3D 建图功能使用与讲解

1. 功能简介

3D 建图功能主要是通过视觉 slam 结合激光 slam 的方式来实现的，视觉建图与激光雷达建图并行，最终由 rtabmap 进行整合，建图效果在 rviz 中的呈现就是既有 2D 平面地图，又有立体图像效果，因此该功能所呈现的效果为 3D 效果。

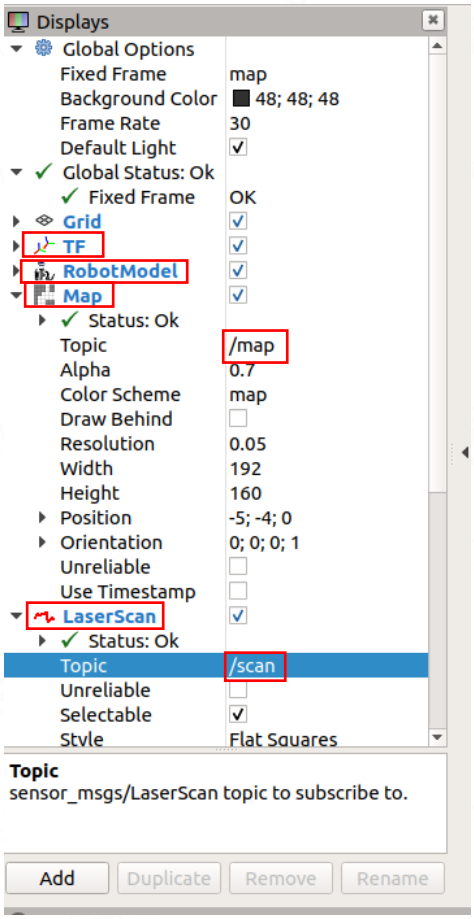
2. 使用方法

1) 打开终端输入启动 3D 建图功能的 launch 文件指令

```
roslaunch turn_on_wheeltec_robot 3d_mapping.launch
```

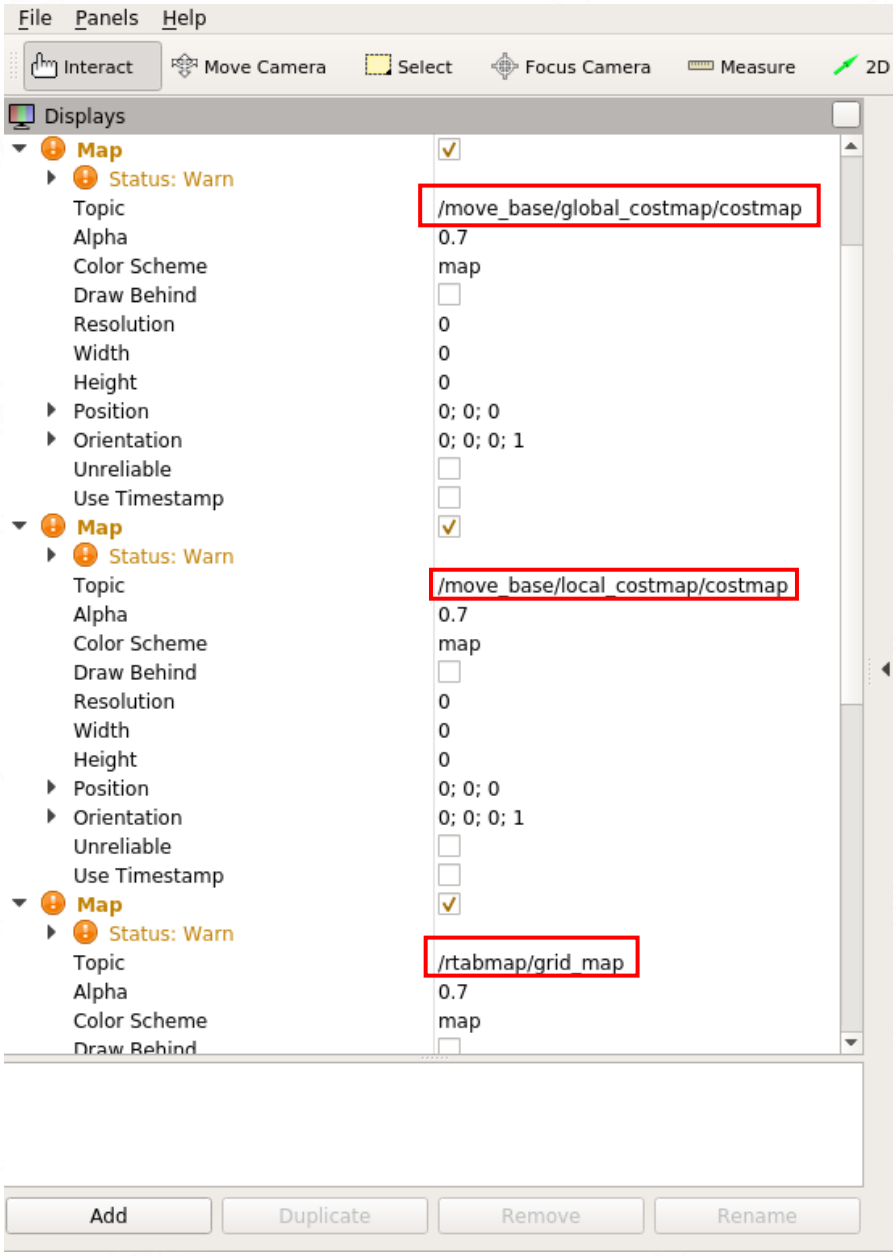
2) 虚拟机端打开 rviz 可视化工具，具体效果如下

首先配置基本的 rviz 可视化选项，选择“TF”、“RobotModel”、“Map”和“LaserScan”，话题对应设置如下



配置 rviz 基本的可视化选项

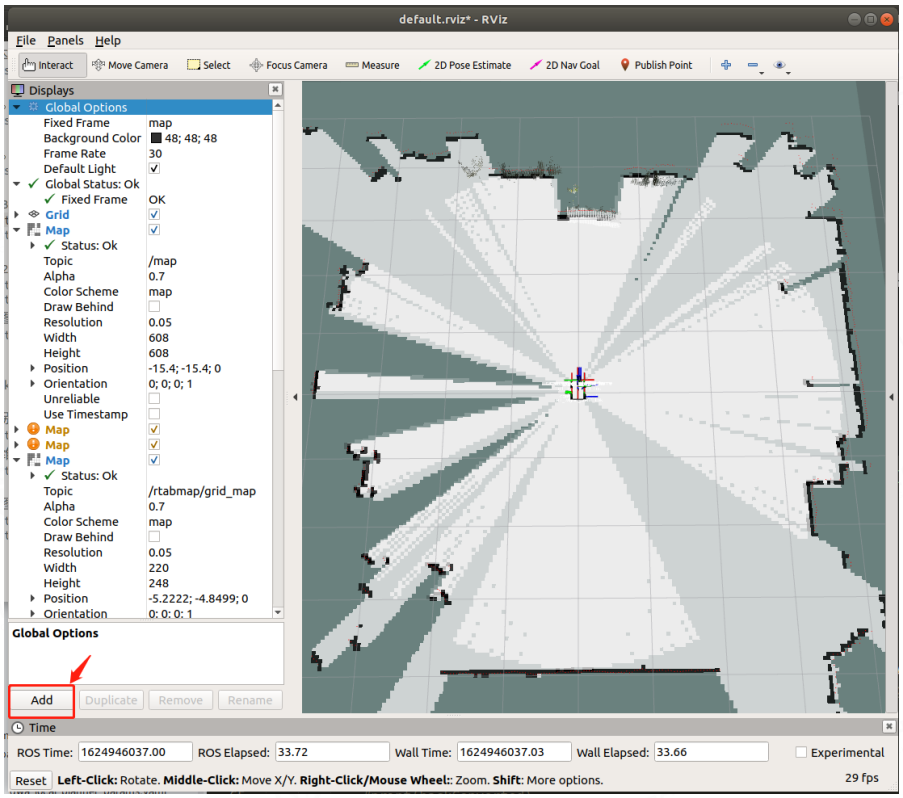
除此之外，还可以添加多个 map 插件辅助



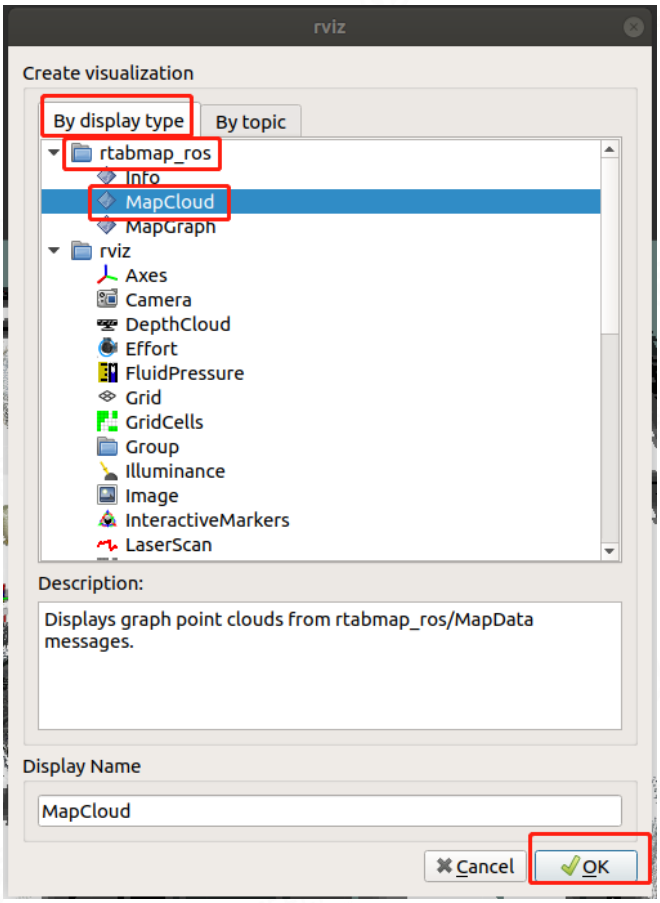
添加多个 map 插件辅助

其中“/move_base/global_costmap/costmap”为全局代价地图，作为进行全局路径规划时的参考，“/move_base/local_costmap/costmap”为局部代价地图所建立的地图主要是为局部路径规划所使用，“/rtabmap/grid_map”是雷达激光扫描建立的占据图。

Rviz 配置除了这些基本配置外，还需要增加一个可以查看摄像头点云信息的选项，点击上图所示 rviz 界面左下角的 ADD 选项，选择 rtabmap_ros 下的 MapCloud，点击 ok 添加，

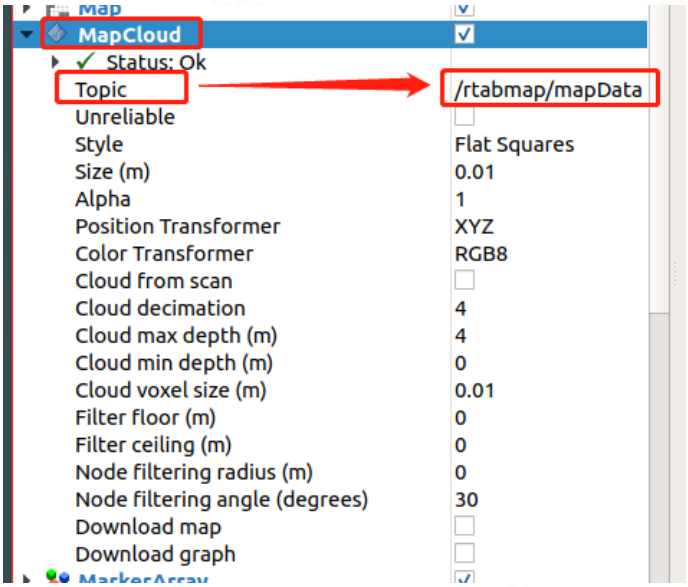


点击左下角“Add”选项



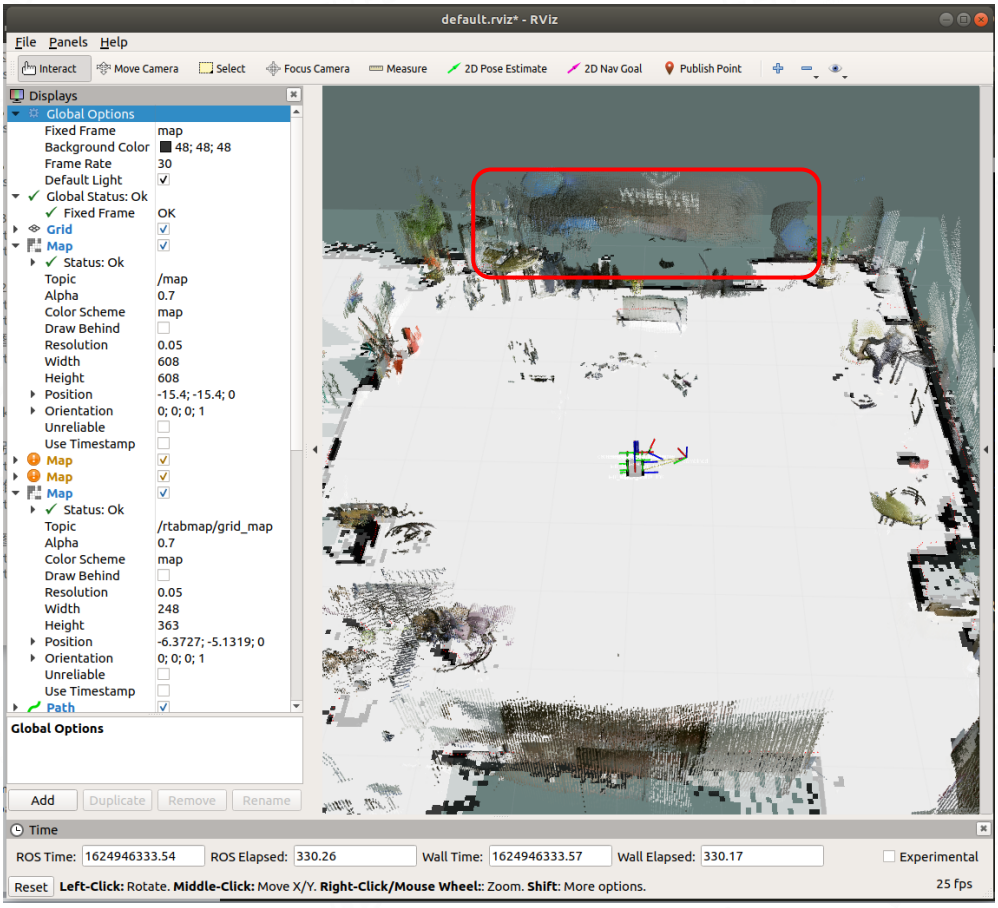
添加 MapCloud 选项

MapCloud 中的 Topic 要选择/rtabmap/mapData，之后 rviz 界面中就会有点云信息（我们所提供的虚拟机镜像都是已经配置好的，用户无需重复配置）。



选择/rtabmap/mapData 话题

可以看到刚打开时，rviz 中显示了两个图层，这两个图层分别就是点云图层和 2D 地图图层，随着小车的运动这两个图层会逐渐完整。



3D 建图完成后效果

3) 打开键盘控制功能控制小车进行完整建图，另外还可以采用蓝牙控制、手柄控制、航模控制或 APP 控制的方式来使小车运动（控制方式详见对应章节，3d_mapping.launch 默认已开启底层节点）

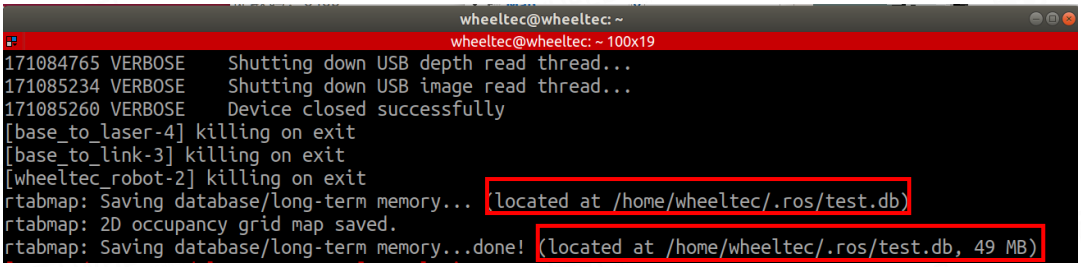
```
roslaunch wheeltec_robot_rc keyboard_teleop.launch          #键盘控制
```

4) 建图完成后输入指令保存地图

```
roslaunch turn_on_wheeltec_robot map_saver.launch
```

3. 注意事项

注意：3D 建图可直接 Ctrl+C 退出并保存地图，无需输入保存地图的指令，终端也会有相应的提示，包括地图数据的保存路径与大小。



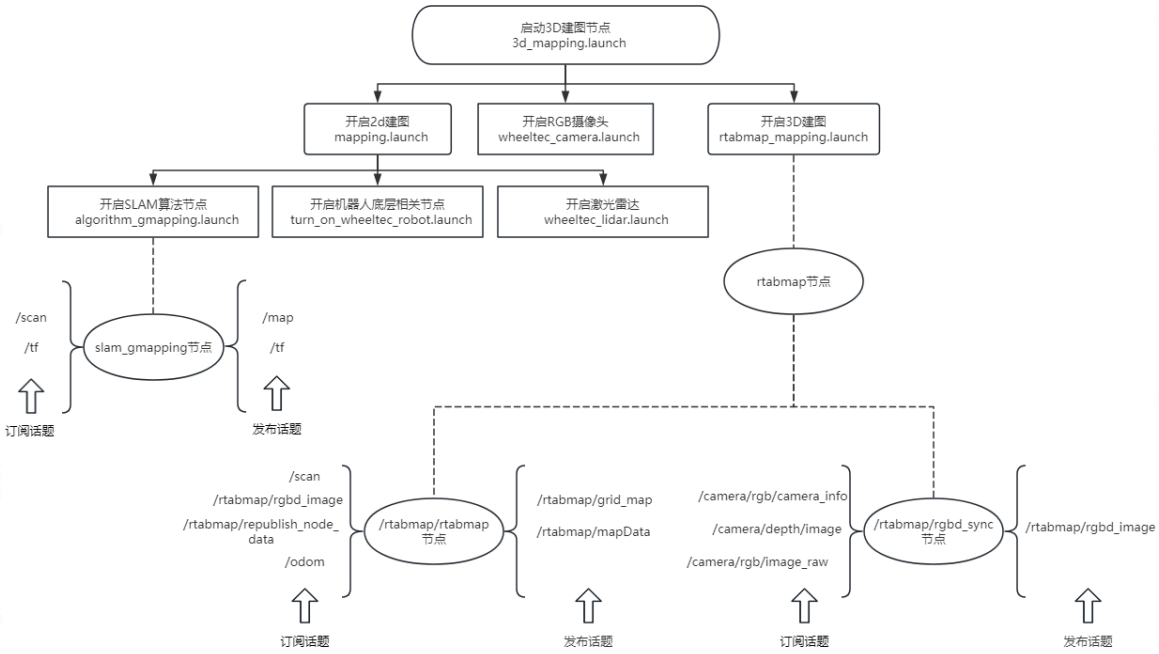
终端提示保存地图的路径与大小

要想修改保存地图的路径，在“turn_on_wheeltec_robot”功能包里的“launch/include”目录下修改“rtabmap_mapping.launch”文件此处即可。

修改 3D 建图的地图保存路径

4. 3D 建图功能程序讲解

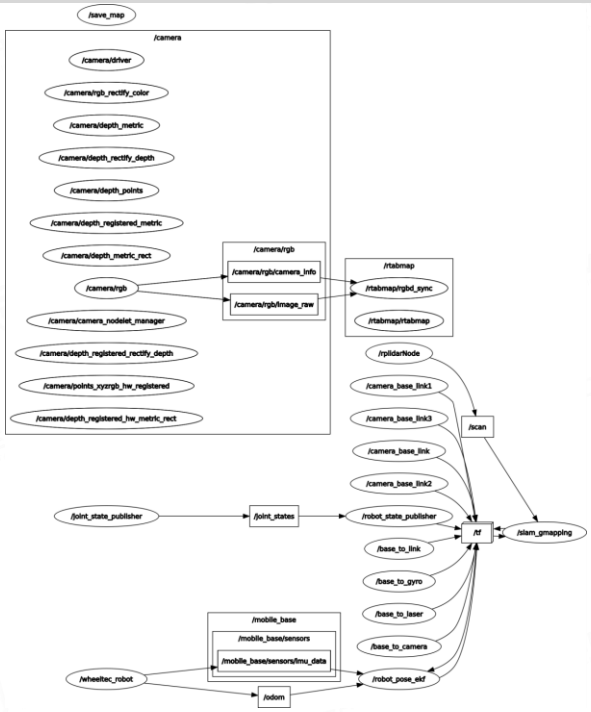
① 3D 建图功能启动框架：



3D 建图功能启动框架图

② 查看 3D 建图功能节点：

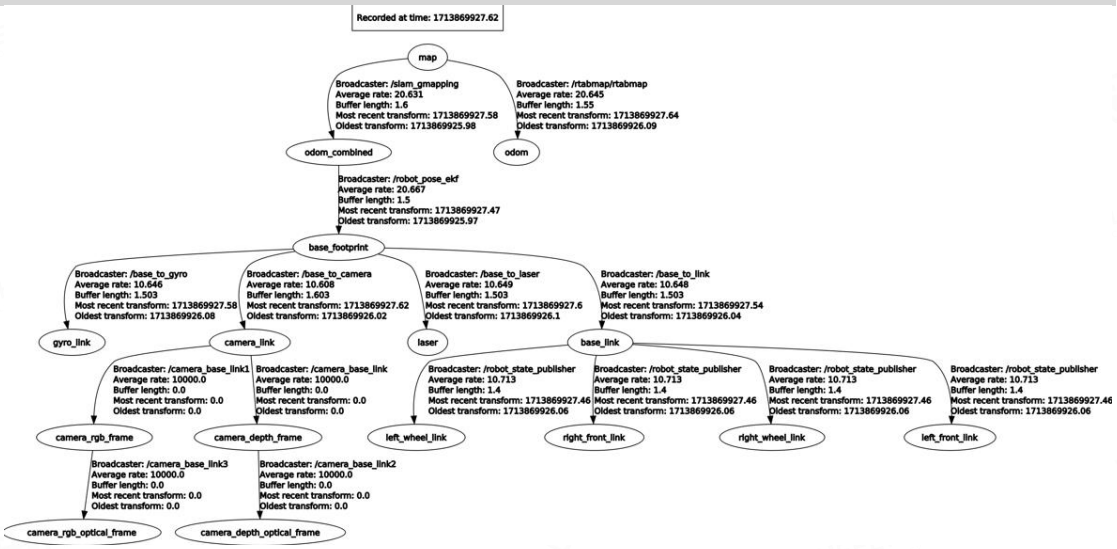
```
rqt_graph
```



3D 建图功能节点关系图

③ 查看 3D 建图功能 TF 树：

```
rosrun rqt_tf_tree rqt_tf_tree
```



3D 建图功能 TF 树

④ **3D 建图功能 launch 文件解析：**

3D 建图节点的开启是靠启动 turn_on_wheeltec_robot 功能包下的 3d_mapping.launch 文件来实现的，这个 launch 文件的内容比较简单，主要是嵌套了三个 launch 文件，分别是开启 2D 建图、开启摄像头以及开启 3D 建图的 launch 文件。

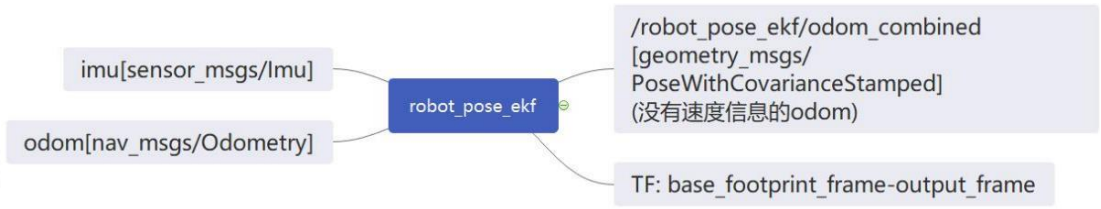
```
<launch>
  <!-- 开启机器人 2d 建图 -->
  <include file="$(find turn_on_wheeltec_robot)/launch/mapping.launch" >
    <!-- 关闭 rtab 提供的定位补偿，使用 2D 建图进行定位补偿 -->
    <arg name="odom_frame_id"   value="odom"/>
  </include>

  <!-- 开启摄像头 -->
  <include file="$(find turn_on_wheeltec_robot)/launch
/wheeltec_camera.launch" />

  <!-- 开启 3d 建图 -->
  <include file="$(find
turn_on_wheeltec_robot)/launch/include/rtabmap_mapping.launch" />
</launch>
```

在 3D 建图过程中，定位是使用里程计（odom）+imu+雷达点云（ekf+建图算法），在 turn_on_wheeltec_robot.launch 文件中，有用于开启卡尔曼滤波算法的节点。这个节点的作用是读取里程计信息然后发布 TF 变换。

```
<!-- 扩张卡尔曼滤波 发布 odom_combined 到 footprint 的 TF,即小车定位 使用
```

```
cartographer 算法时不使用该滤波算法-->
  <include file="$(find
turn_on_wheeltec_robot)/launch/include/robot_pose_ekf.launch" unless="$(arg
repeat)">
    <arg name="is_cartographer" value="$(arg is_cartographer)"/>
  </include>
```

建图与导航的定位一般使用 robot_pose_ekf 功能包进行定位，该功能包订阅 odom 和 imu 话题，发布地图和小车的 TF 坐标关系和经过扩展卡尔曼滤波的 odom 话题。



Robot_pose_ekf 节点话题关系

RTAB-Map (Real-Time 外观- based Mapping)是一种基于全局贝叶斯环路闭合检测器的 RGB-D 图 SLAM 方法。关于 rtabmap 的详细介绍可以查看相应的 ROS wiki 页面：http://wiki.ros.org/rtabmap。

在 3D 建图中，起关键作用的主要是 rtabmap_mapping.launch 文件：

首先定义了 3D 地图数据的保存路径为 home 目录下隐藏文件夹.ros，RTAB-Map 数据库的默认位置是“~/.ros/rtabmap.db”，工作空间也被设置为“~/.ros”，在我们的源码中将数据库的位置设置成了~/.ros/test.db。地图数据为一个 db 文件，无法直接查看，可用专用软件打开，这里不做介绍。

```
<arg name="database_path" default="~/.ros/test.db"/>
```

接下来使用了 rtabmap 功能包提供的 nodelet 中的 rtabmap_ros/rgbd_sync，它的作用是将 RGB、depth 和 camera_info 消息同步到单个消息中，然后可以使用 subscribe_rgbd 使 rtabmap 节点订阅该消息，这样我们就可以确保图像正确地同步在一起。有些相机的深度图和 RGB 图不是同步的，例如 AstraPro 相机，那么下面的参数 approx_sync 就需要设置为 true，

```
<node pkg="nodelet" type="nodelet" name="rgbd_sync" args="standalone
rtabmap_ros/rgbd_sync" output="screen">
      <remap from="rgb/image"       to="/camera/rgb/image_raw"/>
      <remap from="depth/image"     to="/camera/depth/image"/>
      <remap from="rgb/camera_info" to="/camera/rgb/camera_info"/>
```

```
      <remap from="rgbd_image"      to="rgbd_image"/> <!-- output -->
      <remap from="rtabmap/get_map_data" to="get_map_data"/>

      <!-- Should be true for not synchronized camera topics
           (e.g., false for kinectv2, zed, realsense, true for xtion,
kinect360)-->
      <param name="approx_sync"       value="true"/>
    </node>
```

接下来就开启 rtabmap_ros 功能包中最主要的节点：rtabmap，是 RTAB-Map 核心库的封装，也就是在检测到循环闭合时增量构建和优化地图的图形的地方。该节点的在线输出是带有最新添加到地图中的数据的本地图。要获得环境的 3D 点云或 2D 占用网格，可以订阅 cloud_map 或 grid_map 主题。--delete_db_on_start 参数使 rtabmap 在启动时删除数据库，因此我们每次启动 3D 建图时，之前所建立的地图数据是会被清除的，

```
<node name="rtabmap" pkg="rtabmap_ros" type="rtabmap" output="screen" args="--
delete_db_on_start">
```

其他参数的作用与设置方法也可以在 rtabmap_ros 功能包的 ROS wiki 页面中进行查看：http://wiki.ros.org/rtabmap_ros，这里同样不做描述。

简单来说，3D 建图的过程可以概括为：小车通过激光雷达、里程计等提供定位信息，而车身上的摄像头提供点云，rtabmap 的作用就是根据定位将这些点云信息做一个拼接。