

纯视觉建图功能使用与讲解

1. 功能简介

纯视觉功能顾名思义，只使用了视觉 slam 来进行地图的构建，没有使用激光雷达，因此在可视化界面中呈现的地图仅有图像点云信息，且通过图像点云信息来进行障碍物的识别。

2. 使用方法

使用前需要确保上位机连接小车 WiFi，并已经 SSH 远程登录到小车端。

1) 打开终端输入启动纯视觉建图的指令

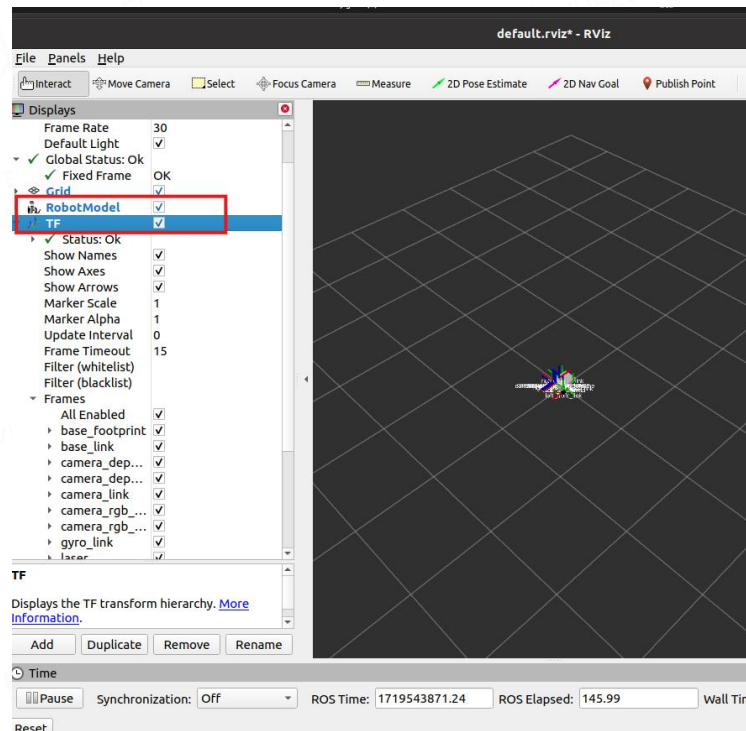
```
roslaunch turn_on_wheeltec_robot pure3d_mapping.launch
```

2) 在上位机端打开 rviz 可视化工具

```
rviz
```

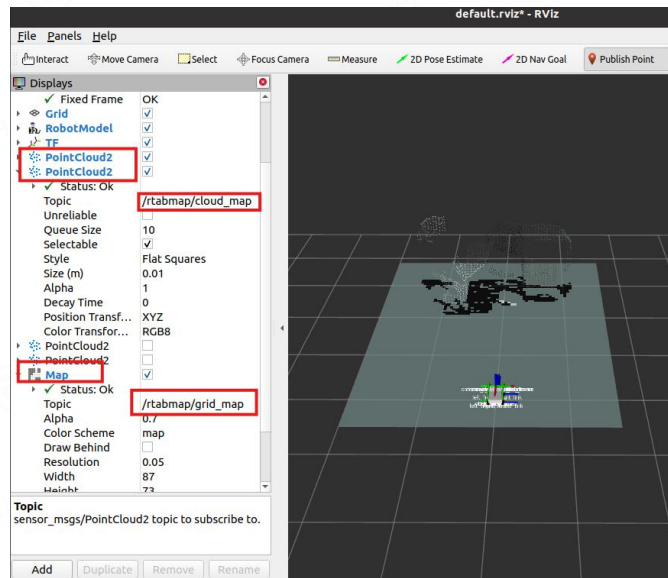
开始配置 rviz 可视化选项：

首先配置机器人基础的可视化选项，在 rviz 中的 Add 按键中的 By display type 选项通过可视化类型的形式添加可视化选项，如下图添加 TF、RobotModel 等机器人基础可视化选项。



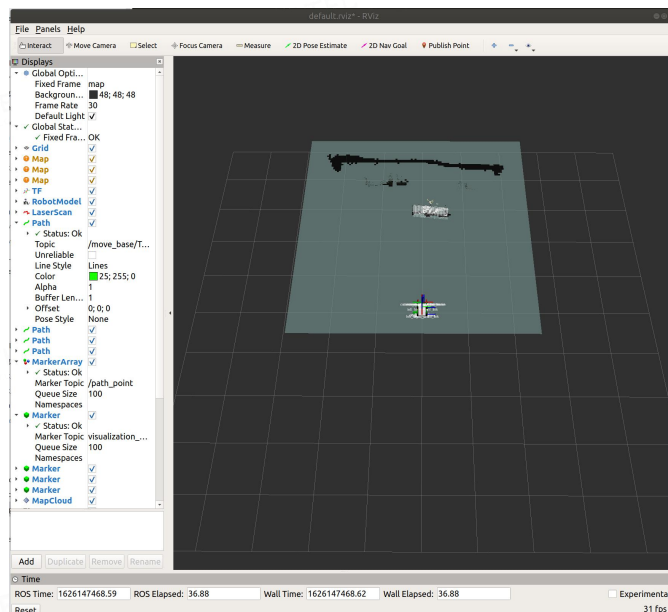
rviz 中配置机器人基础可视化选项

接下来配置建图功能可视化选项：在 rviz 中的 Add 按键中的 By topic 选项通过话题的形式添加可视化选项。如下图所示，在纯视觉建图中重点需要的可视化选项是点云数据 PointCloud2 和地图 Map，其他选项可以视情况自行添加。点云数据可以订阅 /rtabmap/cloud_map 或 /rtabmap/cloud_ground 话题，获得环境点云；地图数据可以订阅 /rtabmap/grid_map，可视化实时生成的地图。



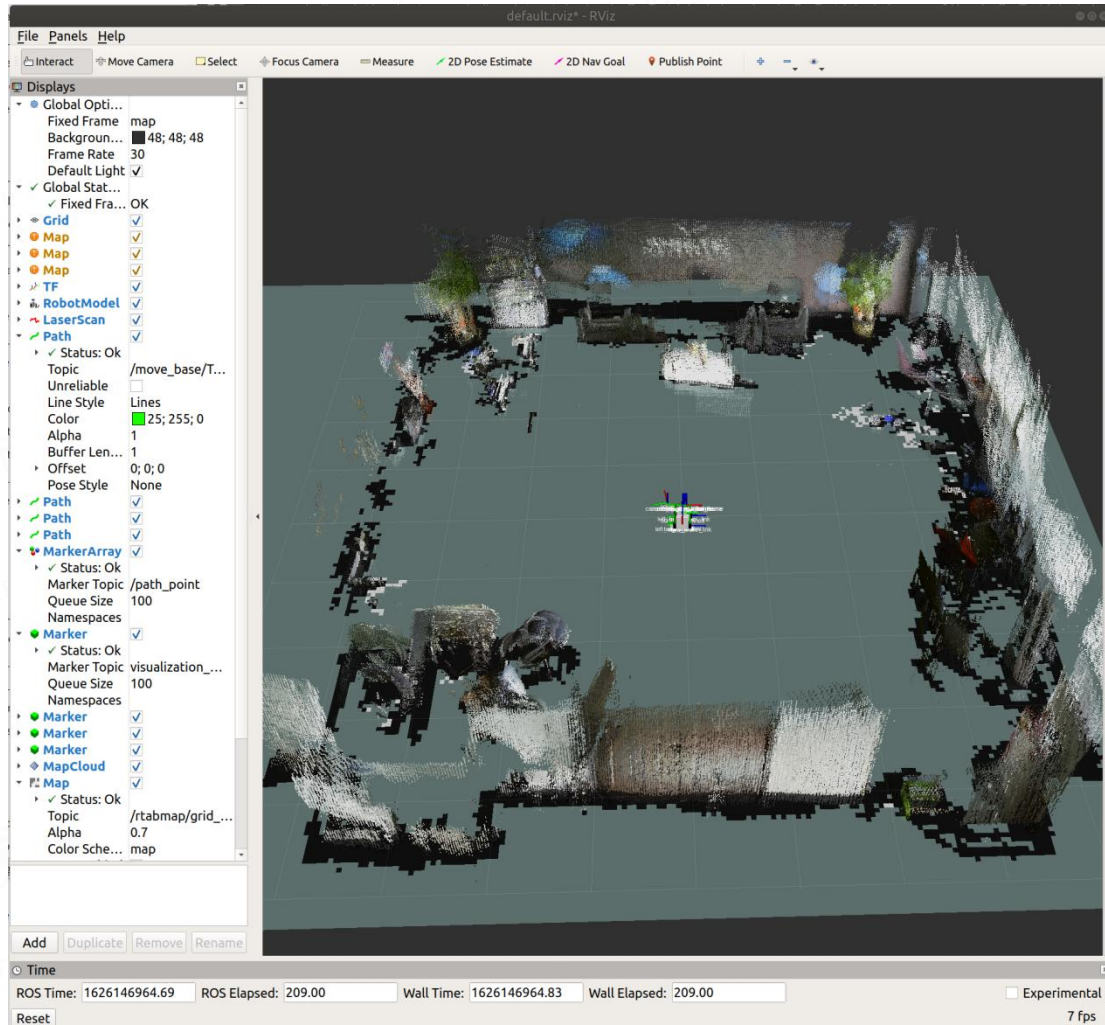
rviz 中配置建图功能可视化选项

完成添加可视化选项后如下图所示，可以观察到界面是比较空的，不像调用了雷达的建图功能一样有大部分地图已经加载出来



纯视觉建图初始状态 rviz 显示界面

- 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，pure3d_mapping.launch 默认已开启底层节点）



纯视觉建图完成效果

- 4) 建图完成后可直接 Ctrl+C 退出并保存地图，终端也会有相应的提示，包括地图数据的保存路径与大小。

```
rtabmap: rtabmap: Saving database/long-term memory... (located at /
~         home/wheeltec/.ros/rtabmap.db)
rtabmap: rtabmap: 2D occupancy grid map saved.
rtabmap: rtabmap: Saving database/long-term memory...done! (located
~         at /home/wheeltec/.ros/rtabmap.db, 40 MB)
rtabmap: rtabmap exited with status 0
```

终端提示保存地图的路径与大小

3. 注意事项

① 建图过程问题

打开rviz后会出现完全没有地图的情况,这是因为小车摄像头距离物体较远,点云信息还没有接收到,可以让小车运动靠近可视物体,观察rviz中有没有图像点云信息的出现。

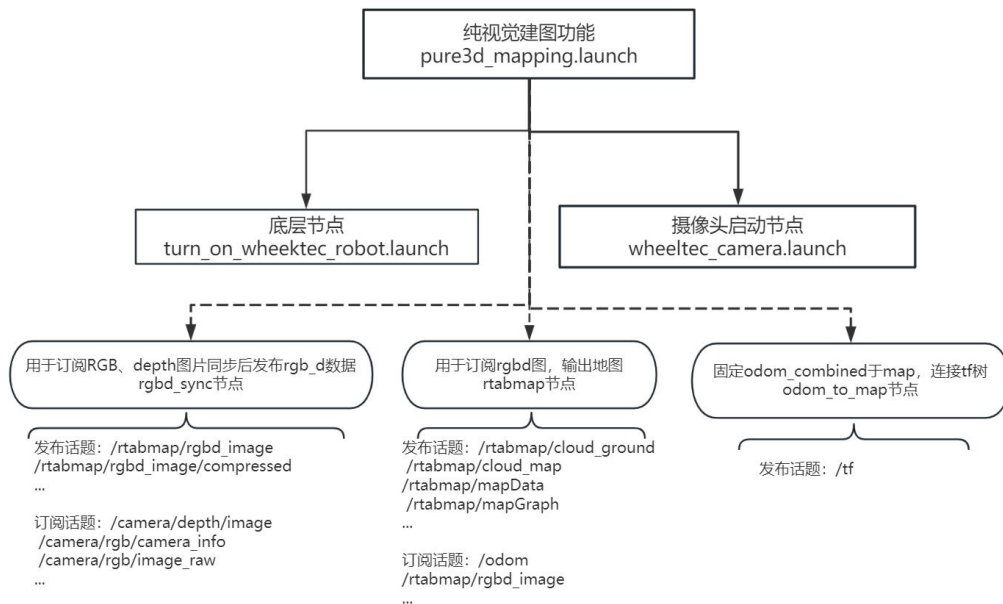
② 建图效果问题

纯视觉建图因为只依赖摄像头与里程计,可能无法达到依赖雷达时的精度,因此运行纯视觉建图功能时,小车运动速度尽量不要过快,也不要过慢,保持匀速前进转弯即可。小车建图起点尽量选择一个比较空旷的地方,以防地图中障碍物在导航时因为膨胀处理与起点重合,导致小车无法进行导航。

4. 功能讲解

纯视觉建图功能是通过启动文件 pure3d_mapping.launch 来开启的

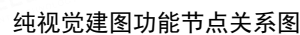
① 启动纯视觉建图功能



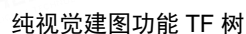
纯视觉建图功能启动框架

② 纯视觉建图功能节点关系

rqt_graph



```
roslaunch rqt_tf_tree rqt_tf_tree
```



`pure3d_mapping.launch` 启动文件在 `turn_on_wheeltec_robot/launch` 路径下，下面就对这个 `launch` 文件做解析，首先是开启机器人底层节点与开启摄像头，接下来通过一个 `tf` 坐标变换将 `odom combined` 与 `map` 进行连接，这是因为纯视

觉建图没有使用到激光 slam 算法，因此这两个 tf 没有通过算法进行连接，需要我们额外进行连接。

```
<!-- 开启机器人底层相关节点 -->
<include file="$(find
turn_on_wheeltec_robot)/launch/turn_on_wheeltec_robot.launch" >
  <arg name="odom_frame_id" value="odom"/>
</include>
<!-- 开启摄像头 -->
<include file="$(find turn_on_wheeltec_robot)/launch/wheeltec_camera.launch"
/>
<!-- 固定 odom_combined 于 map，连接 tf 树 -->
<node pkg="tf" type="static_transform_publisher" name="odom_to_map" args="0 0 0
0 0 0 map odom_combined 100" />
```

接下来就是调用 rtabmap 功能包来进行图像点云的处理，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>。

在下面的代码中使用了 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

```
<!-- rtabmap -->
<group ns="rtabmap">
  <!-- 订阅 RGB、depth 图片同步后发布 rgb_d 数据 -->
  <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，因此我们每次启动纯视觉建图时，之前所建立的地图数据是会被清除的。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。

```
<!-- rtabmap 节点，订阅 rgb_d 图，输出地图 -->
  <!-- 参数 delete_db_on_start 使 3D 地图数据包在程序开始前被删除 -->
  <node name="rtabmap" pkg="rtabmap_ros" type="rtabmap" output="screen"
    args="--delete_db_on_start">
    <param name="frame_id" type="string" value="base_footprint"/>

    <!-- 不订阅 RGB、depth 图片，订阅 rgb_d 数据 -->
    <param name="subscribe_depth" type="bool" value="false"/>
    <param name="subscribe_rgb" type="bool" value="false"/>
    <param name="subscribe_rgbd" type="bool" value="true"/>

    <!-- 订阅 odom 数据用于定位，rgb_d 用于生成地图数据包 -->
    <remap from="odom" to="/odom"/>
    <remap from="rgb_d_image" to="rgb_d_image"/>

    <param name="queue_size" type="int" value="10"/>

    <!-- RTAB-Map's parameters -->
    <param name="RGBD/AngularUpdate" type="string" value="0.01"/>
    <param name="RGBD/LinearUpdate" type="string" value="0.01"/>
    <param name="RGBD/OptimizeFromGraphEnd" type="string" value="false"/>

    <param name="RGBD/NeighborLinkRefining" type="string" value="false"/>
    <param name="RGBD/ProximityBySpace" type="string" value="false"/>
  </node>
</group>
```

整个 `launch` 文件的内容与 3D 建图是非常相似的，两者只区别在有没有使用雷达进行建图上。纯视觉建图源码中对于地图数据的保存路径没有进行设置，因此地图被保存在了默认路径中，即 `~/ros/rtabmap.db`。