

1、手持雷达建图

1、手持雷达建图

1.1、建图

1.1.1、gmapping

1.1.2、hector

1.1.3、karto

1.1.4、cartographer

1.2、地图保存

1.3、查看相关信息

Gmapping: <http://wiki.ros.org/gmapping/>

hector_slam: http://wiki.ros.org/hector_slam

hector_slam/Tutorials: http://wiki.ros.org/hector_slam/Tutorials/SettingUpYourRobot

hector_mapping: http://wiki.ros.org/hector_mapping

karto: http://wiki.ros.org/slam_karto

Cartographer: <https://google-cartographer.readthedocs.io/en/latest/>

Cartographer ROS: <https://google-cartographer-ros.readthedocs.io/en/latest/>

rrt_exploration: http://wiki.ros.org/rrt_exploration

rrt_exploration/Tutorials: http://wiki.ros.org/rrt_exploration/Tutorials

map_server: https://wiki.ros.org/map_server

1.1、建图

安装依赖库

```
sudo apt install ros-melodic-moveit ros-melodic-moveit-visual-tools ros-melodic-kdl-* ros-melodic-joint-state-publisher-gui ros-melodic-trac-ik liborocos-kdl-dev ros-melodic-teleop-twist-keyboard ros-melodic-moveit-resources ros-melodic-navigation ros-melodic-gmapping ros-melodic-hector-slam ros-melodic-slam-karto ros-melodic-robot-state-publisher ros-melodic-geographic-msgs ros-melodic-libuv-* ros-melodic-rtabmap-ros libavformat-dev libavcodec-dev libswresample-dev libswscale-dev libavutil-dev libstdl1.2-dev ros-melodic-libuv ros-melodic-pointcloud-to-laserscan ros-melodic-mbf-msgs ros-melodic-mbf-costmap-core ros-melodic-costmap-converter ros-melodic-bfl ros-melodic-serial ros-melodic-teleop-twist-joy ros-melodic-laser-proc ros-melodic-rosserial-arduino ros-melodic-rosserial-python ros-melodic-rosserial-server ros-melodic-rosserial-client ros-melodic-rosserial-msgs ros-melodic-amcl ros-melodic-map-server ros-melodic-urdf ros-melodic-xacro ros-melodic-interactive-markers ros-melodic-octomap* ros-melodic-joy* ros-melodic-dwa-local-planner ros-melodic-multirobot-map-merge python-catkin-tools python3-dev python3-catkin-pkg-modules python3-numpy python3-yaml build-essential ros-melodic-imu-tools ros-melodic-cartographer*
```

如果是新环境则需要把cartographer的lua文件和launch文件复制到对应的位置

```
cd ~/rplidar_ws/src/transbot_nav/scripts/  
sudo bash create.sh
```

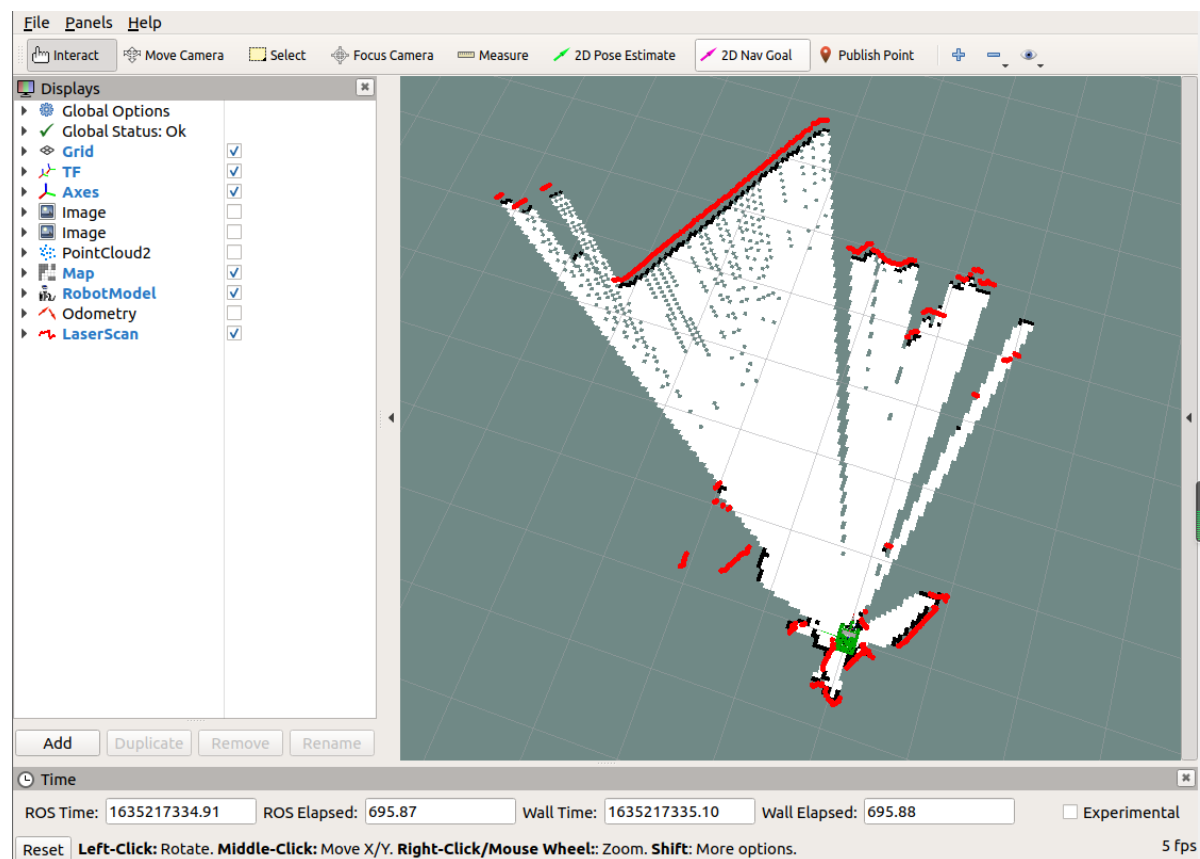
建图启动命令

```
roslaunch transbot_nav laser_map.launch lidar_type:=a1 map_type:=gmapping  
robot_model:=astra
```

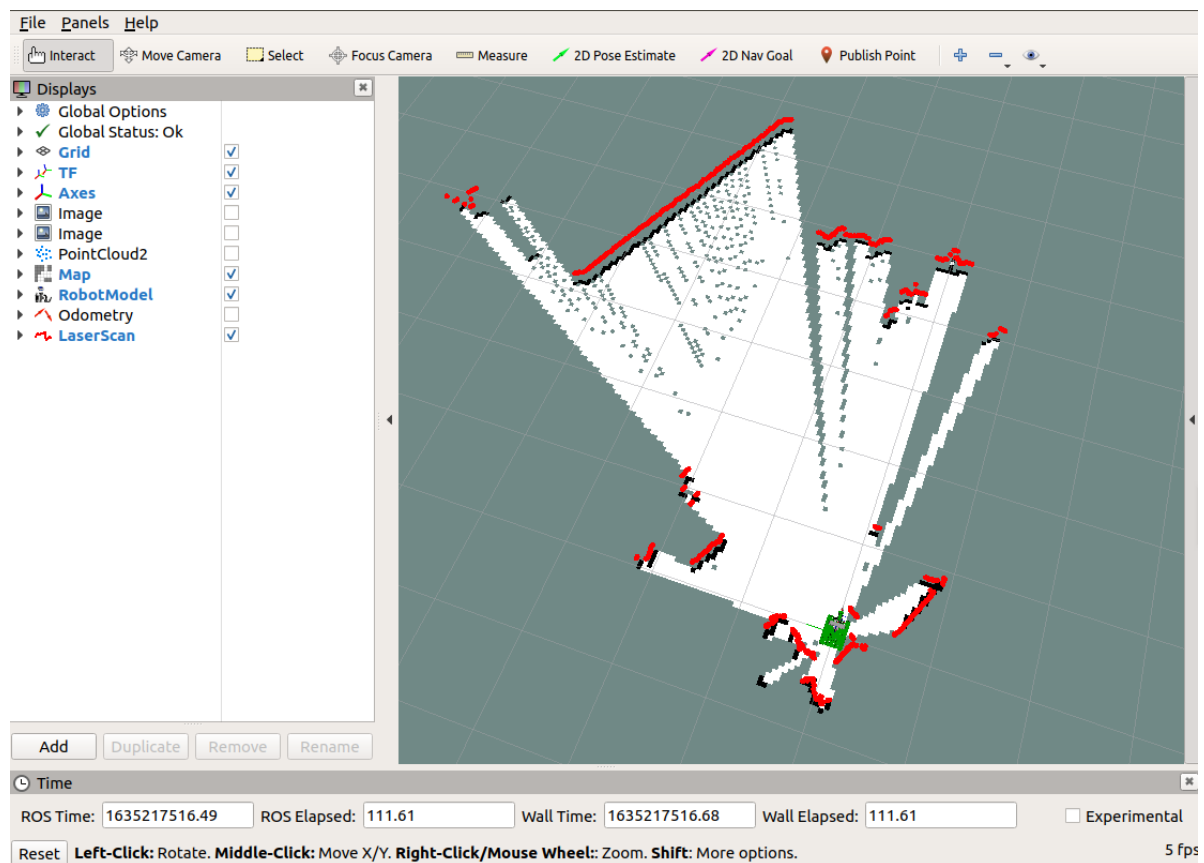
- lidar_type参数: 使用激光雷达的型号: 【a1, a2, a3, s1, s2】, 默认是【a1】。
- map_type参数: 建图算法【gmapping, hector, karto, cartographer】, 默认是【gmapping】。
- robot_model参数: 仿真模型【astra, camera】。

在rviz建图的时候, 如果【LaserScan】报错, 未能加载激光雷达数据; 则选中后, 点击【Remove】移除, 点击【Add】重新添加。选择相应的话题即可。

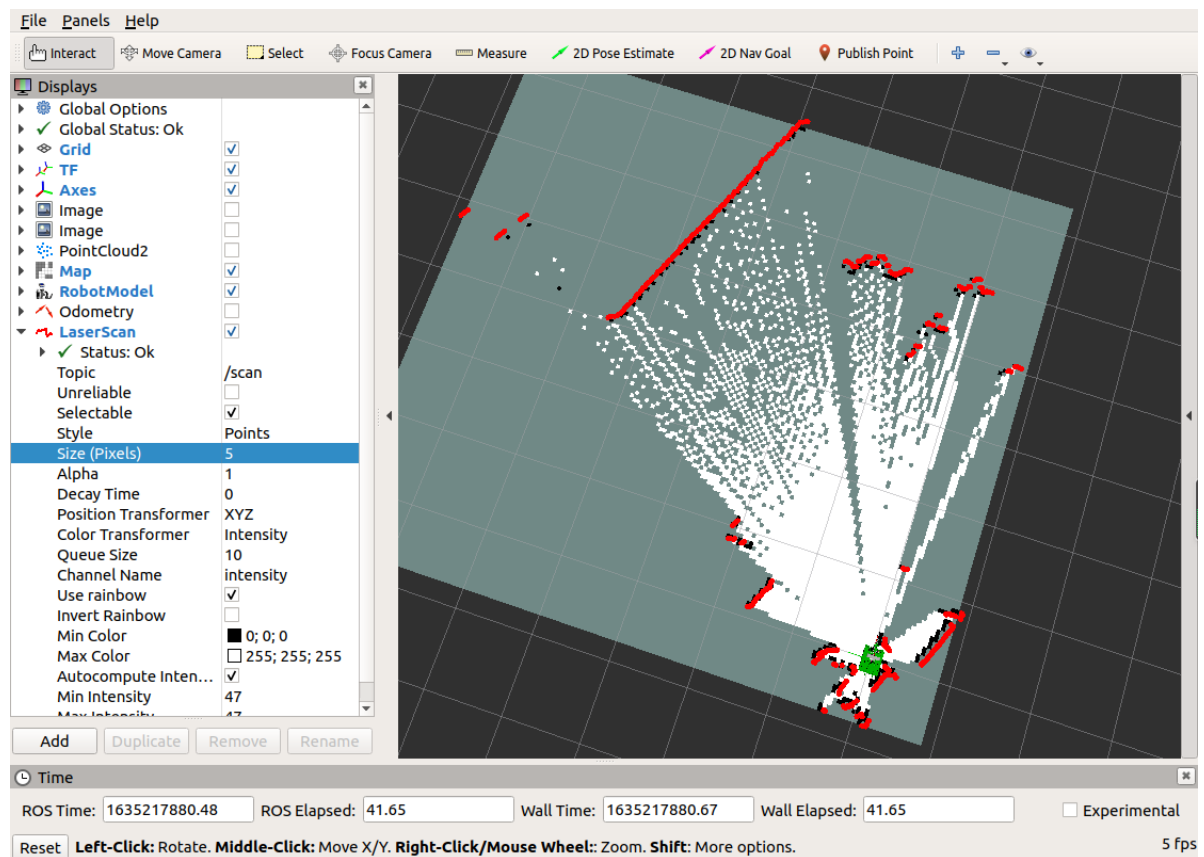
1.1.1、gmapping



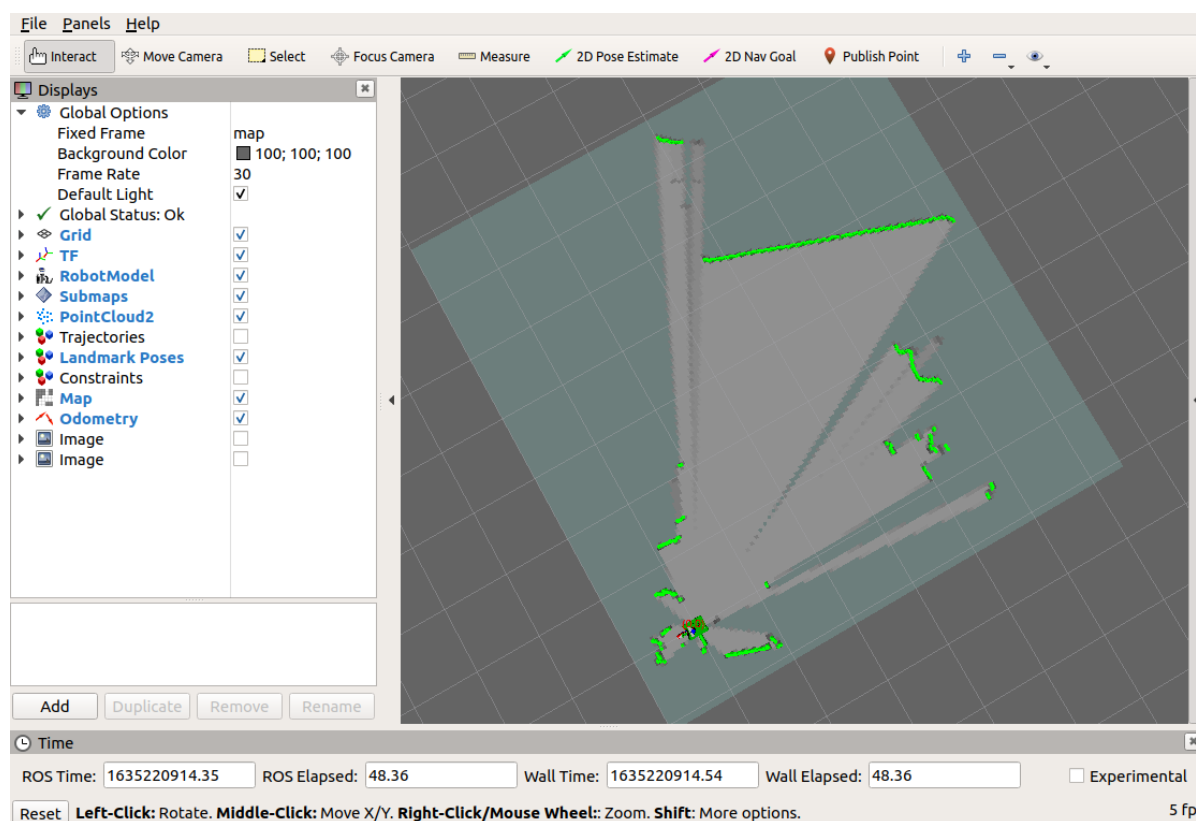
1.1.2、hector



1.1.3, karto



1.1.4, cartographer



1.2、地图保存

几种建图算法保存地图的方式是不同的，

- cartographer: 执行下列命令

```
bash ~/rplidar_ws/src/transbot_nav/maps/carto_map.sh
```

- gmapping,hector,karto: 执行下面命令保存即可。

```
roslaunch map_server map_saver -f ~/rplidar_ws/src/transbot_nav/maps/my_map # 第
一种方式
bash ~/rplidar_ws/src/transbot_nav/maps/map.sh # 第
二种方式
```

地图将被保存到~/rplidar_ws/src/transbot_nav/maps/文件夹下，一个pgm图片，一个yaml文件。

map.yaml

```
image: map.pgm
resolution: 0.05
origin: [-15.4,-12.2,0.0]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

参数解析：

- image: 地图文件的路径，可以是绝对路径，也可以是相对路径
- resolution: 地图的分辨率，米/像素
- origin: 地图左下角的 2D 位姿(x,y,yaw), 这里的yaw是逆时针方向旋转的 (yaw=0 表示没有旋转)。目前系统中的很多部分会忽略yaw值。

- negate: 是否颠倒 白/黑、自由/占用 的意义 (阈值的解释不受影响)
- occupied_thresh: 占用概率大于这个阈值的的像素, 会被认为是完全占用。
- free_thresh: 占用概率小于这个阈值的的像素, 会被认为是完全自由。

1.3、查看相关信息

查看tf树

```
roslaunch rqt_tf_tree rqt_tf_tree
```

节点查看

```
rqt_graph
```