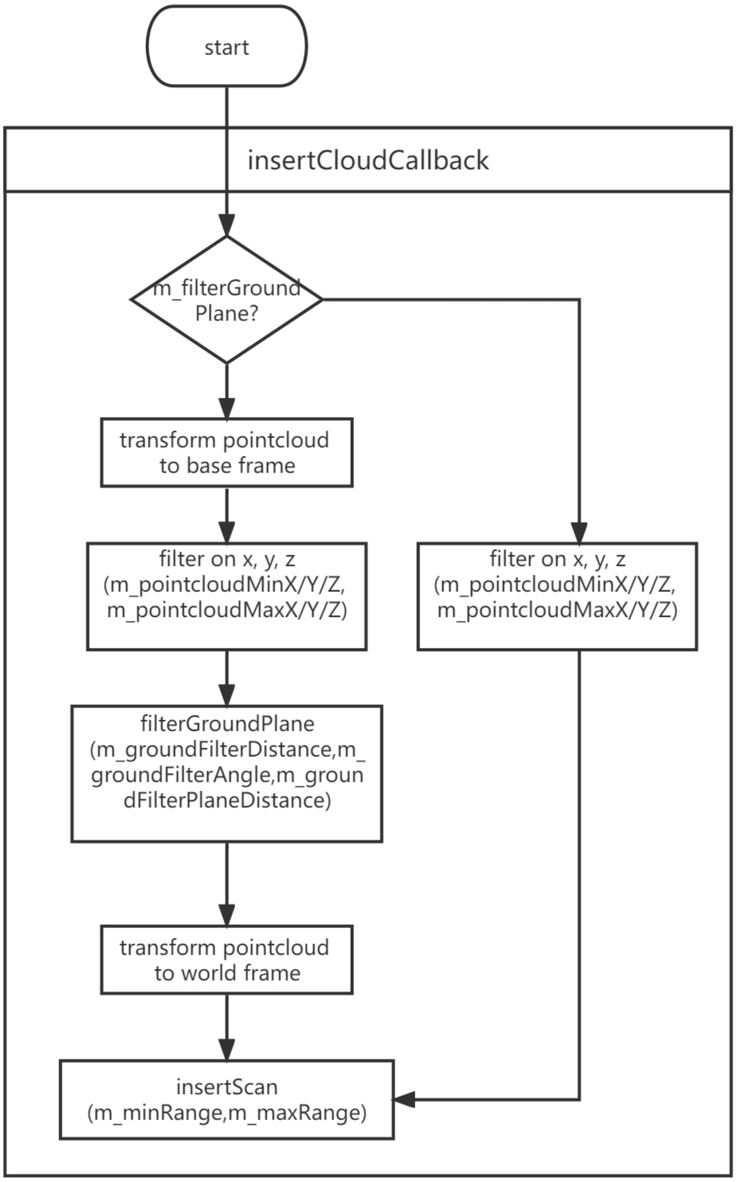
Octmap_Server 简介及其转栅格地图的配置说明



1，简单介绍

Octmap_Server 是一个基于给定位姿和点云的增量式三维八叉树建图方法（只有建图功能，无法定位），同时具有将八叉树地图转化成栅格地图的功能，所有也常被用于将点云地图转成栅格地图，用于机器人导航。整个建图的流程如上图所示，其输入为 sensor::msg::PointCloud2 点云以及位姿 tf(world_frame, base_frame, sensor_frame)，原始点云经过一系列滤波处理后输入到 insertScan 进行八叉树建图。

2，建图配置及说明

2.1，在线建图配置及说明

```
<launch>
<node pkg="octomap_server" type="octomap_server_node" name="octomap_server">
<!--octmap resolution, same for grid map) -->
<param name="resolution" value="0.05" />
<!--/ fixed world map frame (set to 'map' if SLAM or localization running!) -->
<param name="frame_id" type="string" value="map" />
<!--base of the robot for ground plane filtering, set to be the robot center projection frame on groudplane(base_link or base_foorprint_frame)) -->
<param name="base_frame_id" type="string" value="base_link" />
<!-- maximum range(point to its sensor center) to integrate (speedup!) -->
<param name="sensor_model/max_range" value="50.0" />
<!-- whether or not to filter groud plane -->
<param name="filter_ground" value="true" />
<!--z range in world map frame to be converted to grid map -->
<param name="occupancy_min_z" value="0.1" />
<param name="occupancy_max_z" value="3.0" />
<!--z range to prefilter input pointcloud. if filter_ground, work in base_link frame, else in world frame -->
<param name="pointcloud_max_z" value="0.5" />
<param name="pointcloud_min_z" value="-0.1" />
<!--distance of points from plane for RANSAC-->
<param name="ground_filter/distance" value="0.1" />
<!--distance of found plane from z=0 to be detected as ground (e.g. to exclude tables)-->
<param name="ground_filter/plane_distance" value="0.1" />
<!-- data source to integrate (PointCloud2) -->
<remap from="cloud_in" to="/velodyne_points" />
</node>
</launch>
```

建图模块需要实时发布世界坐标系到传感器坐标系的 tf，需要滤除地面的时候，还需要发布机器人本体坐标系到传感器坐标系的 tf

2.2，离线建图配置及说明

```
<launch>
<!-- PCD map server -->
<arg name="pcd_map_file" default="$(find octomap_server)/data/indoor_left1.pcd"/>
<node pkg="pcl_ros" name="pcd_to_pointcloud" type="pcd_to_pointcloud" args="$(arg pcd_map_file) 1.0 _frame_id:=map">
</node>
<node pkg="octomap_server" type="octomap_server_node" name="octomap_server">
<!--octmap resolution, same for grid map) -->
<param name="resolution" value="0.05" />
<!--/ fixed world map frame (set to 'map' if SLAM or localization running!) -->
```

```
<param name="frame_id" type="string" value="map" />
<!--base of the robot for ground plane filtering, set to be the robot center projection frame on groudplane(base_link or base_foorprint_frame)) -->
<param name="base_frame_id" type="string" value="base_link" />
<!-- maximum range(point to its sensor center) to integrate (speedup!) -->
<param name="sensor_model/max_range" value="200.0" />
<!-- whether or not to filter groud plane -->
<param name="filter_ground" value="false" />
<!--z range in world map frame to be converted to grid map -->
<param name="occupancy_min_z" value="-0.5" />
<param name="occupancy_max_z" value="3.0" />
<!--z range to prefilter input pointcloud. if filter_ground, work in base_link frame, else in world frame -->
<param name="pointcloud_max_z" value="1." />
<param name="pointcloud_min_z" value="-0.5" />
<!--distance of points from plane for RANSAC-->
<param name="ground_filter/distance" value="0.1" />
<!--distance of found plane from z=0 to be detected as ground (e.g. to exclude tables)-->
<param name="ground_filter/plane_distance" value="0.1" />
<!-- data source to integrate (PointCloud2) -->
<remap from="cloud_in" to="/cloud_pcd" />
</node>
</launch>
```

离线建图可以看作是一种特殊的在线建图，只是将整个 pcd 地图看作一帧传感器数据，实际坐标系到传感器坐标系就是单位阵，由 pcd_to_pointcloud 节点发布，max_range 等参数需要进行调整。同样地，需要滤除地面的时候，需要发布机器人本体坐标系到传感器坐标系的 tf。

3，离线建图与在线建图优劣对比

建立八叉树和转栅格的过程都会按照给定的高度和距离范围进行点云的过滤，如果点云不准确，将会带来问题。
在线建图：可实时，开启滤除地面时，高度(z)方向的滤波是在单帧上进行的，不会因为建图的高程差而导致高度上的过滤出现偏差，可获得较干净的地图，后期的编辑工作较少；目前的缺点仅在于没有回环功能。

离线建图：优势是不需要回环功能，因为建好的全局地图保证了做到了这一点；缺点在于无法实时展示，因高程差的原因可能造成转换失败：高度范围设置小，转换不完整; 高度范围设置大，导致点云在高度方向上过滤不理想，使得图中有较多地面元素，需要做较大的后期地图编辑工作，甚至出现程序异常退出。

如果在代码中加入回环校正功能，在线建图可满足实际业务需求；或者利用建图数据在建好的地图上运行定位，定位准确的条件下，也可获得足够好的地图。

4，不增加回环功能使用的方法

这是一种受在线方式启发的离线处理方法，需要在建图过程中保存最终的关键帧位姿和点云，运行的时候读取点云和对应的位姿，输入到建图流程即可。对此做了一些修改：

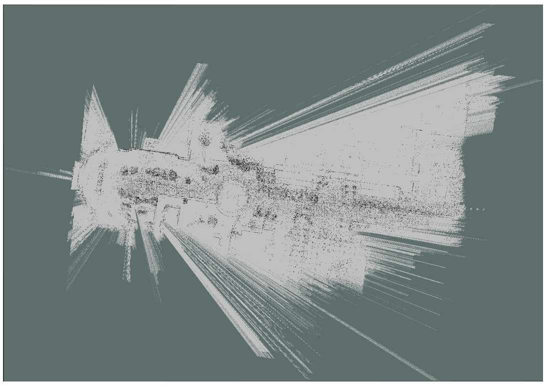
- 增加对 TUM 位姿文件和单帧点云 pcd 文件的读取，并喂给建图模块

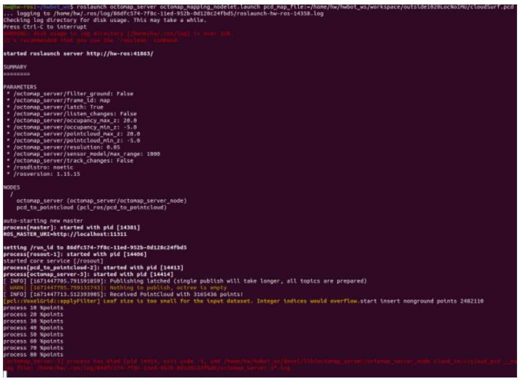
```
void MappingOfflineWithPosesAndPCFrame(const std::string& tum_poses_file, const std::string& lidar_scan_dir, OctomapServer& server, ros::NodeHandle& nh)
{
ROS_INFO("tum_poses_file: %s, lidar_scan_dir: %s", tum_poses_file.c_str(), lidar_scan_dir.c_str());
vector<double> timestamps;
vector<Eigen::Matrix4f> poses;
vector<string> pcd_files;
LoadTUMPosesWithTimestamp(tum_poses_file, timestamps, poses); //加载 TUM 位姿和时间戳
GetFilelistFromDir(lidar_scan_dir, pcd_files, "pcd");//加载单帧 pcd 点云
ROS_INFO(" poses.size %lu and pcd_files.size %lu", poses.size(), pcd_files.size());
const string frame_id = "rslidar";
if(poses.size() != pcd_files.size()){
ROS_ERROR(" poses.size is not equal to pcd_files.size.");
}
OctomapServer::PCLPointCloud::Ptr pc(new OctomapServer::PCLPointCloud);
ros::Publisher trajectory_pub;
nav_msgs::Path trajectory;
trajectory.header.frame_id = "map";
trajectory.header.stamp = ros::Time::now();
geometry_msgs::PoseStamped pose;
pose.header = trajectory.header;
trajectory_pub = nh.advertise<nav_msgs::Path>("tum_poses_trajectory", 10);
ros::WallTime startTime = ros::WallTime::now();
for(size_t i = 0; i < pcd_files.size(); i++){
pcl::io::loadPCDFile(lidar_scan_dir + "/" + pcd_files[i], *pc);
server.insertCloudWithPose(*pc, poses[i], ros::Time(timestamps[i]), frame_id); // 输入到建图模块
pose.pose.position.x = poses[i](0, 3);
pose.pose.position.y = poses[i](1, 3);
pose.pose.position.z = 0.0; //poses[i](2, 3);
trajectory.poses.push_back(pose);
if(trajectory_pub.getNumSubscribers() > 0){
trajectory_pub.publish(trajectory);
}
}
double building_octree_map_elapsed = (ros::WallTime::now() - startTime).toSec();
ROS_INFO("Building Map took %f sec", building_octree_map_elapsed);
}
```

- 处理完一帧点云就进行转栅格地图，速度较慢，遂将该步骤移到建图结束后进行;
- 出于方便考虑，点云按 x,y,z 滤波这一步改到传感器坐标系进行

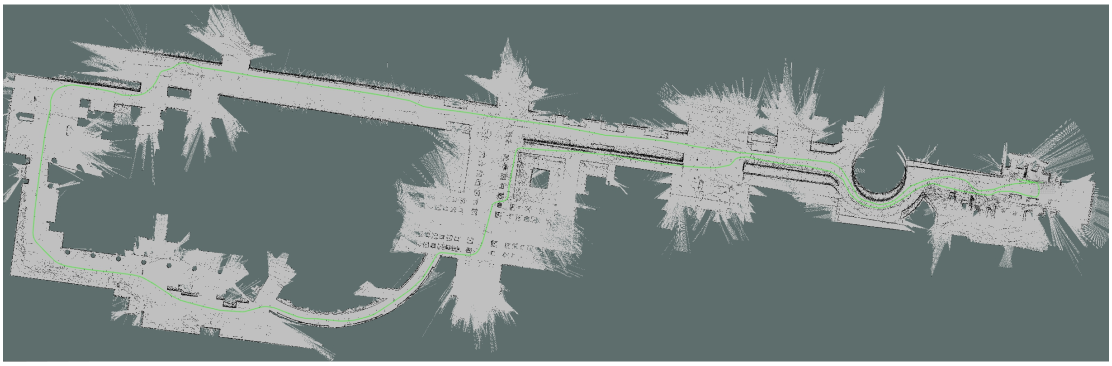
效果对比

对于同一场景采集的同一个建图数据，利用 lio-sam 建图得到 pcd 点云地图，直接将该点云地图转换，当高度范围设置较小时，得到的栅格地图不完整，如下左图所示，高度范围设置较大，则转换失败，如下右图





采用改进后的栅格地图转换方法，地图效果如下，能转换完整，图中绿色为建图轨迹



注：图中重影是由于建图不准导致的，与地图转换无关，用建图后的点云转换栅格地图也有这个问题。