7. ORB_SLAM2 Octomap Mapping

Note: This example is implemented on Orin NX, and cannot be implemented on a virtual machine. It only explains how the process is implemented. If you want to implement this function on your own motherboard, you need to compile the entire function package and connect related peripherals and debug.

octomap official website: http://octomap.github.io/

octomap source code: https://github.com/OctoMap/octomap

octomap wiki: http://wiki.ros.org/octomap

octomap_server: http://wiki.ros.org/octomap_server

The operating environment and reference configuration of software and hardware are as follows:

- Reference vehicle: ROSMASTER X3
- Robot hardware configuration: Arm series master control, Silan A1 laser radar, depth camera
- Robot system: Ubuntu 20.04
- PC virtual machine: Ubuntu (20.04) + ROS2 (Foxy)
- Usage scenario: Use on a relatively clean 2D plane

7.1, Introduction

Octomap is an octree-based 3D map creation tool that can display complete 3D graphics including barrier-free areas and unmapped areas, and sensor data based on occupancy grids can be fused and updated in multiple measurements; Maps are available in multiple resolutions, data can be compressed, and storage is compact. In fact, the code of octomap mainly consists of two modules: octomap, a 3D map creation tool, and octovis, a visualization tool.
Compared with point clouds, it can save a lot of space. The map created by octomap looks like this: (different resolutions from left to right)

7.2 , octomap map based on camera

Execute the following launch file in the terminal:

1. Start the camera

```
#Camera startup
ros2 launch astra_camera gemini.launch.xml
```

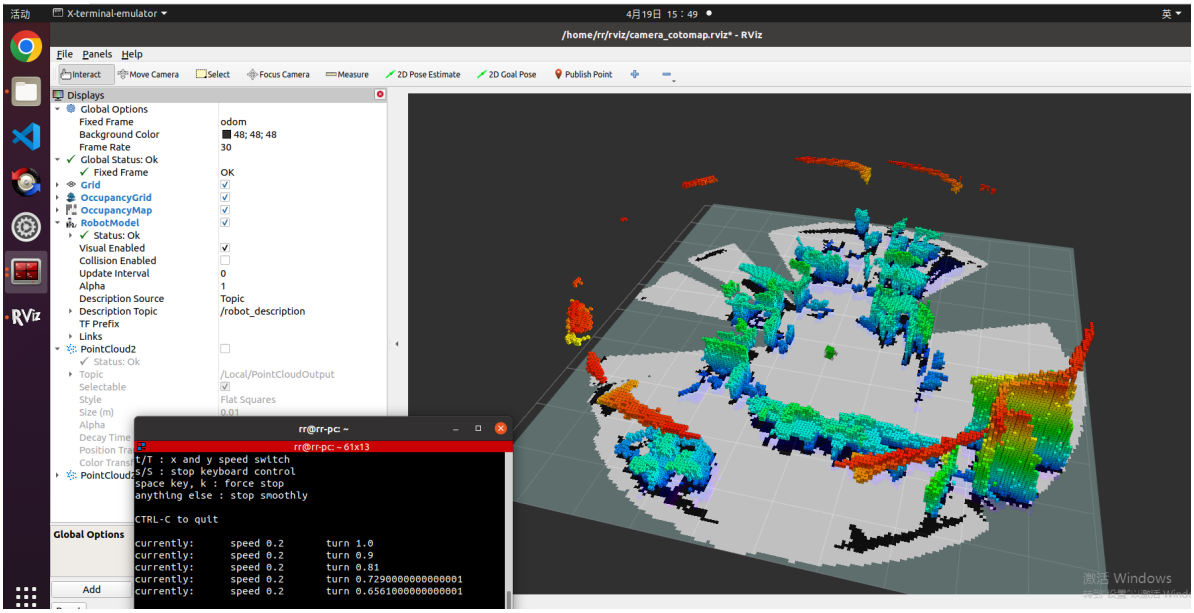
2. Start octomap_server map

```
ros2 launch yahboomcar_slam camera_octomap_launch.py
```

3. Open rviz in docker or virtual machine [recommended]:

```
ros2 launch yahboomcar_slam display_octomap_launch.py
```

4. Use the remote control or keyboard to control the node to slowly move the robot to build the map. Losing key frames may cause the map to fail.



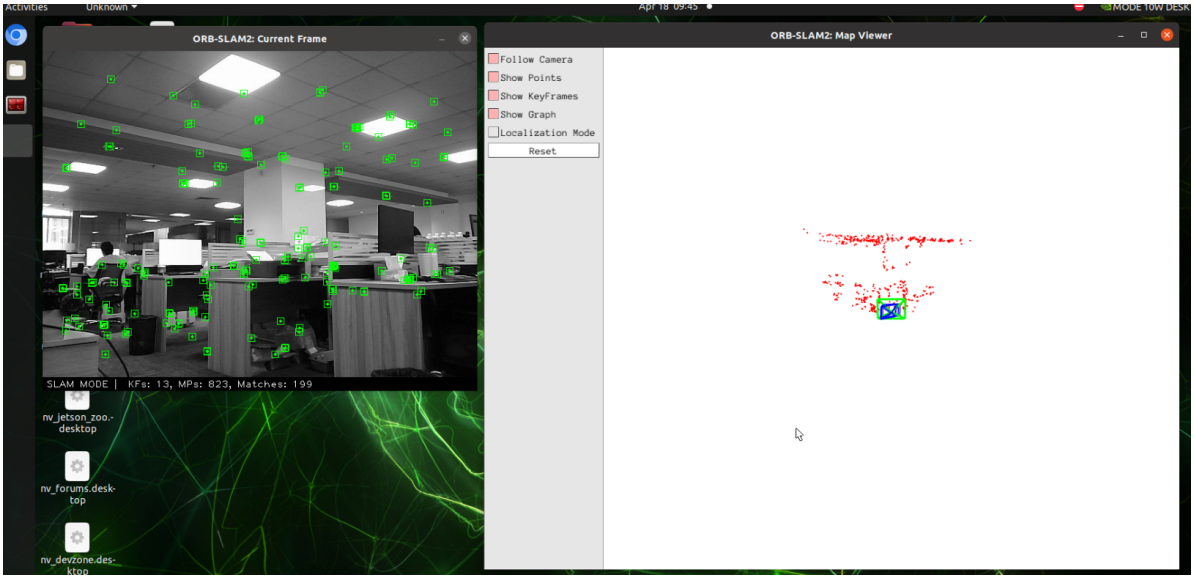
7.3, Octomap construction based on orbslam and pointcloud_mapping

1, Start the camera

```
#Camera startup
ros2 launch astra_camera gemini.launch.xml
```

2, Start orbslam to publish the camera pose, color map and depth map. Depending on the performance of different master controllers, the waiting time here is about 10 seconds

```
ros2 launch yahboomcar_slam orbslam_base_launch.py
```



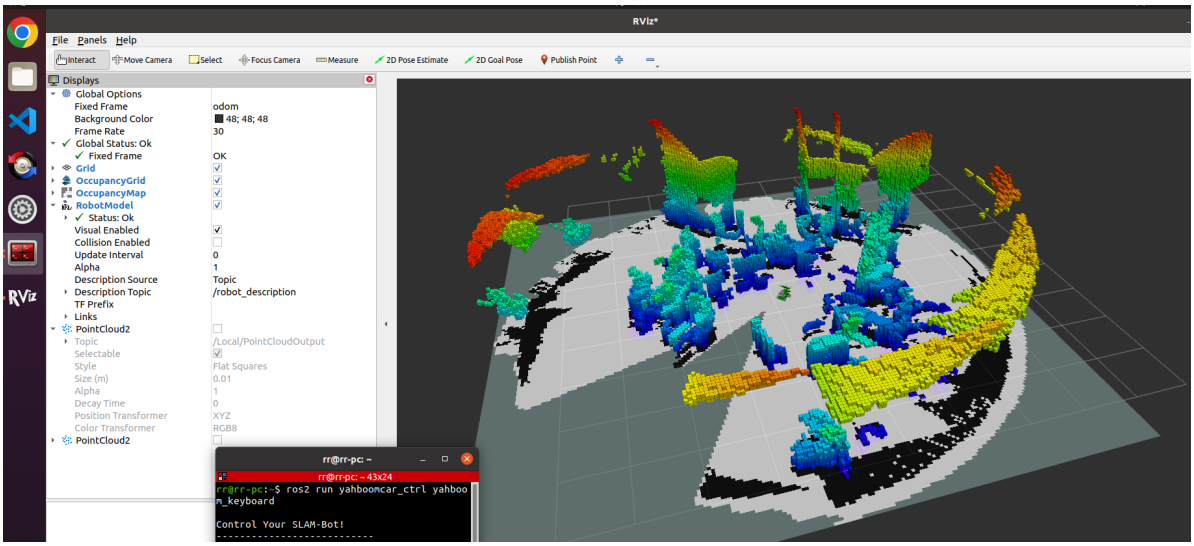
3, Start octomap_server to build a map

```
ros2 launch yahboomcar_slam orbslam_pcl_octomap_launch.py
```

4. Start rviz:

```
ros2 launch yahboomcar_slam display_octomap_launch.py
```

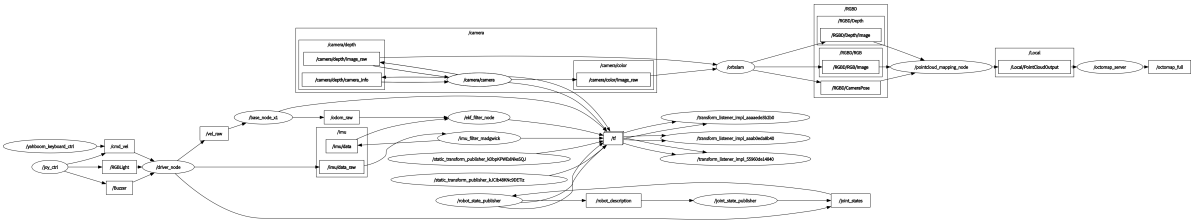
5. Use the remote control or keyboard to control the node to slowly move the robot to build a map. Losing key frames may cause the map to fail.



7.4、Node analysis

7.3.1、Display calculation graph

```
rqt_graph
```



7.3.2、Details of each node

```
rr@rr-pc:~/rviz$ ros2 node info /pointcloud_mapping_node
/pointcloud_mapping_node
  Subscribers:
    /RGBD/CameraPose: geometry_msgs/msg/PoseStamped
    /RGBD/Depth/Image: sensor_msgs/msg/Image
    /RGBD/RGB/Image: sensor_msgs/msg/Image
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /Global/PointCloudOutput: sensor_msgs/msg/PointCloud2
    /Local/PointCloudOutput: sensor_msgs/msg/PointCloud2
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /pointcloud_mapping_node/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /pointcloud_mapping_node/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /pointcloud_mapping_node/get_parameters: rcl_interfaces/srv/GetParameters
    /pointcloud_mapping_node/list_parameters: rcl_interfaces/srv/ListParameters
    /pointcloud_mapping_node/set_parameters: rcl_interfaces/srv/SetParameters
    /pointcloud_mapping_node/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:

  Action Clients:
```

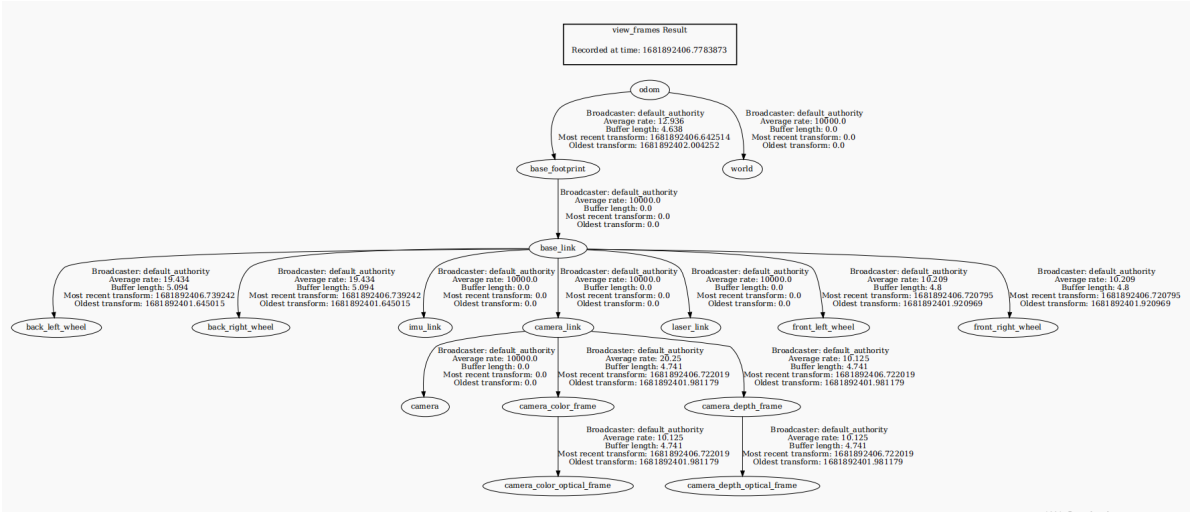
```
rr@rr-pc:~/rviz$ ros2 node info /octomap_server
/octomap_server
  Subscribers:
    /Local/PointCloudOutput: sensor_msgs/msg/PointCloud2
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /free_cells_vis_array: visualization_msgs/msg/MarkerArray
    /occupied_cells_vis_array: visualization_msgs/msg/MarkerArray
    /octomap_binary: octomap_msgs/msg/Octomap
    /octomap_full: octomap_msgs/msg/Octomap
    /octomap_point_cloud_centers: sensor_msgs/msg/PointCloud2
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /projected_map: nav_msgs/msg/OccupancyGrid
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /octomap_binary: octomap_msgs/srv/GetOctomap
    /octomap_full: octomap_msgs/srv/GetOctomap
    /octomap_server/clear_bbox: octomap_msgs/srv/BoundingBoxQuery
    /octomap_server/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /octomap_server/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /octomap_server/get_parameters: rcl_interfaces/srv/GetParameters
    /octomap_server/list_parameters: rcl_interfaces/srv/ListParameters
    /octomap_server/reset: std_srvs/srv/Empty
    /octomap_server/set_parameters: rcl_interfaces/srv/SetParameters
    /octomap_server/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:

  Action Clients:
```

7.3.3、TF transformation

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


view_frames Result
Recorded at time: 1681892406.7783873
odom
Broadcaster: default_authority
Average rate: 12.936
Buffer length: 4.638
Most recent transform: 1681892406.642514
Oldest transform: 1681892402.004252
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base_footprint
world
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: 19.434
Buffer length: 5.094
Most recent transform: 1681892406.739242
Oldest transform: 1681892401.645015
Broadcaster: default_authority
Average rate: 19.434
Buffer length: 5.094
Most recent transform: 1681892406.739242
Oldest transform: 1681892401.645015
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.8
Most recent transform: 1681892406.720795
Oldest transform: 1681892401.920969
Broadcaster: default_authority
Average rate: 10.209
Buffer length: 4.8
Most recent transform: 1681892406.720795
Oldest transform: 1681892401.920969
back_left_wheel
back_right_wheel
imu_link
camera_link
laser_link
front_left_wheel
front_right_wheel
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: default_authority
Average rate: 20.25
Buffer length: 4.741
Most recent transform: 1681892406.722019
Oldest transform: 1681892401.981179
Broadcaster: default_authority
Average rate: 10.125
Buffer length: 4.741
Most recent transform: 1681892406.722019
Oldest transform: 1681892401.981179
camera
camera_color_frame
camera_depth_frame
Broadcaster: default_authority
Average rate: 10.125
Buffer length: 4.741
Most recent transform: 1681892406.722019
Oldest transform: 1681892401.981179
Broadcaster: default_authority
Average rate: 10.125
Buffer length: 4.741
Most recent transform: 1681892406.722019
Oldest transform: 1681892401.981179
camera_color_optical_frame
camera_depth_optical_frame