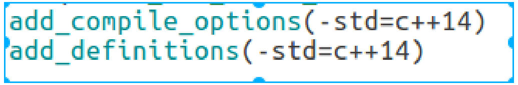
第一次作业讲解（组内）

主讲人—土山石

0 编译

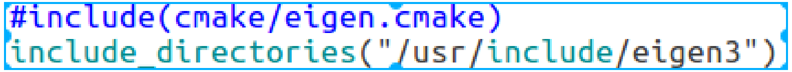
CmakeList 修改：

1. C++14



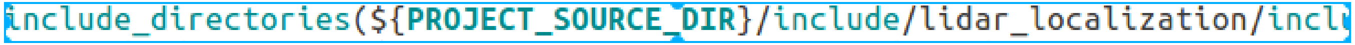
```
add_compile_options(-std=c++14)
add_definitions(-std=c++14)
```

2. 更换 Eigen 版本



```
#include(cmake/eigen.cmake)
include_directories("/usr/include/eigen3")
```

3. 增加文件后不识别

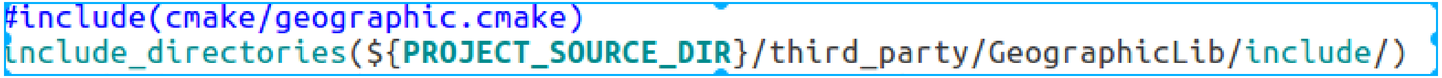


```
include_directories(${PROJECT_SOURCE_DIR}/include/lidar_localization/incl
```

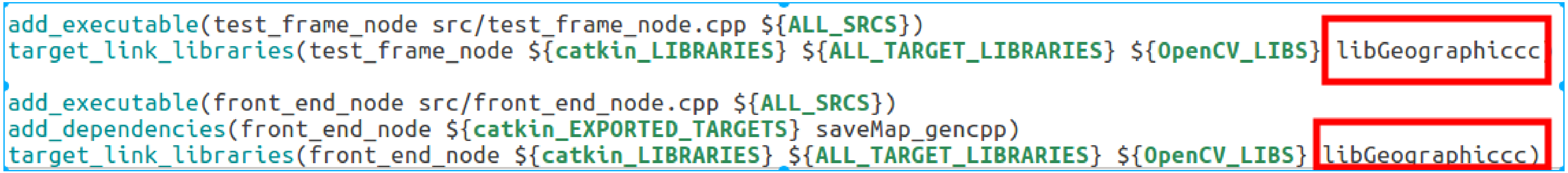
0 编译

CmakeList 修改：

4. LibGeographiccc 冲突



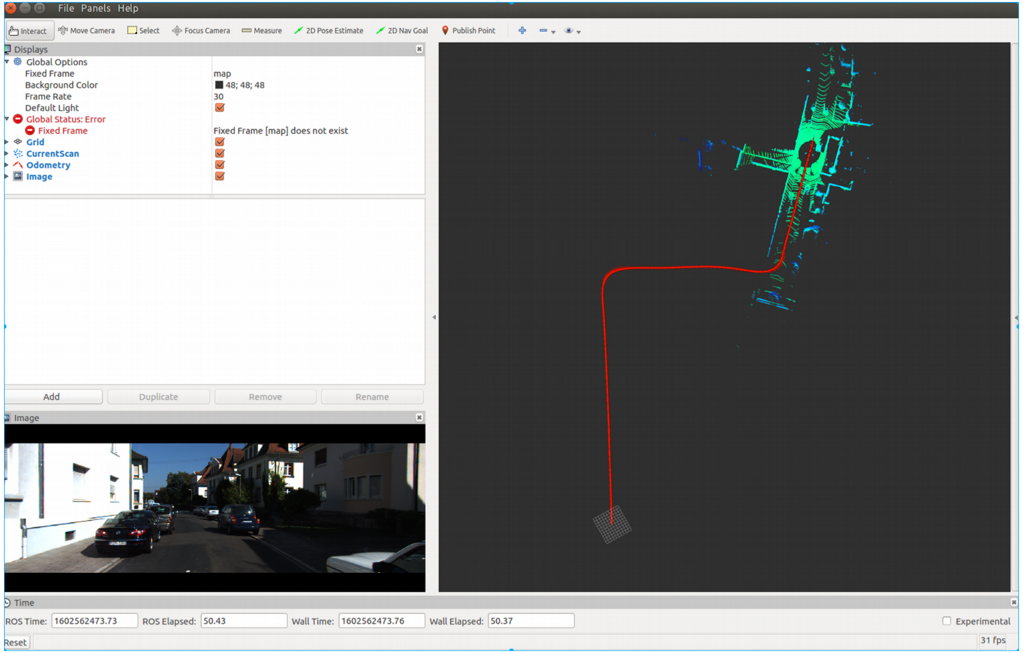
```
#include(cmake/geographic.cmake)
include_directories(${PROJECT_SOURCE_DIR}/third_party/GeographicLib/include/)
```



```
add_executable(test_frame_node src/test_frame_node.cpp ${ALL_SRCS})
target_link_libraries(test_frame_node ${catkin_LIBRARIES} ${ALL_TARGET_LIBRARIES} ${OpenCV_LIBS} libGeographiccc)

add_executable(front_end_node src/front_end_node.cpp ${ALL_SRCS})
add_dependencies(front_end_node ${catkin_EXPORTED_TARGETS} saveMap_gencpp)
target_link_libraries(front_end_node ${catkin_LIBRARIES} ${ALL_TARGET_LIBRARIES} ${OpenCV_LIBS} libGeographiccc)
```

1 作业一

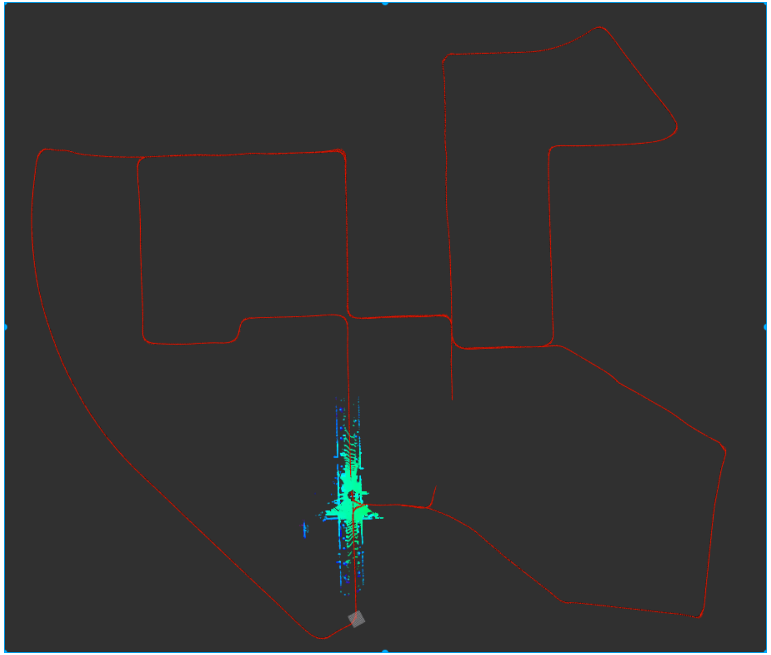
test_frame.launch



Global Options
Fixed Frame: map
Background Color: ■ 48; 48; 48
Frame Rate: 30
Default Light: ☑
Global Status: Error
Fixed Frame: Fixed Frame [map] does not exist
Grid: ☑
CurrentScan: ☑
Odometry: ☑
Image: ☑

ROS Time: 1602562473.73 ROS Elapsed: 50.43 Wall Time: 1602562473.76 Wall Elapsed: 50.37 □ Experimental

31 fps

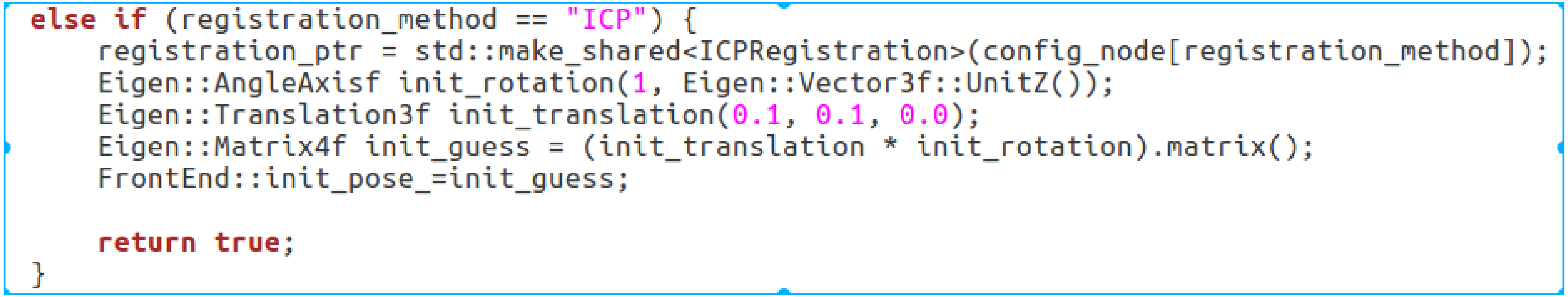


2 作业二

step1: ICP 增加 icp_registration.cpp 与 .hpp，

step2: 在 front_end.cpp InitRegistration 中增加选项

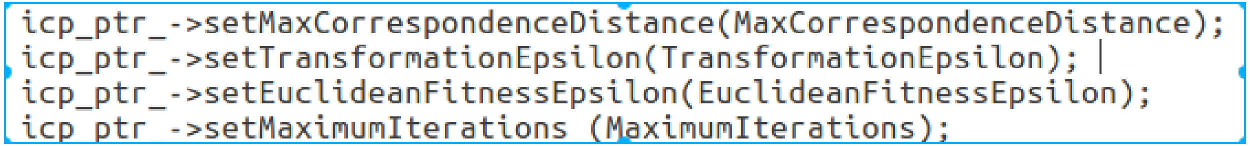
。



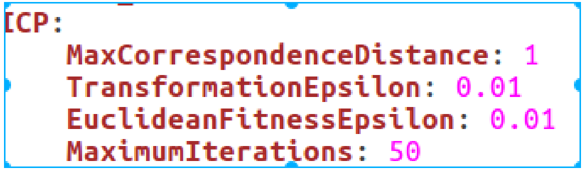
```
else if (registration_method == "ICP") {
    registration_ptr = std::make_shared<ICPRegistration>(config_node[registration_method]);
    Eigen::AngleAxisf init_rotation(1, Eigen::Vector3f::UnitZ());
    Eigen::Translation3f init_translation(0.1, 0.1, 0.0);
    Eigen::Matrix4f init_guess = (init_translation * init_rotation).matrix();
    FrontEnd::init_pose_=init_guess;

    return true;
}
```

2 作业二

step3: config 文件配置



```
icp_ptr_->setMaxCorrespondenceDistance(MaxCorrespondenceDistance);
icp_ptr_->setTransformationEpsilon(TransformationEpsilon);
icp_ptr_->setEuclideanFitnessEpsilon(EuclideanFitnessEpsilon);
icp ptr ->setMaximumIterations (MaximumIterations);
```



```
ICP:
    MaxCorrespondenceDistance: 1
    TransformationEpsilon: 0.01
    EuclideanFitnessEpsilon: 0.01
    MaximumIterations: 50
```

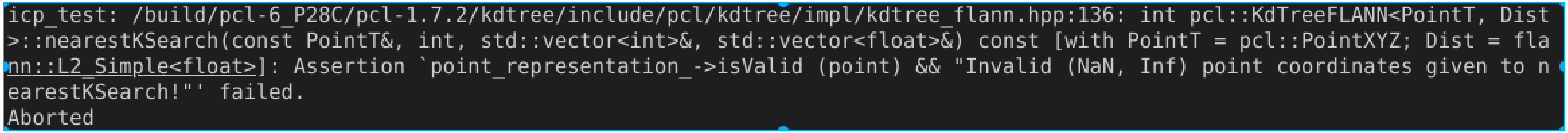
setMaxCorrespondenaceDistance	设置对应点对之间的最大距离
setTransformtionEpsilon	设置两次变化矩阵之间的差值
setEuclideanFitnessEpsilon	设置收敛条件是均方误差和小于阈值
setMaximumIterations	最大迭代次数

3 作业三

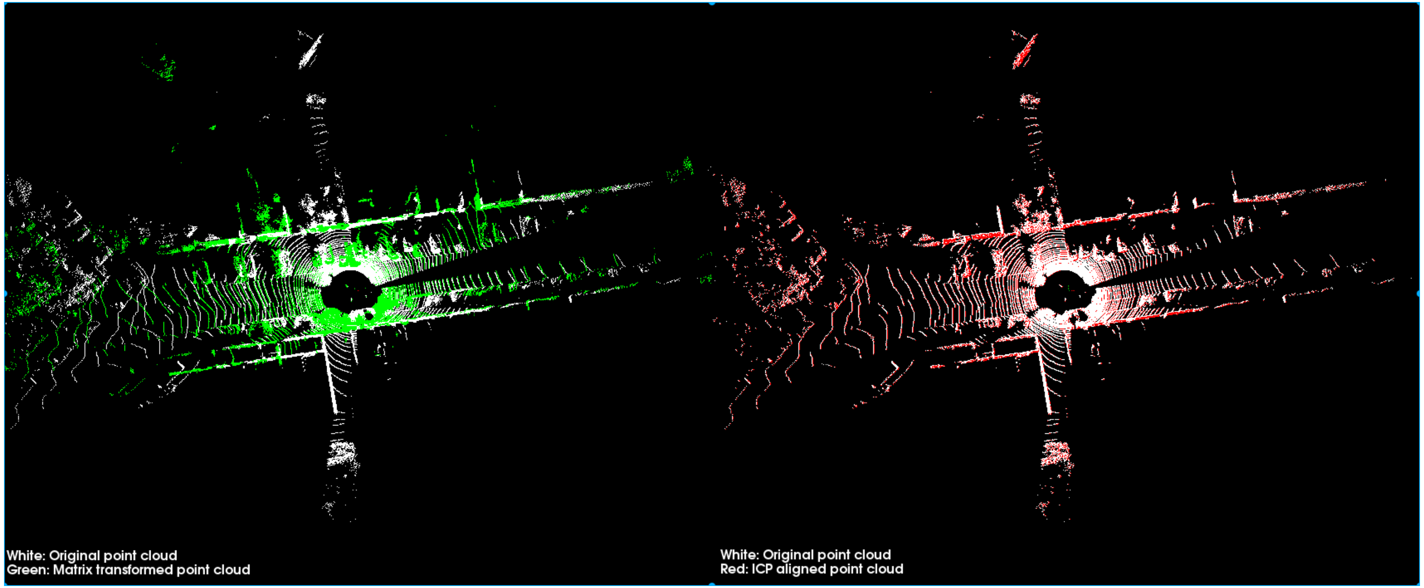
对目标点云 (**MAP**) 构建 **Kd-tree** ， 当前帧以迭代的方式寻找最近点对，并计算变换矩阵。

1. 去质心
2. 构建去质心点对匹配误差模型
3. **SVD** 计算 **R**
4. 根据质心变换方程计算平移向量 **t**

3 作业三



```
icp_test: /build/pcl-6_P28C/pcl-1.7.2/kdtree/include/pcl/kdtree/impl/kdtree_flann.hpp:136: int pcl::KdTreeFLANN<PointT, Dist
>::nearestKSearch(const PointT&, int, std::vector<int>&, std::vector<float>&) const [with PointT = pcl::PointXYZ; Dist = fla
nn::L2_Simple<float>]: Assertion `point_representation_->isValid (point) && "Invalid (NaN, Inf) point coordinates given to n
earestKSearch!"' failed.
Aborted
```


White: Original point cloud
Green: Matrix transformed point cloud
White: Original point cloud
Red: ICP aligned point cloud

3　作业二三的结果评价

分段：　**evo_rpe kitti ground_truth.txt laser_odom.txt -r trans_part --delta 100 --plot --plot_mode xyz**

整体：　**evo_ape kitti ground_truth.txt laser_odom.txt -r full --plot --plot_mode xyz**

本次作业的结果可见 **NDT** 方法效果最好， **ICP** 及其变种经过各种调参实验效果均不好，主要原因是，作业的前端里程计方法采用的是 **scan2map** 的方式，这种很显然 **NDT** 概率的方式理论上更合理，而 **ICP*** 的方式更适合 **scan-scan** 的扫描匹配。

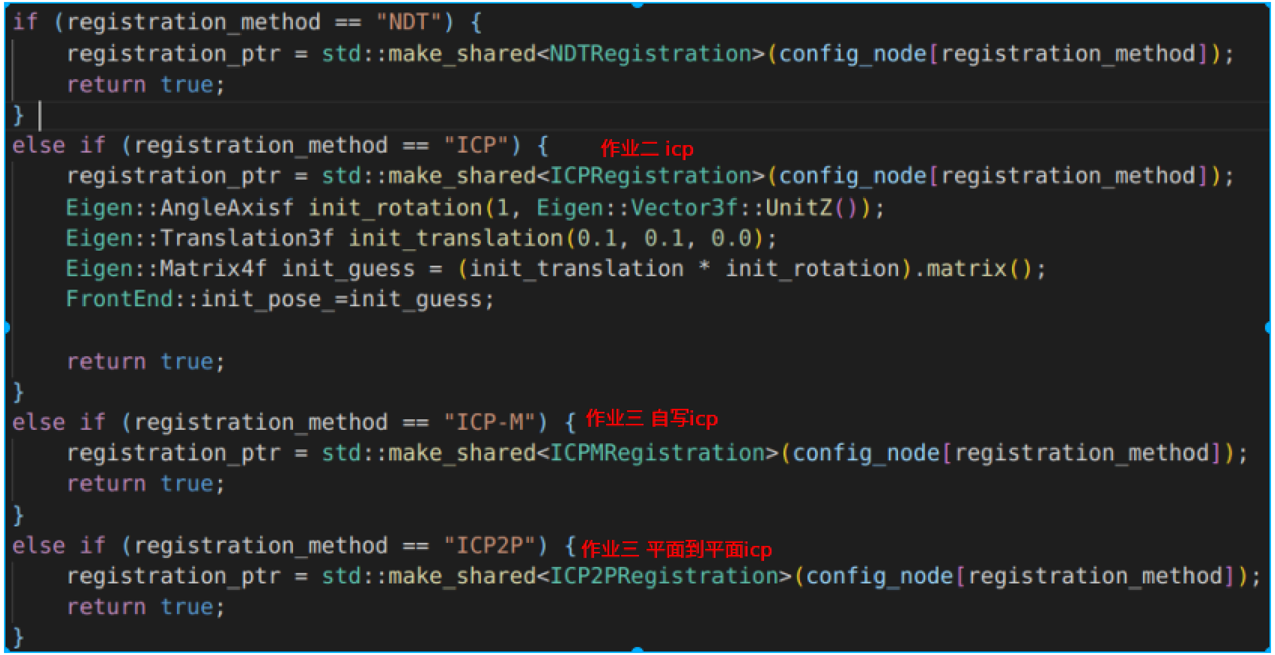
4 附件说明

icp_test

迭代ICP测试程序

lidar_localization

作业参考代码



```
if (registration_method == "NDT") {
    registration_ptr = std::make_shared<NDTRegistration>(config_node[registration_method]);
    return true;
}
else if (registration_method == "ICP") {    作业二 icp
    registration_ptr = std::make_shared<ICPRegistration>(config_node[registration_method]);
    Eigen::AngleAxisf init_rotation(1, Eigen::Vector3f::UnitZ());
    Eigen::Translation3f init_translation(0.1, 0.1, 0.0);
    Eigen::Matrix4f init_guess = (init_translation * init_rotation).matrix();
    FrontEnd::init_pose_=init_guess;

    return true;
}
else if (registration_method == "ICP-M") { 作业三 自写icp
    registration_ptr = std::make_shared<ICPMRegistration>(config_node[registration_method]);
    return true;
}
else if (registration_method == "ICP2P") { 作业三 平面到平面icp
    registration_ptr = std::make_shared<ICP2PRegistration>(config_node[registration_method]);
    return true;
}
```

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening