6. gmapping graph building algorithm

6.1 Program function description

After the programs on the virtual machine and the car are started, the car is controlled by the handle or keyboard. The car uses the lidar scanning data during the moving process, uses gmapping algorithm to build a map, and saves the map after construction. This process is visualized in rviz.

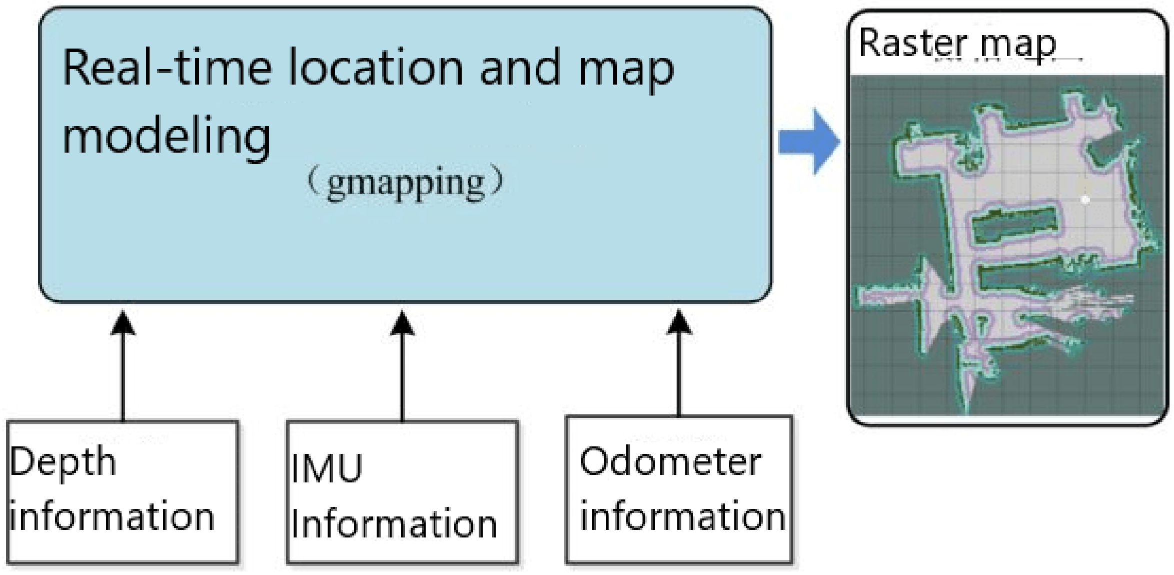
6.2. Introduction to Gmapping

> wiki: http://wiki.ros.org/gmapping/
>
> ros2 gmapping: https://github.com/Project-MANAS/slam_gmapping

- Gmapping is only applicable to the point where the number of 2D laser dots in a single frame is less than 1440. If the number of 2D laser dots in a single frame is greater than 1440, the [[mapping-4] process has died] problem will occur.
- Gmapping is based on the RBpf particle filter algorithm. The real-time positioning process is separated from the mapping process.
  Gmapping makes two major improvements to the RBpf algorithm: improved proposal distribution and selective resampling.

Advantages: Gmapping can build indoor maps in real time, and the calculation required for building small-scene maps is small and the precision is high.

Disadvantages: The number of particles required increases as the scene grows, because each particle carries a map, so the memory and computation required to build a large map increase. Therefore, it is not suitable for building large scene maps. And there is no loop detection, so the map may be misaligned when the loop is closed, although increasing the number of particles can make the map closed, but at the cost of increasing the amount of computation and memory.



6.3 Program code reference path

After SSH connection car, the location of the function source code is located at,

```
/userdata/yahboomcar_ws/src/yahboomcar_nav/launch/map_gmapping_launch.py
```

The virtual machine side source code is located at,

```
/home/yahboom/dev_ws/src/yahboomcar_rviz/launch/yahboomcar_mapping_launch.py
```

6.4. Program Startup

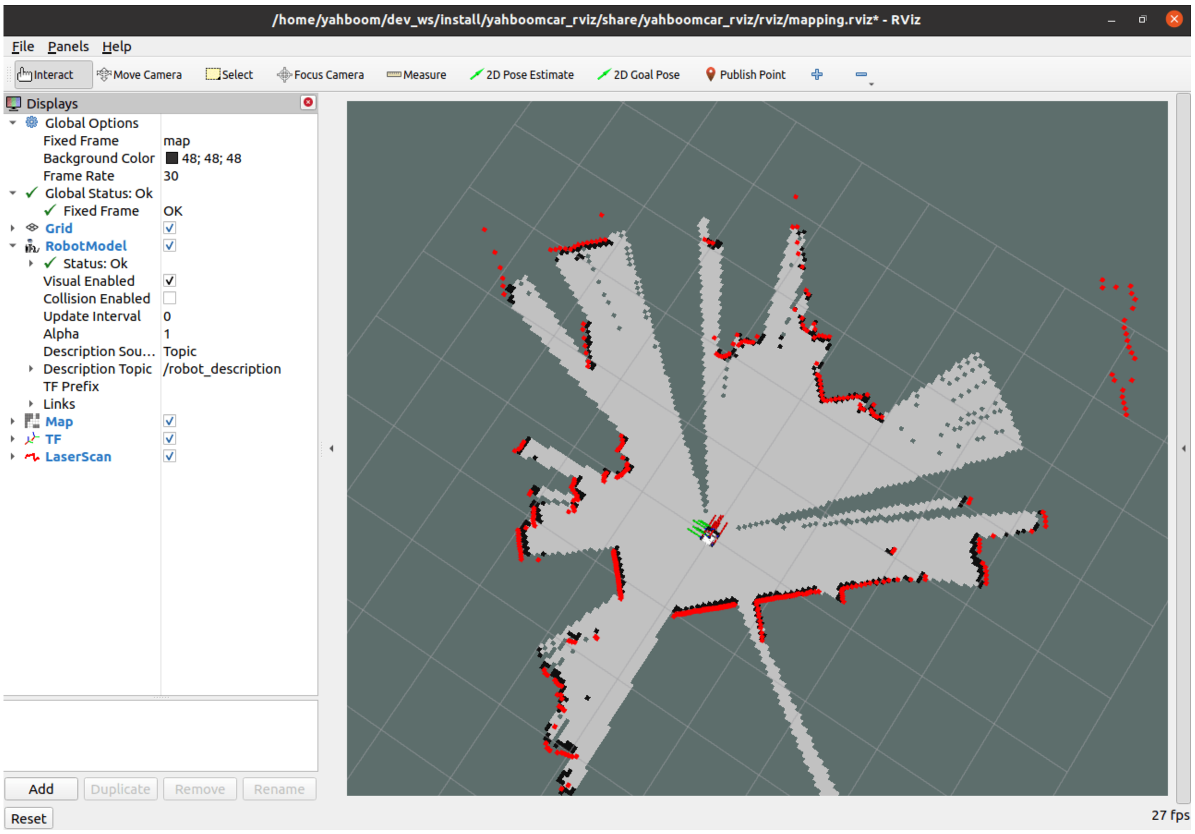
6.4.1. Build the diagram

Open the virtual machine terminal and type,

```
ros2 launch yahboomcar_rviz yahboomcar_mapping_launch.py
```

After SSH connects to the car, terminal input,

```
ros2 launch yahboomcar_nav map_gmapping_launch.py
```



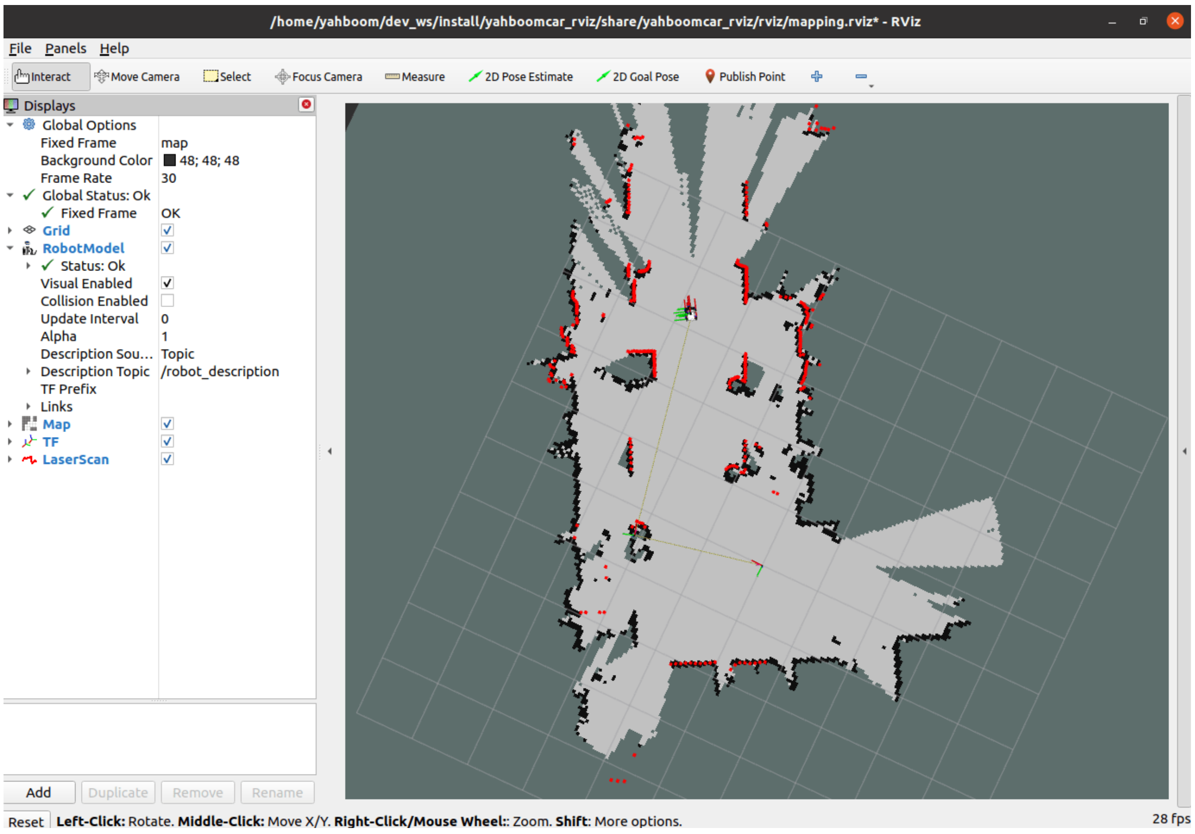
/home/yahboom/dev_ws/install/yahboomcar_rviz/share/yahboomcar_rviz/rviz/mapping.rviz* - RViz

Global Options	
Fixed Frame	map
Background Color	48; 48; 48
Frame Rate	30
Global Status: Ok	
Fixed Frame	OK
Grid	✓
RobotModel	✓
Status: Ok	
Visual Enabled	✓
Collision Enabled	
Update Interval	0
Alpha	1
Description Sou…	Topic
Description Topic	/robot_description
TF Prefix	
Links	
Map	✓
TF	✓
LaserScan	✓

Handle [L1] key unlock, control car movement. If you do not use the handle, you can use the keyboard control, the car terminal input,

```
ros2 run yahboomcar_ctrl yahboom_keyboard
```

【 Slowly move the cart 】 Start to build the map until the complete map is built.



/home/yahboom/dev_ws/install/yahboomcar_rviz/share/yahboomcar_rviz/rviz/mapping.rviz* - RViz

Global Options	
Fixed Frame	map
Background Color	48; 48; 48
Frame Rate	30
Global Status: Ok	
Fixed Frame	OK
Grid	✓
RobotModel	✓
Status: Ok	
Visual Enabled	✓
Collision Enabled	
Update Interval	0
Alpha	1
Description Sou…	Topic
Description Topic	/robot_description
TF Prefix	
Links	
Map	✓
TF	✓
LaserScan	✓

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

Note: When building the diagram, the slower the better the effect (especially the rotation speed should be slower), the faster the speed, the effect will be poor.

In addition, the paths of gmapping parameters are,

```
/userdata/software/slam_ws/src/slam_gmapping/params/slam_gmapping.yaml
```

You can modify it according to your needs, compile and run it.

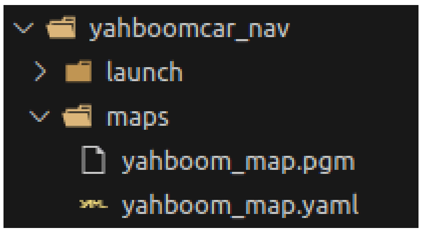
6.4.2. Map saving

Car terminal input,

```
ros2 launch yahboomcar_nav save_map_launch.py
```

Map saving path is as follows:,

```
/userdata/yahboomcar_ws/src/yahboomcar_nav/maps
```



```
yahboomcar_nav
  launch
  maps
    yahboom_map.pgm
    yahboom_map.yaml
```

Includes a.pGM image and a.YAML file. The.yaml file contains the following contents:

```
image: /userdata/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.pgm
mode: trinary
resolution: 0.05
origin: [-5.95, -3.26, 0]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.25
```

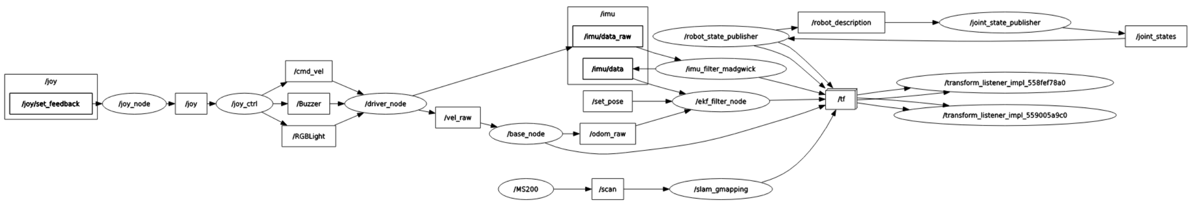
Parameter analysis:

- image: indicates the path of the map file. The path can be absolute or relative
- mode: This property can be one of trinary, scale, or raw, depending on the mode selected. trinary is the default mode
- resolution: Map resolution in meters per pixel
- origin: 2D pose (x,y,yaw) in the lower left corner of the map, where yaw is rotated counterclockwise (yaw=0 indicates no rotation). yaw values are currently ignored in many parts of the system.
- negate: Whether to reverse the meanings of white/black, free/occupied (the interpretation of the threshold is not affected)
- occupied_thresh: The occupied_thresh pixel whose occupied_probability is greater than this threshold will be considered as fully occupied.
- free_thresh: Pixels whose occupancy probability is less than this threshold are considered completely free.

6.4.3. View the topic communication node graph

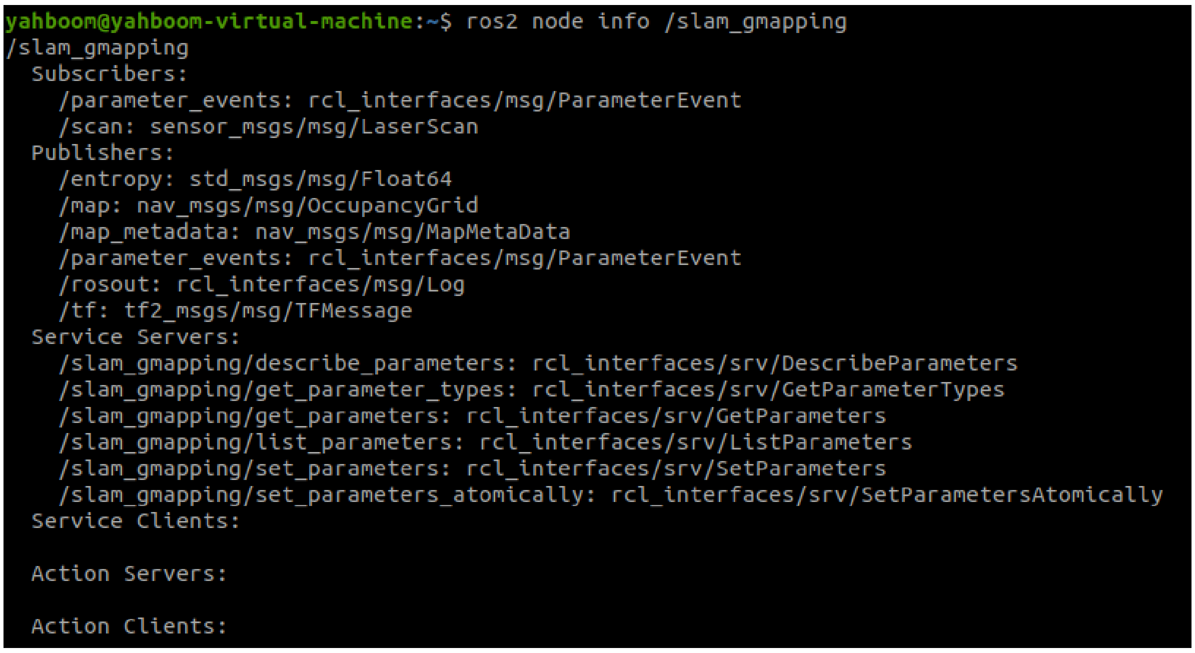
Virtual machine terminal input,

```
ros2 run rqt_graph rqt_graph
```



To view gmapping node details, enter,

```
ros2 node info /slam_gmapping
```

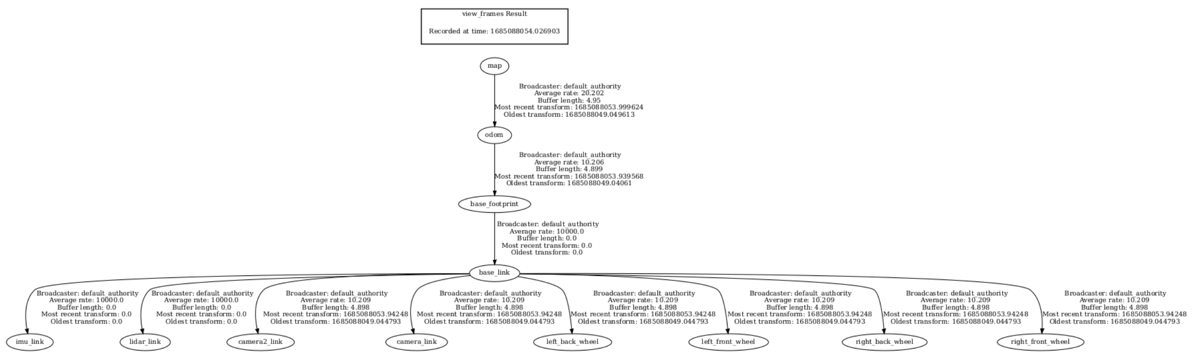


```
yahboom@yahboom-virtual-machine:~$ ros2 node info /slam_gmapping
/slam_gmapping
  Subscribers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /scan: sensor_msgs/msg/LaserScan
  Publishers:
    /entropy: std_msgs/msg/Float64
    /map: nav_msgs/msg/OccupancyGrid
    /map_metadata: nav_msgs/msg/MapMetaData
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
    /tf: tf2_msgs/msg/TFMessage
  Service Servers:
    /slam_gmapping/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /slam_gmapping/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /slam_gmapping/get_parameters: rcl_interfaces/srv/GetParameters
    /slam_gmapping/list_parameters: rcl_interfaces/srv/ListParameters
    /slam_gmapping/set_parameters: rcl_interfaces/srv/SetParameters
    /slam_gmapping/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:

  Action Clients:
```

6.4.4 View the TF tree

Open the virtual machine terminal and type,

```
#保存tf树# Save tf tree
ros2 run tf2_tools view_frames.py
#查看tf树# View tf tree
evince frames.pdf
```


view_frames Result
Recorded at time: 1685088054.026903
map
Broadcaster: default_authority
Average rate: 20.202
Buffer length: 4.95
Most recent transform: 1685088053.999624
Oldest transform: 1685088049.049613
odom
Broadcaster: default_authority
Average rate: 10.206
Buffer length: 4.899
Most recent transform: 1685088053.939568
Oldest transform: 1685088049.04061
base_footprint
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base_link
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
imu_link
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
lidar_link
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.898
Most recent transform: 1685088053.94248
Oldest transform: 1685088049.044793
camera2_link
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.898
Most recent transform: 1685088053.94248
Oldest transform: 1685088049.044793
camera_link
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.898
Most recent transform: 1685088053.94248
Oldest transform: 1685088049.044793
left_back_wheel
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.898
Most recent transform: 1685088053.94248
Oldest transform: 1685088049.044793
left_front_wheel
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.898
Most recent transform: 1685088053.94248
Oldest transform: 1685088049.044793
right_back_wheel
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.898
Most recent transform: 1685088053.94248
Oldest transform: 1685088049.044793
right_front_wheel