1、Hand-held radar mapping

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

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

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

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、Mapping

Install dependent libraries

```
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-libuvc-* ros-melodic-rtabmap-ros libavformat-dev libavcodec-dev libswresample-dev libswscale-dev libavutil-dev libsdl1.2-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*
```

If it is a new environment, you need to copy the lua file and launch file of cartographer to the corresponding location

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

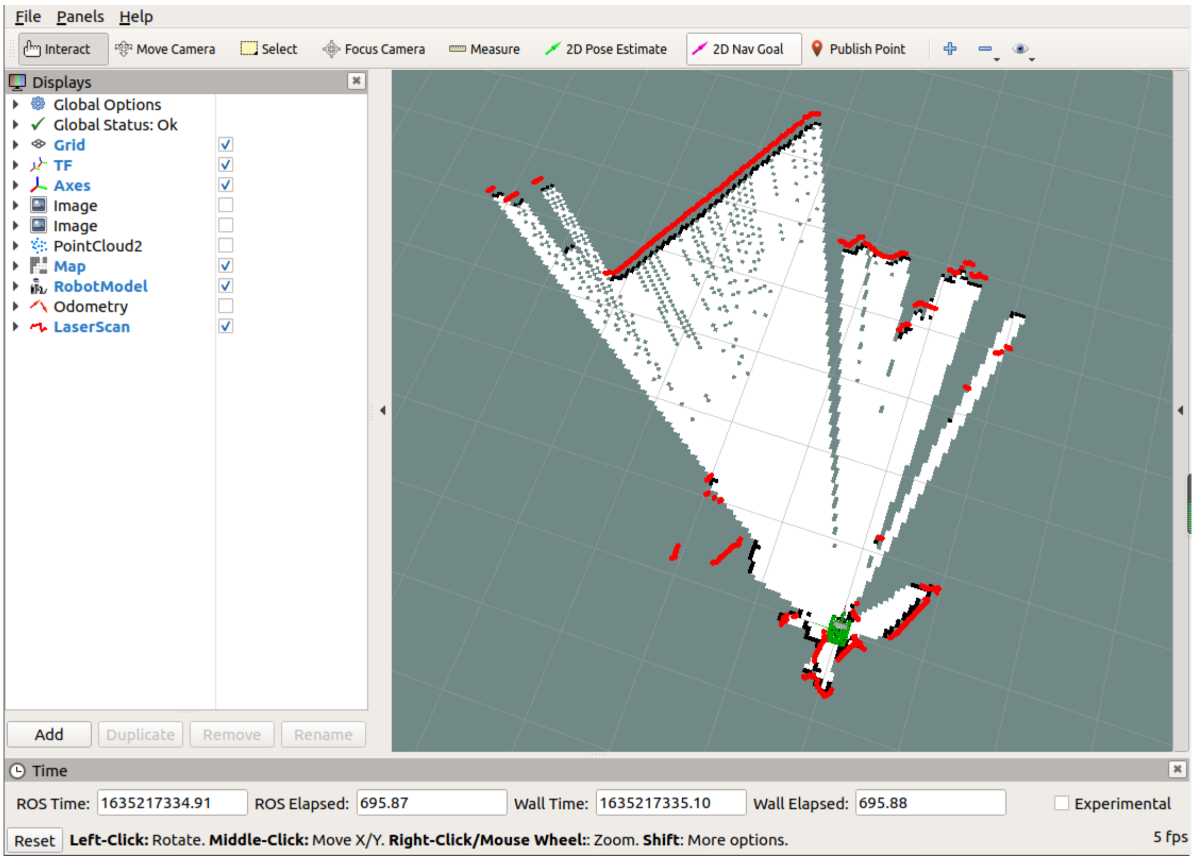
Build map start command

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

- lidar_type parameter: the type of lidar used: [a1, a2, a3, s1, s2], the default is [a1].
- map_type parameter: mapping algorithm [gmapping, hector, karto, cartographer], the default is [gmapping].
- robot_model parameters: simulation model [astra, camera].

  When creating a map in rviz, if [LaserScan] reports an error and fails to load the lidar data; select it, click [Remove] to remove it, and click [Add] to add it again. Just select the corresponding topic.

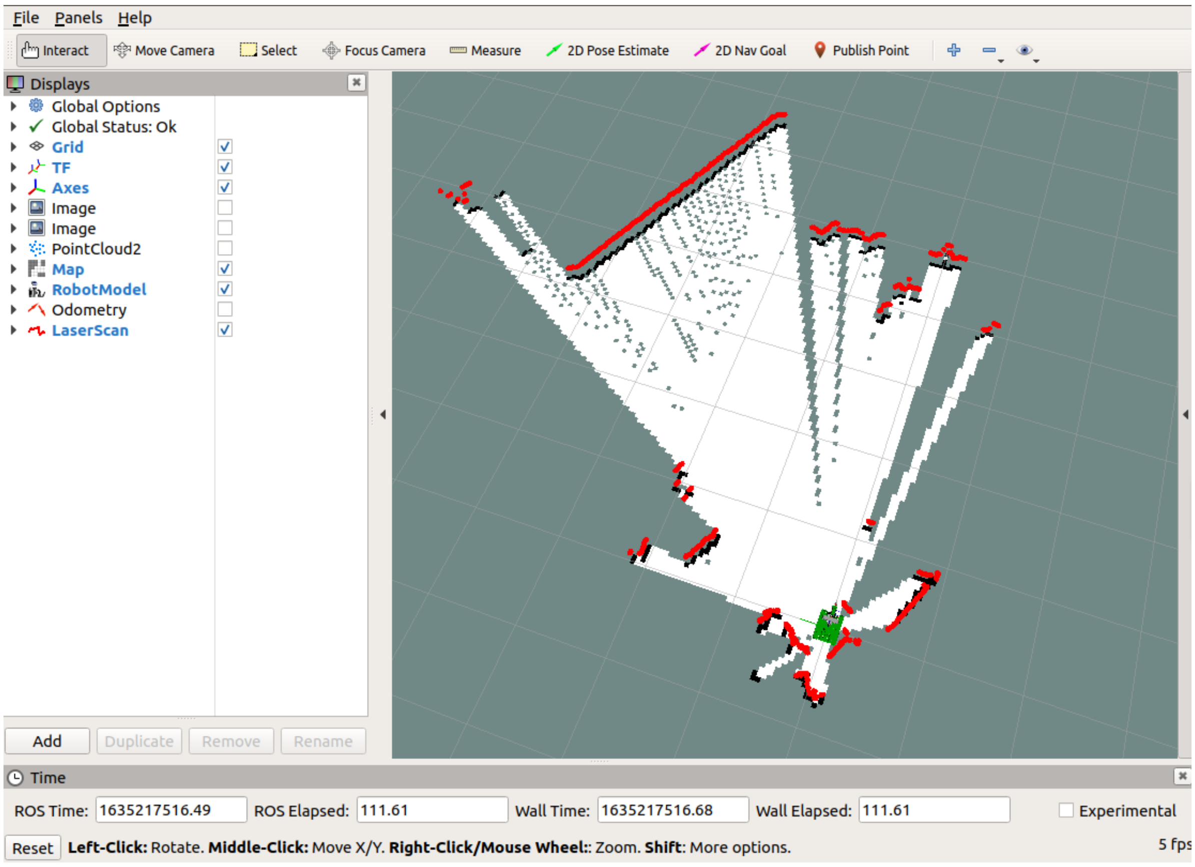
1.1.1、gmapping



ROS Time: 1635217334.91 ROS Elapsed: 695.87 Wall Time: 1635217335.10 Wall Elapsed: 695.88

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

1.1.2、hector

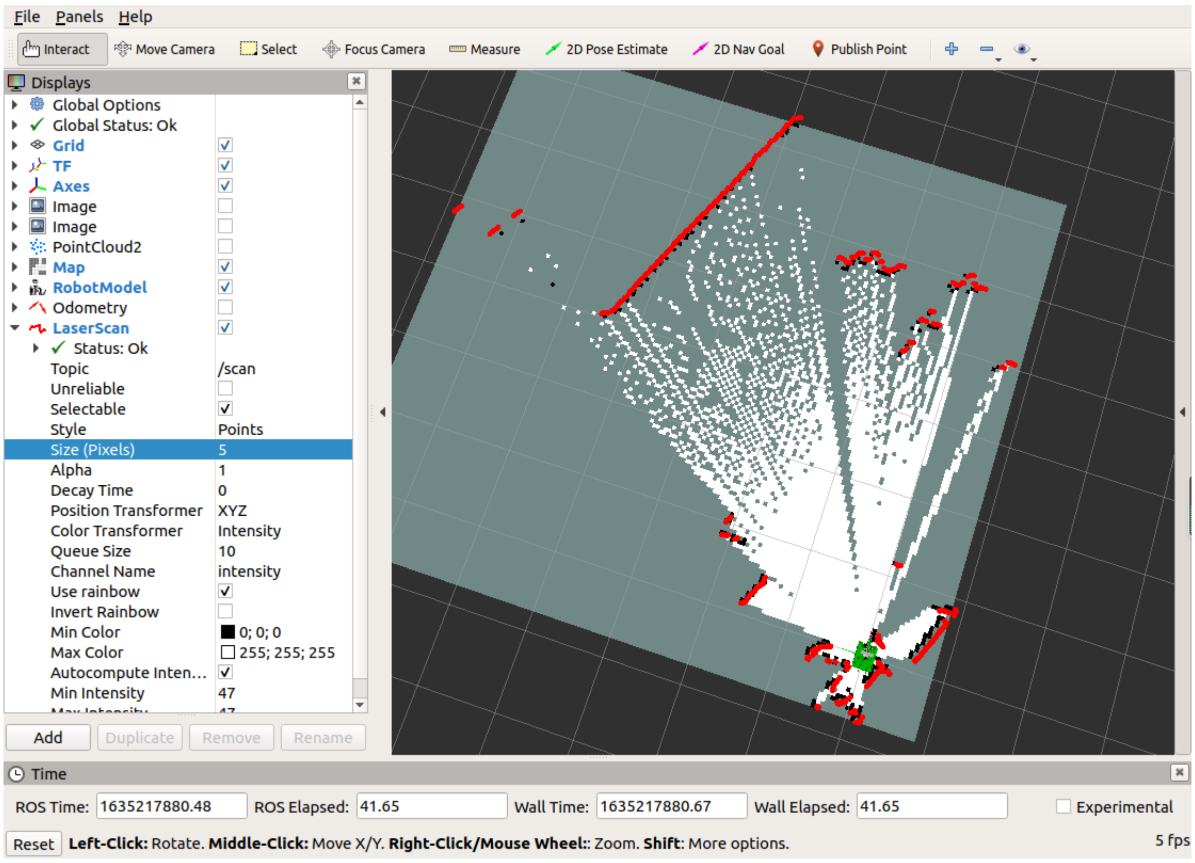


Global Options
Global Status: Ok
Grid ✓
TF ✓
Axes ✓
Image
Image
PointCloud2
Map ✓
RobotModel ✓
Odometry
LaserScan ✓

ROS Time: 1635217516.49 ROS Elapsed: 111.61 Wall Time: 1635217516.68 Wall Elapsed: 111.61 Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 5 fps

1.1.3、karto

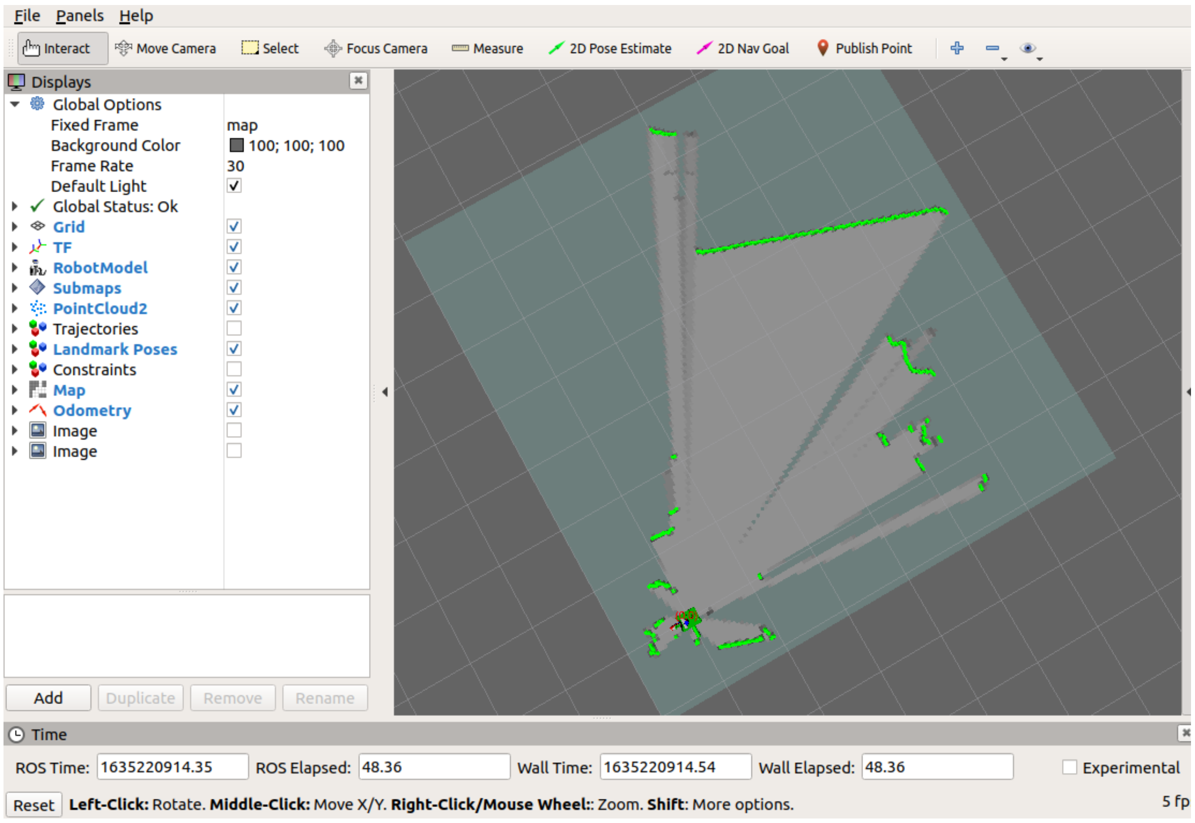


Global Options
Global Status: Ok
Grid ✓
TF ✓
Axes ✓
Image
Image
PointCloud2
Map ✓
RobotModel ✓
Odometry
LaserScan ✓
Status: Ok
Topic /scan
Unreliable
Selectable ✓
Style Points
Size (Pixels) 5
Alpha 1
Decay Time 0
Position Transformer XYZ
Color Transformer Intensity
Queue Size 10
Channel Name intensity
Use rainbow ✓
Invert Rainbow
Min Color ■ 0; 0; 0
Max Color □ 255; 255; 255
Autocompute Inten... ✓
Min Intensity 47

ROS Time: 1635217880.48 ROS Elapsed: 41.65 Wall Time: 1635217880.67 Wall Elapsed: 41.65 Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 5 fps

1.1.4、cartographer



Global Options
Fixed Frame: map
Background Color: 100; 100; 100
Frame Rate: 30
Default Light: ✓
Global Status: Ok
Grid ✓
TF ✓
RobotModel ✓
Submaps ✓
PointCloud2 ✓
Trajectories
Landmark Poses ✓
Constraints
Map ✓
Odometry ✓
Image
Image

ROS Time: 1635220914.35 ROS Elapsed: 48.36 Wall Time: 1635220914.54 Wall Elapsed: 48.36

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

1.2、 Save map

Several mapping algorithms save maps in different ways.

- cartographer: execute the following commands

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

- gmapping,hector,karto： Execute the following command to save.

```
rosrun map_server map_saver -f ~/rplidar_ws/src/transbot_nav/maps/my_map    # Method 1
bash ~/rplidar_ws/src/transbot_nav/maps/map.sh                               # Method 2
```

The map will be saved this path: ~/rplidar_ws/src/transbot_nav/maps/

one pgm picture， one yaml file.

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
```

Parameter analysis:

- image: The path of the map file, which can be an absolute path or a relative path
- resolution: The resolution of the map, m/pixel

- origin: The 2D pose (x,y,yaw) in the lower left corner of the map, where yaw is rotated counterclockwise (yaw=0 means no rotation). Many parts of the current system ignore the yaw value.
- negate: Whether to reverse the meaning of white/black and free/occupied (the interpretation of the threshold is not affected)
- occupied_thresh: Pixels with occupation probability greater than this threshold will be considered as fully occupied.
- free_thresh: Pixels whose occupancy probability is less than this threshold will be considered completely free.

1.3、View related information

View tf tree

```
rosrun rqt_tf_tree rqt_tf_tree
```

View node

```
rqt_graph
```