8、RTAB-Map 3D mapping navigation

Official website of rtabmap： http://introlab.github.io/rtabmap/

rtabmap ros-foxy：https://github.com/introlab/rtabmap/tree/foxy-devel

The operating environment, software and hardware reference configurations are as follows:

- Reference model：ROSMASTER X3
- Robot hardware configuration：Arm series master controller, Silan A1 lidar, AstraPro Plus depth camera
- Robot system：Ubuntu（version no requirement）+ docker（20.10.21 and above）
- PC virtual machine：Ubuntu（20.04）+ ROS2（Foxy）
- Use scene: use on a relatively clean 2D plane

8.1、Introduction

rtabmap is a SLAM method based on RGB-D images, which uses a bag-of-words-based global Bayesian loop closure detector to build maps in real-time in large-scale environments.

The characteristics of rtabmap are as follows：

- It can use a handheld RGBD camera for 6 degrees of freedom RGB-D mapping, or use a robot equipped with lidar for 3 degrees of freedom (2D laser) or 6 degrees of freedom (3D laser) mapping.

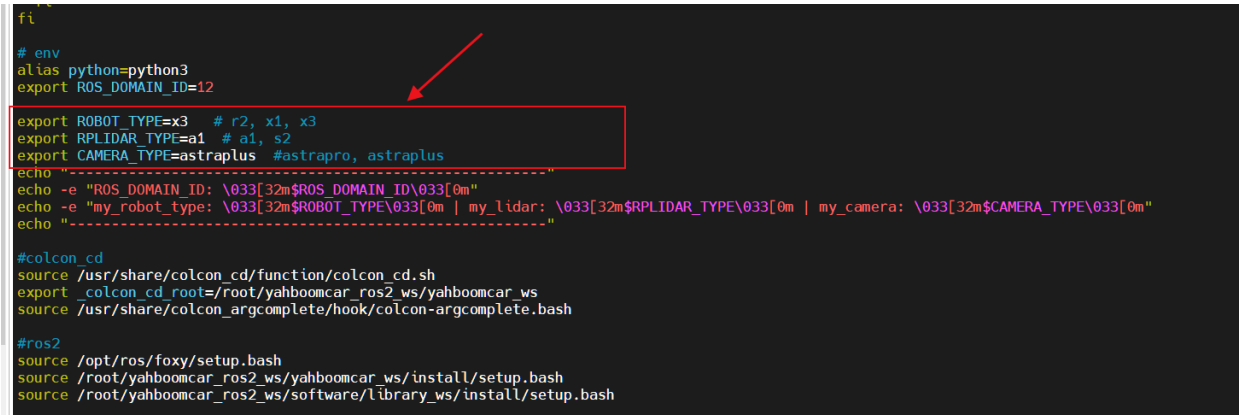
- It uses a memory management mechanism to divide the map into working memory (WM), short-term memory (STM) and long-term memory (LTM) to limit the number of anchor points for closed-loop detection and graph optimization to ensure real-time
- It uses a discrete Bayesian filter to estimate the probability of forming a loop and selects which anchors to transfer or retrieve based on weight and time.
- It uses an incremental dictionary creation method instead of a pre-trained dictionary, which can be adapted to different environments.
- It can run under ROS or as a standalone library and application.

8.2、Configure before use

Note: Since the ROSMASTER series robots are divided into several types of robots and various types of equipment, the factory system has been configured with routines for various types of equipment, but since the product cannot be automatically identified, it is necessary to manually set the machine type and radar model.

After entering the container: make the following modifications according to the car model, radar type and camera type：

```
root@ubuntu:/# cd
root@ubuntu:~# vim .bashrc
```



After the modification is complete, save and exit vim, then execute：

```
root@ubuntu:~# source .bashrc
------------------------------------------------------
ROS_DOMAIN_ID: 12
my_robot_type: x3 | my_lidar: a1 | my_camera: astraplus
------------------------------------------------------
root@ubuntu:~#
```

You can see the current modified car model, radar type and camera type

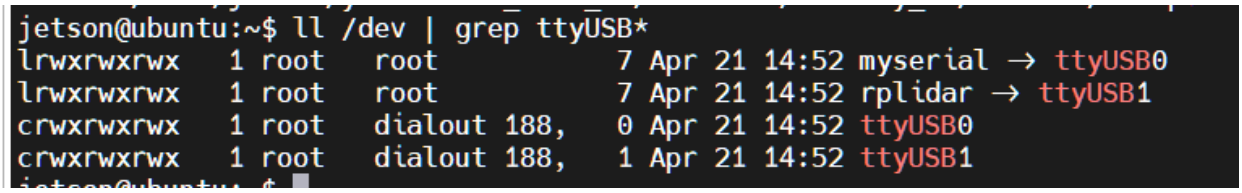
Note: The depth camera of the car should be directly connected to the main controller, not to the USB-HUB, so as to improve the transmission efficiency of the depth camera and improve the effect of map building and navigation. Otherwise, it may result in the inability to build maps and navigate

8.3、Mapping

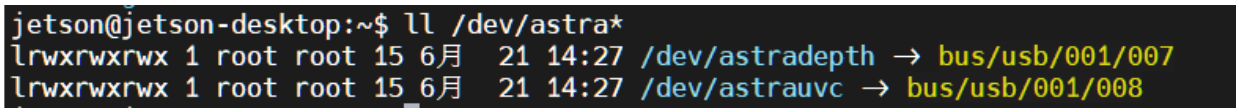
Note: When building a map, the slower the speed, the better the effect (note that if the rotation speed is slower), the effect will be poor if the speed is too fast.

First of all, port binding operation needs to be done on the host machine [that is, the jetson of the car] [see the port binding tutorial chapter], here mainly use radar, serial port and camera three devices.

Then check whether the radar and the serial device are in the port binding state: on the host machine [that is, on the jetson of the car], refer to the following command to execute the check, and the successful binding is as follows：



Check if the camera is in port binding state.



If it is displayed that the radar, serial port, and camera devices are not bound, you can plug and unplug the USB cable to check again.

Enter the docker container, see [in the docker course ----- 5, enter the robot's docker container], and execute the following launch file in the terminal：

1、 Start the Astra camera

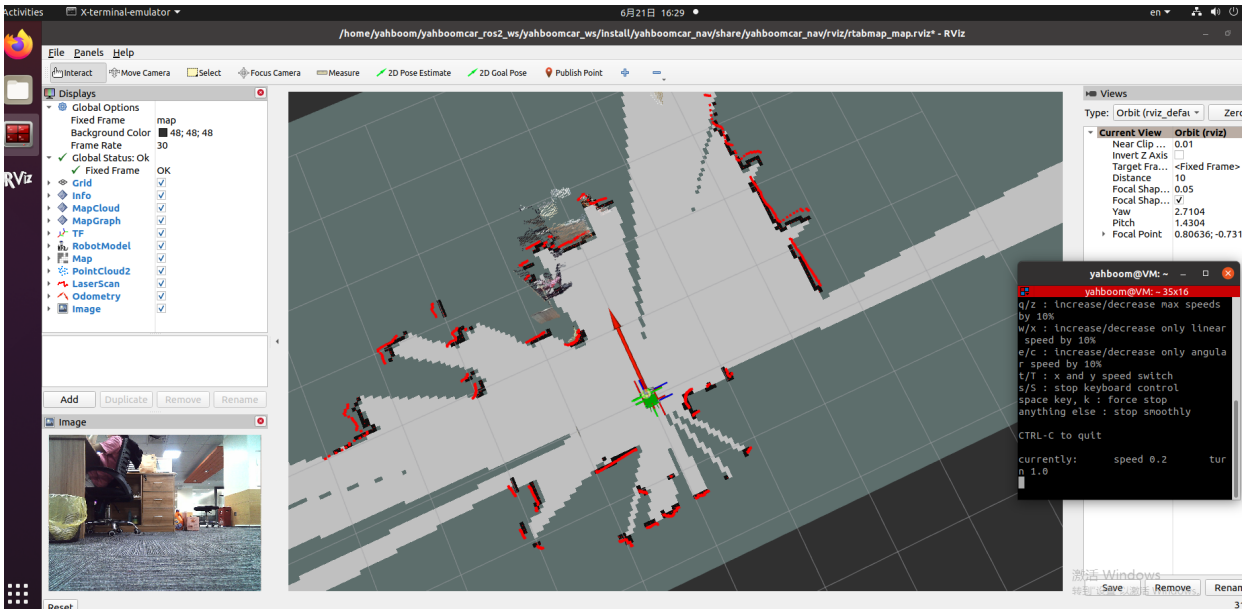
```
ros2 launch astra_camera astro_pro_plus.launch.xml
```

2、Start mapping

```
ros2 launch yahboomcar_nav map_rtabmap_launch.py
```

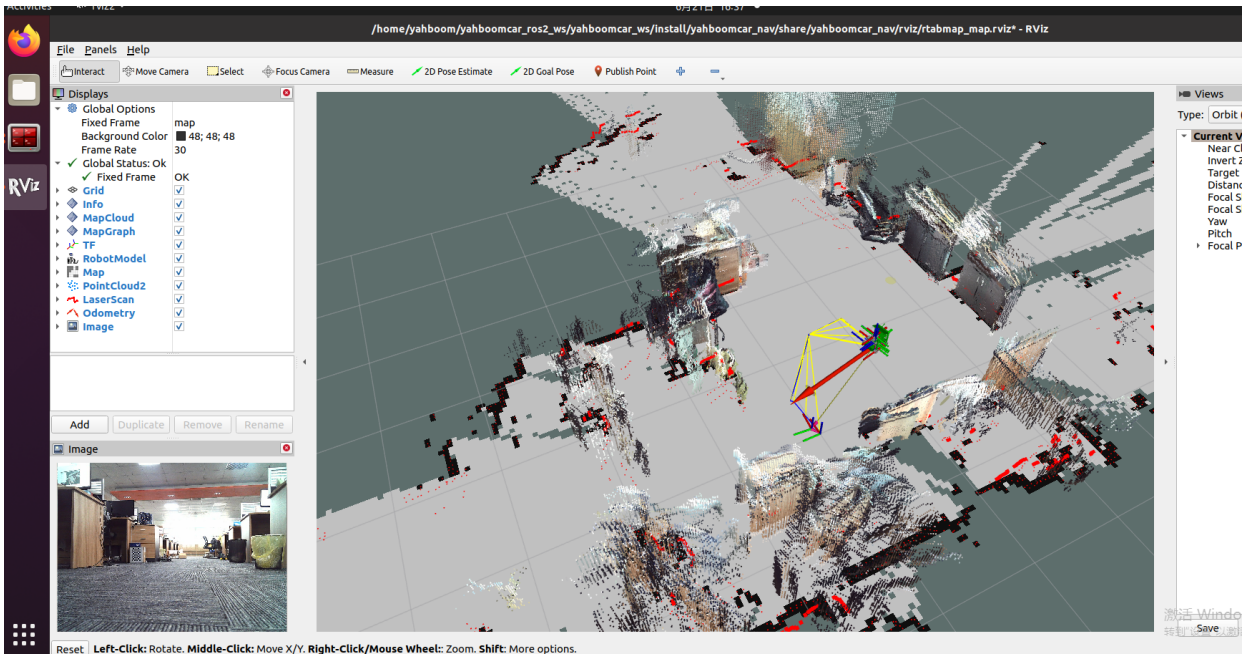
3、Start rviz to display the map. This step is recommended to be performed in a virtual machine. Multi-machine communication needs to be configured in the virtual machine

```
ros2 launch yahboomcar_nav display_rtabmap_map_launch.py
```



4、 Start the keyboard control node. This step is recommended to be performed in a virtual machine. Multi-machine communication needs to be configured in the virtual machine.Or use the remote control [slowly move the car] to start building the map until the complete map is built.

```
ros2 run yahboomcar_ctrl yahboom_keyboard
```



5、Map save

When the map is built, directly [ctrl+c] to exit the map node, and the system will automatically save the map. Map default save path【~/.ros/rtabmap.db】

8.4、Navigation

Enter the docker container (see **[docker course chapter ----- 5, enter the robot's docker container]** for the steps), and execute in a terminal:

8.4.1、Start camera

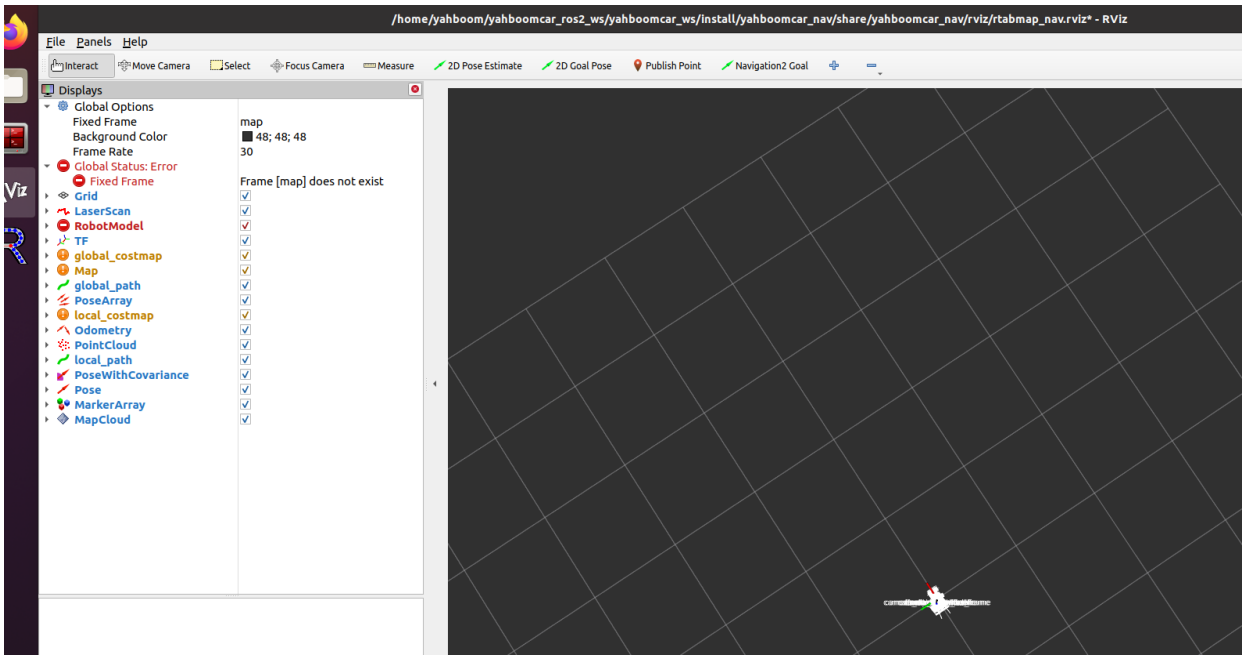
```
ros2 launch astra_camera astro_pro_plus.launch.xml
```

8.4.2、Start rviz to display the map [virtual machine start]

Configure multi-machine communication in the Ubuntu virtual machine and the docker container. This step is recommended to start in the virtual machine: to maintain time synchronization and reduce resource consumption, because if you use vnc, it is very dependent on the network, which may cause navigation failure.

【Note that you must first start the node that displays the map, and then start the navigation node in step 3. This is because the navigation2 terminal map topic is only published once. How to start the navigation node first, and then start the rviz display, you may not be able to subscribe to that The only map topic that was released once, causing the map not to be displayed】

```
ros2 launch yahboomcar_nav display_rtabmap_nav_launch.py
```

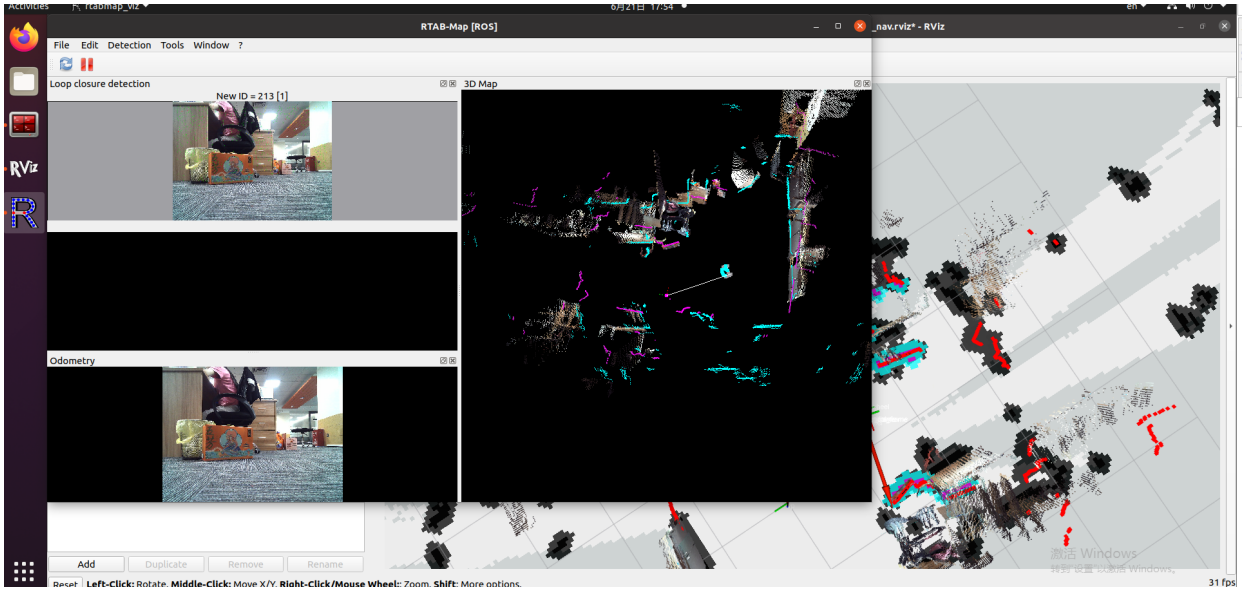


8.4.3、Display rtabmap_viz [Virtual machine start]

rtabmap_viz is the visual interface of RTAB-Map, which is the package of RTAB-Map GUI graphics library, similar to rviz but has options for RTAB-Map. It can subscribe to different topics, such as odom, rgb/image, depth/image, scan, etc., to display the process and results of SLAM, and load the 3D map into rviz at the same time. This step is recommended to start in a virtual machine.

Note that the map will not be displayed after this step is started, and it will not be displayed until the next step of the navigation node is started.

```
ros2 launch yahboomcar_nav rtabmap_viz_launch.py
```



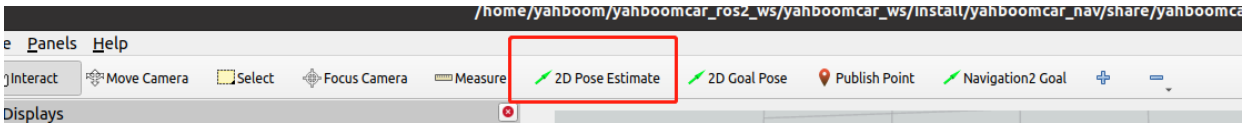
8.4.4、Start navigation node

Navigation can be divided into single-point navigation and multi-point navigation, which will be introduced below

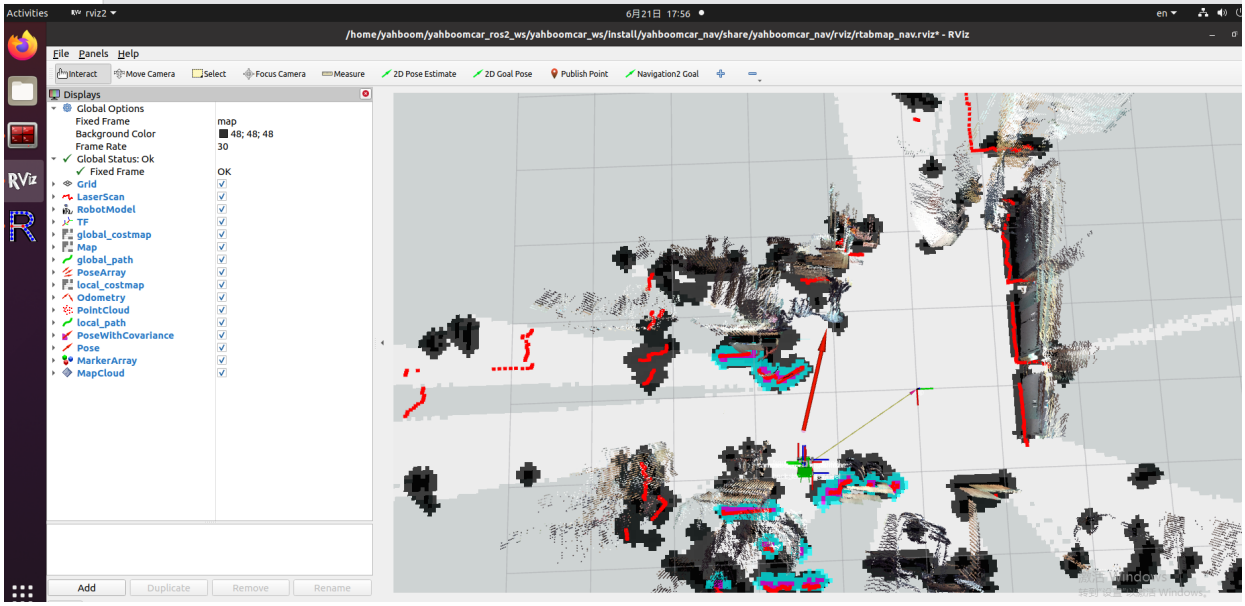
1、Start the navigation node, enter the docker container, and execute in a terminal:

```
ros2 launch yahboomcar_nav navigation_rtabmap_launch.py
```

2、Click [2D Pose Estimate] on rviz, then compare the pose of the car and mark an initial pose for the car on the map.



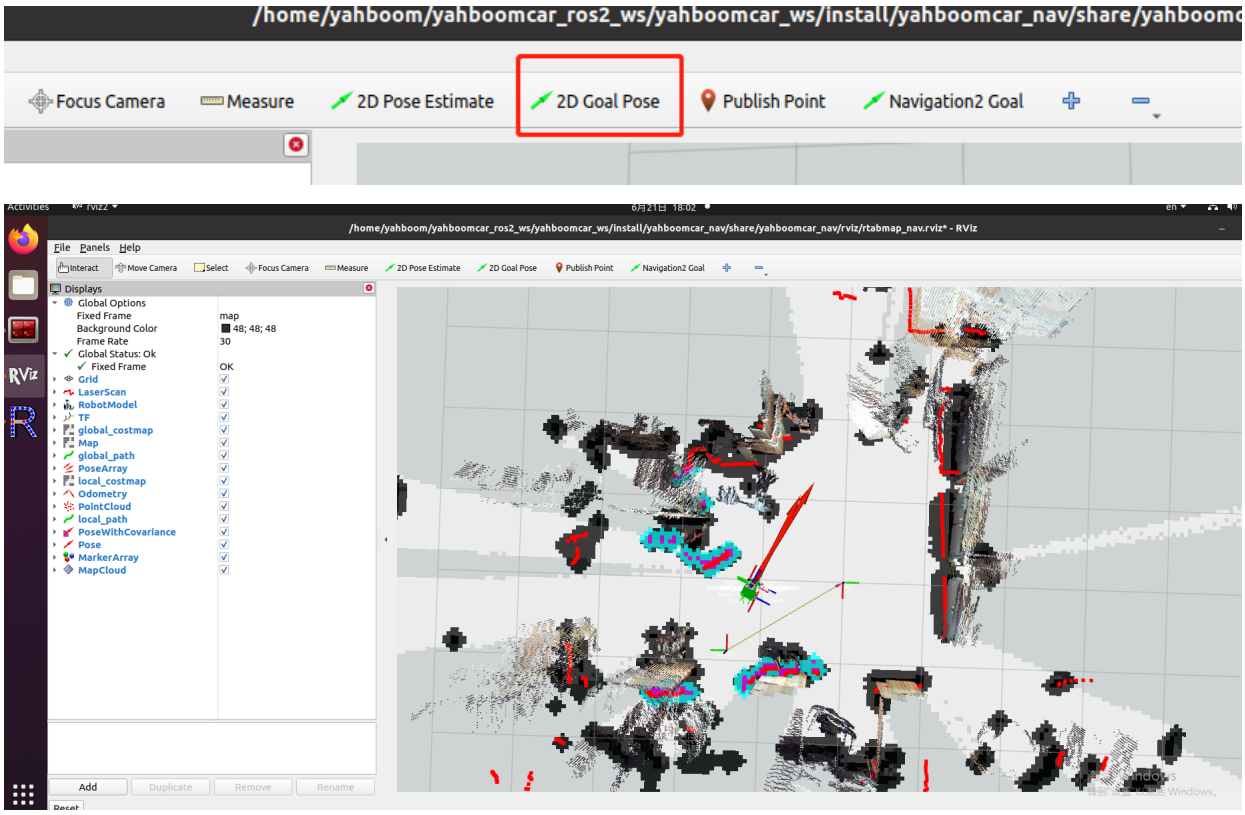
The display after marking is as follows：



3、Comparing the overlap between the radar scanning point and the obstacle, you can set the initial pose of the car several times until the radar scanning point and the obstacle roughly coincide；

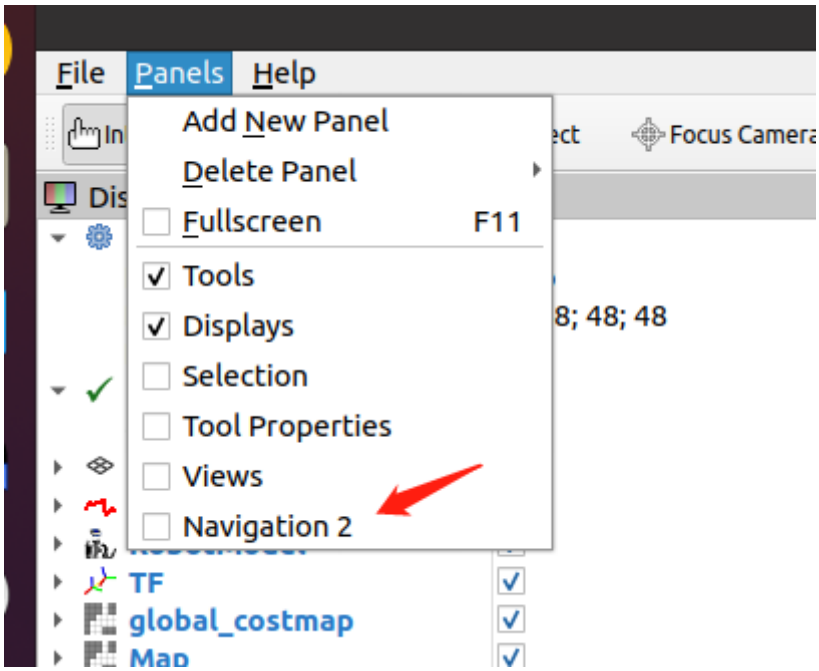
8.4.5、Single point navigation

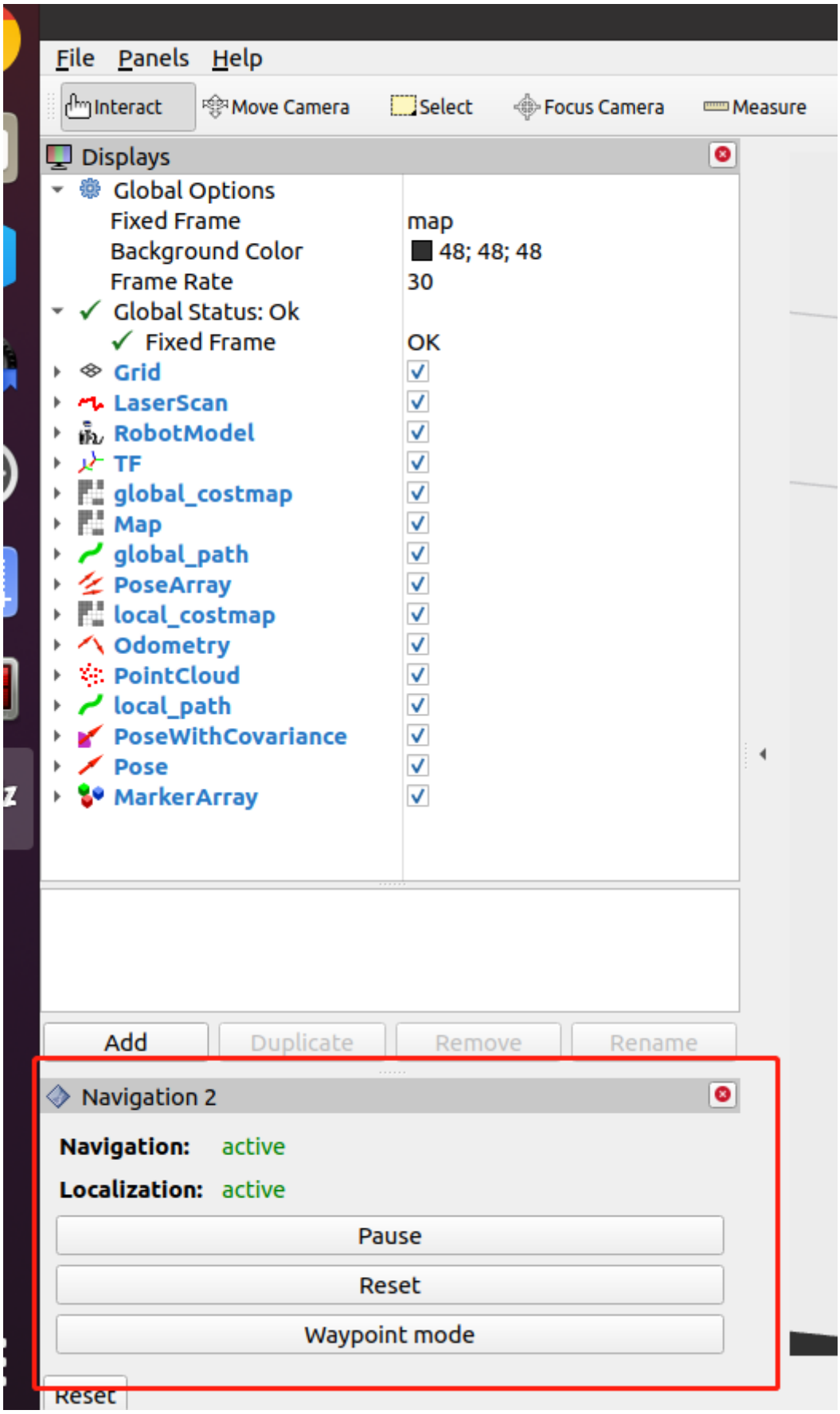
After the initial pose is set, you can click [2D Goal Pose] to set a navigation target point, and the car will start single-point navigation;



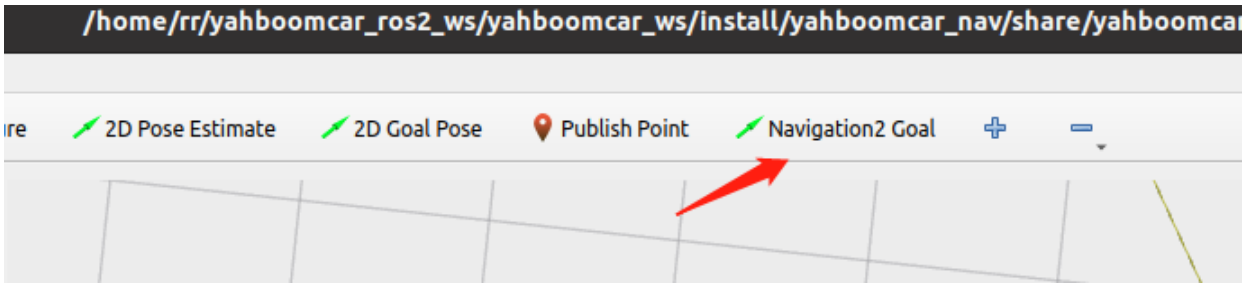
8.2.6、Multipoint navigation

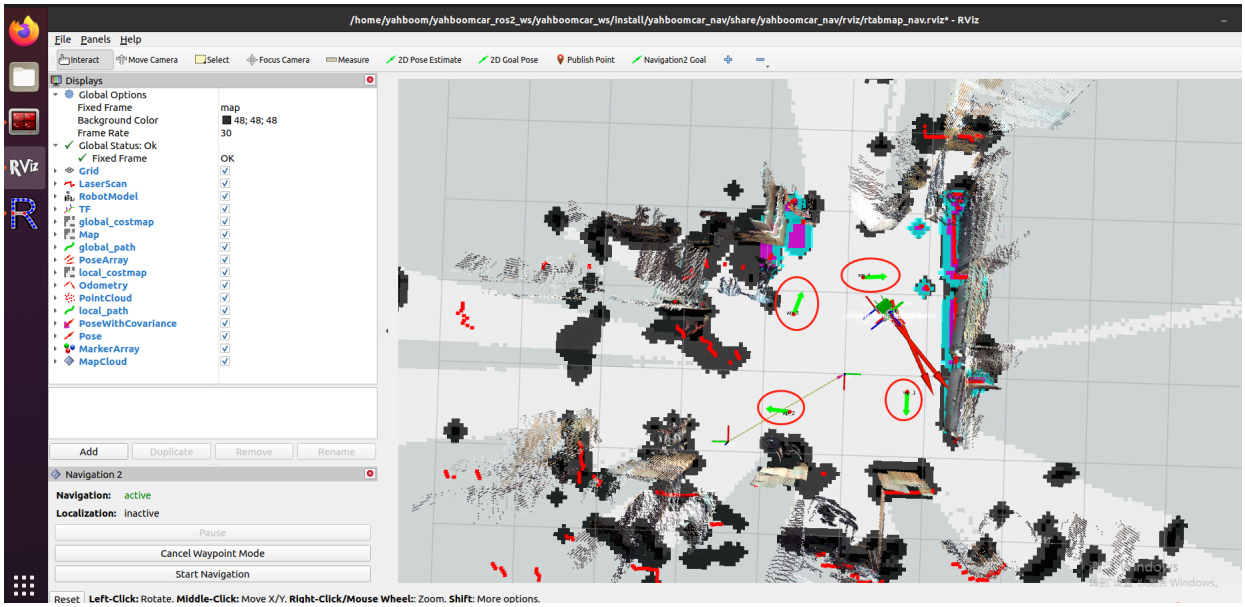
1、After the initial pose is set, you can click [Panels] in the upper left corner of rviz --- select [Navigation 2], and the [Navigation 2] panel will be displayed.



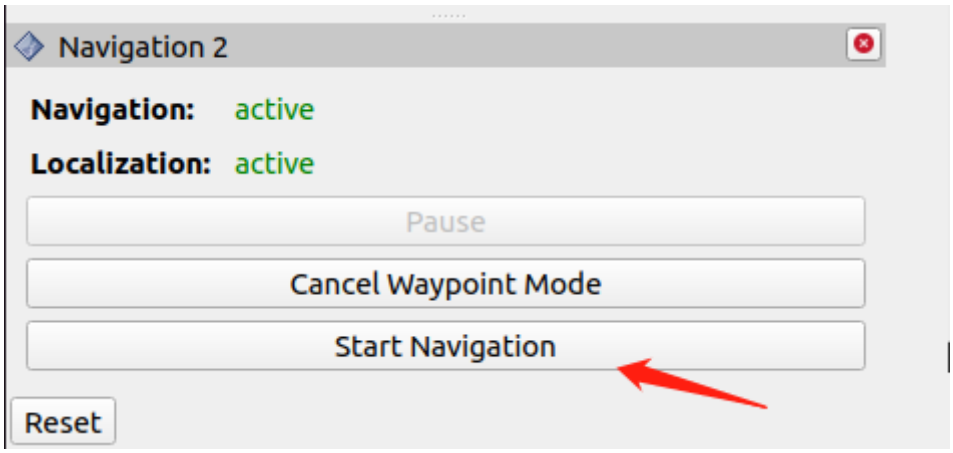

File
Panels
Help
Interact
Move Camera
Select
Focus Camera
Measure
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Global Status: Ok
Fixed Frame
OK
Grid
LaserScan
RobotModel
TF
global_costmap
Map
global_path
PoseArray
local_costmap
Odometry
PointCloud
local_path
PoseWithCovariance
Pose
MarkerArray
Add
Duplicate
Remove
Rename
Navigation 2
Navigation: active
Localization: active
Pause
Reset
Waypoint mode
Reset

2、Click [Waypoint mode] in the above picture, then click [Navigation2 Goal] on rviz, you can mark a target point on the map, click [Navigation2 Goal] again, you can mark the second target point on the map. Multiple target points can be marked at one time.

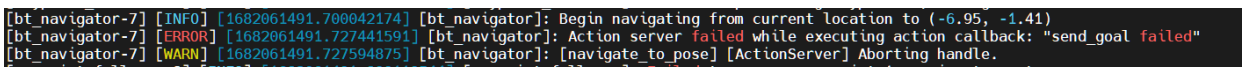




3、After marking multiple target points, click [Start Navigation] to start multi-point navigation. After the multi-point navigation is completed, the car will stay at the pose of the last target point.



4、It may appear during the navigation process. This is due to the navigation2 itself in the ros-foxy version. The subsequent ros2 version has been fixed.

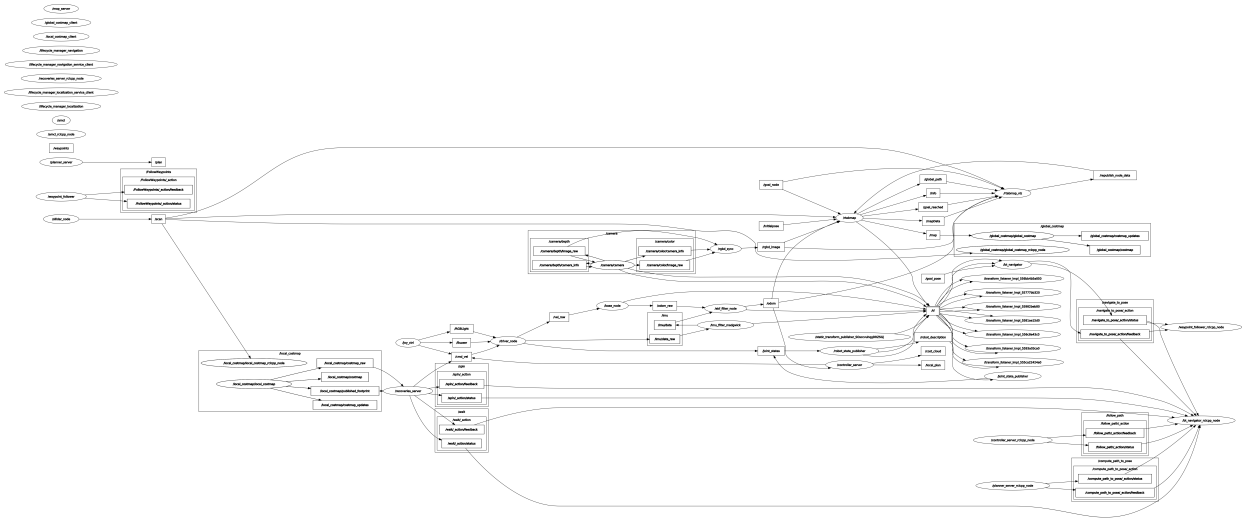


5、During the navigation process, rviz may exit. This may be caused by insufficient resources. You can close the display of rtabmap_viz on the virtual machine after the navigation node starts to fully display the 3D map.

8.5、Node resolution

8.5.1、Show Computational Graph

```
rqt_graph
```



8.5.2、Rtabmap navigation related node details

/rgbd_sync

```
root@jetson-desktop:~# ros2 node info /rgbd_sync
/rgbd_sync
  Subscribers:
    /camera/color/camera_info: sensor_msgs/msg/CameraInfo
    /camera/color/image_raw: sensor_msgs/msg/Image
    /camera/depth/image_raw: sensor_msgs/msg/Image
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rgbd_image: rtabmap_msgs/msg/RGBDImage
    /rgbd_image/compressed: rtabmap_msgs/msg/RGBDImage
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /rgbd_sync/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /rgbd_sync/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /rgbd_sync/get_parameters: rcl_interfaces/srv/GetParameters
    /rgbd_sync/list_parameters: rcl_interfaces/srv/ListParameters
    /rgbd_sync/set_parameters: rcl_interfaces/srv/SetParameters
    /rgbd_sync/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:

  Action Clients:
```

/rtabmap

```
root@jetson-desktop:~# ros2 node info /rtabmap
/rtabmap
  Subscribers:
    /global_pose: geometry_msgs/msg/PoseWithCovarianceStamped
    /goal: geometry_msgs/msg/PoseStamped
    /goal_node: rtabmap_msgs/msg/Goal
    /gps/fix: sensor_msgs/msg/NavSatFix
    /imu: sensor_msgs/msg/Imu
    /initialpose: geometry_msgs/msg/PoseWithCovarianceStamped
    /odom: nav_msgs/msg/Odometry
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /republish_node_data: std_msgs/msg/Int32MultiArray
    /rgbd_image: rtabmap_msgs/msg/RGBDImage
    /scan: sensor_msgs/msg/LaserScan
    /user_data_async: rtabmap_msgs/msg/UserData
  Publishers:
    /cloud_ground: sensor_msgs/msg/PointCloud2
    /cloud_map: sensor_msgs/msg/PointCloud2
    /cloud_obstacles: sensor_msgs/msg/PointCloud2
    /global_path: nav_msgs/msg/Path
    /global_path_nodes: rtabmap_msgs/msg/Path
    /goal_out: geometry_msgs/msg/PoseStamped
    /goal_reached: std_msgs/msg/Bool
    /grid_prob_map: nav_msgs/msg/OccupancyGrid
    /info: rtabmap_msgs/msg/Info
    /labels: visualization_msgs/msg/MarkerArray
    /landmarks: geometry_msgs/msg/PoseArray
    /local_grid_empty: sensor_msgs/msg/PointCloud2
    /local_grid_ground: sensor_msgs/msg/PointCloud2
    /local_grid_obstacle: sensor_msgs/msg/PointCloud2
    /local_path: nav_msgs/msg/Path
    /local_path_nodes: rtabmap_msgs/msg/Path
    /localization_pose: geometry_msgs/msg/PoseWithCovarianceStamped
    /map: nav_msgs/msg/OccupancyGrid
    /mapData: rtabmap_msgs/msg/MapData
    /mapGraph: rtabmap_msgs/msg/MapGraph
    /mapOdomCache: rtabmap_msgs/msg/MapGraph
    /mapPath: nav_msgs/msg/Path
    /octomap_binary: octomap_msgs/msg/Octomap
    /octomap_empty_space: sensor_msgs/msg/PointCloud2
```

```
    /octomap_empty_space: sensor_msgs/msg/PointCloud2
    /octomap_full: octomap_msgs/msg/Octomap
    /octomap_global_frontier_space: sensor_msgs/msg/PointCloud2
    /octomap_grid: nav_msgs/msg/OccupancyGrid
    /octomap_ground: sensor_msgs/msg/PointCloud2
    /octomap_obstacles: sensor_msgs/msg/PointCloud2
    /octomap_occupied_space: sensor_msgs/msg/PointCloud2
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
    /tf: tf2_msgs/msg/TFMessage
  Service Servers:
    /add_link: rtabmap_msgs/srv/AddLink
    /backup: std_srvs/srv/Empty
    /cancel_goal: std_srvs/srv/Empty
    /cleanup_local_grids: rtabmap_msgs/srv/CleanupLocalGrids
    /detect_more_loop_closures: rtabmap_msgs/srv/DetectMoreLoopClosures
    /get_map: nav_msgs/srv/GetMap
    /get_map_data: rtabmap_msgs/srv/GetMap
    /get_map_data2: rtabmap_msgs/srv/GetMap2
    /get_node_data: rtabmap_msgs/srv/GetNodeData
    /get_nodes_in_radius: rtabmap_msgs/srv/GetNodesInRadius
    /get_plan: nav_msgs/srv/GetPlan
    /get_plan_nodes: rtabmap_msgs/srv/GetPlan
    /get_prob_map: nav_msgs/srv/GetMap
    /global_bundle_adjustment: rtabmap_msgs/srv/GlobalBundleAdjustment
    /list_labels: rtabmap_msgs/srv/ListLabels
    /load_database: rtabmap_msgs/srv/LoadDatabase
    /log_debug: std_srvs/srv/Empty
    /log_error: std_srvs/srv/Empty
    /log_info: std_srvs/srv/Empty
    /log_warning: std_srvs/srv/Empty
    /octomap_binary: octomap_msgs/srv/GetOctomap
    /octomap_full: octomap_msgs/srv/GetOctomap
    /pause: std_srvs/srv/Empty
    /publish_map: rtabmap_msgs/srv/PublishMap
    /remove_label: rtabmap_msgs/srv/RemoveLabel
    /reset: std_srvs/srv/Empty
    /resume: std_srvs/srv/Empty
    /rtabmap/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /rtabmap/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
```

```
    /rtabmap/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /rtabmap/get_parameters: rcl_interfaces/srv/GetParameters
    /rtabmap/list_parameters: rcl_interfaces/srv/ListParameters
    /rtabmap/set_parameters: rcl_interfaces/srv/SetParameters
    /rtabmap/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
    /set_goal: rtabmap_msgs/srv/SetGoal
    /set_label: rtabmap_msgs/srv/SetLabel
    /set_mode_localization: std_srvs/srv/Empty
    /set_mode_mapping: std_srvs/srv/Empty
    /trigger_new_map: std_srvs/srv/Empty
    /update_parameters: std_srvs/srv/Empty
  Service Clients:
    /rtabmap/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /rtabmap/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /rtabmap/get_parameters: rcl_interfaces/srv/GetParameters
    /rtabmap/list_parameters: rcl_interfaces/srv/ListParameters
    /rtabmap/set_parameters: rcl_interfaces/srv/SetParameters
    /rtabmap/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Action Servers:

  Action Clients:
```

/rtabmap_viz

```
root@jetson-desktop:~# ros2 node info /rtabmap_viz
/rtabmap_viz
  Subscribers:
    /global_path: nav_msgs/msg/Path
    /goal_node: rtabmap_msgs/msg/Goal
    /goal_reached: std_msgs/msg/Bool
    /info: rtabmap_msgs/msg/Info
    /mapData: rtabmap_msgs/msg/MapData
    /odom: nav_msgs/msg/Odometry
    /rgbd_image: rtabmap_msgs/msg/RGBDImage
    /scan: sensor_msgs/msg/LaserScan
  Publishers:
    /republish_node_data: std_msgs/msg/Int32MultiArray
  Service Servers:

  Service Clients:

  Action Servers:

  Action Clients:
```

/bt_navigator

```
root@ubuntu:/# ros2 node info /bt_navigator
/bt_navigator
  Subscribers:
    /goal_pose: geometry_msgs/msg/PoseStamped
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /tf: tf2_msgs/msg/TFMessage
    /tf_static: tf2_msgs/msg/TFMessage
  Publishers:
    /bt_navigator/transition_event: lifecycle_msgs/msg/TransitionEvent
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /bt_navigator/change_state: lifecycle_msgs/srv/ChangeState
    /bt_navigator/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /bt_navigator/get_available_states: lifecycle_msgs/srv/GetAvailableStates
    /bt_navigator/get_available_transitions: lifecycle_msgs/srv/GetAvailableTransitions
    /bt_navigator/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /bt_navigator/get_parameters: rcl_interfaces/srv/GetParameters
    /bt_navigator/get_state: lifecycle_msgs/srv/GetState
    /bt_navigator/get_transition_graph: lifecycle_msgs/srv/GetAvailableTransitions
    /bt_navigator/list_parameters: rcl_interfaces/srv/ListParameters
    /bt_navigator/set_parameters: rcl_interfaces/srv/SetParameters
    /bt_navigator/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:
    /navigate_to_pose: nav2_msgs/action/NavigateToPose
  Action Clients:
```

/controller_server

```
root@ubuntu:/# ros2 node info /controller_server
/controller_server
  Subscribers:
    /odom: nav_msgs/msg/Odometry
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /cmd_vel: geometry_msgs/msg/Twist
    /controller_server/transition_event: lifecycle_msgs/msg/TransitionEvent
    /cost_cloud: sensor_msgs/msg/PointCloud
    /evaluation: dwb_msgs/msg/LocalPlanEvaluation
    /local_plan: nav_msgs/msg/Path
    /marker: visualization_msgs/msg/MarkerArray
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /received_global_plan: nav_msgs/msg/Path
    /rosout: rcl_interfaces/msg/Log
    /transformed_global_plan: nav_msgs/msg/Path
  Service Servers:
    /controller_server/change_state: lifecycle_msgs/srv/ChangeState
    /controller_server/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /controller_server/get_available_states: lifecycle_msgs/srv/GetAvailableStates
    /controller_server/get_available_transitions: lifecycle_msgs/srv/GetAvailableTransitions
    /controller_server/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /controller_server/get_parameters: rcl_interfaces/srv/GetParameters
    /controller_server/get_state: lifecycle_msgs/srv/GetState
    /controller_server/get_transition_graph: lifecycle_msgs/srv/GetAvailableTransitions
    /controller_server/list_parameters: rcl_interfaces/srv/ListParameters
    /controller_server/set_parameters: rcl_interfaces/srv/SetParameters
    /controller_server/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:
    /controller_server/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /controller_server/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /controller_server/get_parameters: rcl_interfaces/srv/GetParameters
    /controller_server/list_parameters: rcl_interfaces/srv/ListParameters
    /controller_server/set_parameters: rcl_interfaces/srv/SetParameters
    /controller_server/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Action Servers:

  Action Clients:
```

/global_costmap/global_costmap

```
root@ubuntu:/# ros2 node info /global_costmap/global_costmap
/global_costmap/global_costmap
  Subscribers:
    /global_costmap/footprint: geometry_msgs/msg/Polygon
    /map: nav_msgs/msg/OccupancyGrid
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /global_costmap/costmap: nav_msgs/msg/OccupancyGrid
    /global_costmap/costmap_raw: nav2_msgs/msg/Costmap
    /global_costmap/costmap_updates: map_msgs/msg/OccupancyGridUpdate
    /global_costmap/global_costmap/transition_event: lifecycle_msgs/msg/TransitionEvent
    /global_costmap/published_footprint: geometry_msgs/msg/PolygonStamped
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /global_costmap/clear_around_global_costmap: nav2_msgs/srv/ClearCostmapAroundRobot
    /global_costmap/clear_entirely_global_costmap: nav2_msgs/srv/ClearEntireCostmap
    /global_costmap/clear_except_global_costmap: nav2_msgs/srv/ClearCostmapExceptRegion
    /global_costmap/get_costmap: nav2_msgs/srv/GetCostmap
    /global_costmap/global_costmap/change_state: lifecycle_msgs/srv/ChangeState
    /global_costmap/global_costmap/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /global_costmap/global_costmap/get_available_states: lifecycle_msgs/srv/GetAvailableStates
    /global_costmap/global_costmap/get_available_transitions: lifecycle_msgs/srv/GetAvailableTransitions
    /global_costmap/global_costmap/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /global_costmap/global_costmap/get_parameters: rcl_interfaces/srv/GetParameters
    /global_costmap/global_costmap/get_state: lifecycle_msgs/srv/GetState
    /global_costmap/global_costmap/get_transition_graph: lifecycle_msgs/srv/GetAvailableTransitions
    /global_costmap/global_costmap/list_parameters: rcl_interfaces/srv/ListParameters
    /global_costmap/global_costmap/set_parameters: rcl_interfaces/srv/SetParameters
    /global_costmap/global_costmap/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:

  Action Clients:
```

/local_costmap/local_costmap

```
root@ubuntu:/# ros2 node info /local_costmap/local_costmap
/local_costmap/local_costmap
  Subscribers:
    /local_costmap/footprint: geometry_msgs/msg/Polygon
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /local_costmap/costmap: nav_msgs/msg/OccupancyGrid
    /local_costmap/costmap_raw: nav2_msgs/msg/Costmap
    /local_costmap/costmap_updates: map_msgs/msg/OccupancyGridUpdate
    /local_costmap/local_costmap/transition_event: lifecycle_msgs/msg/TransitionEvent
    /local_costmap/published_footprint: geometry_msgs/msg/PolygonStamped
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /local_costmap/clear_around_local_costmap: nav2_msgs/srv/ClearCostmapAroundRobot
    /local_costmap/clear_entirely_local_costmap: nav2_msgs/srv/ClearEntireCostmap
    /local_costmap/clear_except_local_costmap: nav2_msgs/srv/ClearCostmapExceptRegion
    /local_costmap/get_costmap: nav2_msgs/srv/GetCostmap
    /local_costmap/local_costmap/change_state: lifecycle_msgs/srv/ChangeState
    /local_costmap/local_costmap/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /local_costmap/local_costmap/get_available_states: lifecycle_msgs/srv/GetAvailableStates
    /local_costmap/local_costmap/get_available_transitions: lifecycle_msgs/srv/GetAvailableTransitions
    /local_costmap/local_costmap/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /local_costmap/local_costmap/get_parameters: rcl_interfaces/srv/GetParameters
    /local_costmap/local_costmap/get_state: lifecycle_msgs/srv/GetState
    /local_costmap/local_costmap/get_transition_graph: lifecycle_msgs/srv/GetAvailableTransitions
    /local_costmap/local_costmap/list_parameters: rcl_interfaces/srv/ListParameters
    /local_costmap/local_costmap/set_parameters: rcl_interfaces/srv/SetParameters
    /local_costmap/local_costmap/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:

  Action Clients:
```

/planner_server

```
root@ubuntu:/# ros2 node info /planner_server
/planner_server
  Subscribers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /plan: nav_msgs/msg/Path
    /planner_server/transition_event: lifecycle_msgs/msg/TransitionEvent
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /planner_server/change_state: lifecycle_msgs/srv/ChangeState
    /planner_server/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /planner_server/get_available_states: lifecycle_msgs/srv/GetAvailableStates
    /planner_server/get_available_transitions: lifecycle_msgs/srv/GetAvailableTransitions
    /planner_server/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /planner_server/get_parameters: rcl_interfaces/srv/GetParameters
    /planner_server/get_state: lifecycle_msgs/srv/GetState
    /planner_server/get_transition_graph: lifecycle_msgs/srv/GetAvailableTransitions
    /planner_server/list_parameters: rcl_interfaces/srv/ListParameters
    /planner_server/set_parameters: rcl_interfaces/srv/SetParameters
    /planner_server/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:
    /planner_server/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /planner_server/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /planner_server/get_parameters: rcl_interfaces/srv/GetParameters
    /planner_server/list_parameters: rcl_interfaces/srv/ListParameters
    /planner_server/set_parameters: rcl_interfaces/srv/SetParameters
    /planner_server/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Action Servers:

  Action Clients:
```

/recoveries_server

```
root@ubuntu:/# ros2 node info /recoveries_server
/recoveries_server
  Subscribers:
    /local_costmap/costmap_raw: nav2_msgs/msg/Costmap
    /local_costmap/published_footprint: geometry_msgs/msg/PolygonStamped
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /cmd_vel: geometry_msgs/msg/Twist
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /recoveries_server/transition_event: lifecycle_msgs/msg/TransitionEvent
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /recoveries_server/change_state: lifecycle_msgs/srv/ChangeState
    /recoveries_server/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /recoveries_server/get_available_states: lifecycle_msgs/srv/GetAvailableStates
    /recoveries_server/get_available_transitions: lifecycle_msgs/srv/GetAvailableTransitions
    /recoveries_server/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /recoveries_server/get_parameters: rcl_interfaces/srv/GetParameters
    /recoveries_server/get_state: lifecycle_msgs/srv/GetState
    /recoveries_server/get_transition_graph: lifecycle_msgs/srv/GetAvailableTransitions
    /recoveries_server/list_parameters: rcl_interfaces/srv/ListParameters
    /recoveries_server/set_parameters: rcl_interfaces/srv/SetParameters
    /recoveries_server/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:
    /backup: nav2_msgs/action/BackUp
    /spin: nav2_msgs/action/Spin
    /wait: nav2_msgs/action/Wait
  Action Clients:
```

/waypoint_follower

```
root@ubuntu:/# ros2 node info /waypoint_follower
/waypoint_follower
  Subscribers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
    /waypoint_follower/transition_event: lifecycle_msgs/msg/TransitionEvent
  Service Servers:
    /waypoint_follower/change_state: lifecycle_msgs/srv/ChangeState
    /waypoint_follower/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /waypoint_follower/get_available_states: lifecycle_msgs/srv/GetAvailableStates
    /waypoint_follower/get_available_transitions: lifecycle_msgs/srv/GetAvailableTransitions
    /waypoint_follower/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /waypoint_follower/get_parameters: rcl_interfaces/srv/GetParameters
    /waypoint_follower/get_state: lifecycle_msgs/srv/GetState
    /waypoint_follower/get_transition_graph: lifecycle_msgs/srv/GetAvailableTransitions
    /waypoint_follower/list_parameters: rcl_interfaces/srv/ListParameters
    /waypoint_follower/set_parameters: rcl_interfaces/srv/SetParameters
    /waypoint_follower/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:
    /FollowWaypoints: nav2_msgs/action/FollowWaypoints
  Action Clients:
```

8.5.3、TF transform

```
ros2 run tf2_tools view_frames.py
```