

7.ORB_SLAM2 Octomap mapping icon

This routine is implemented on Orin NX and cannot be implemented on a virtual machine. We are only explaining how the process is implemented. If you want to implement this function on your own motherboard, you need to compile the entire feature pack and connect relevant peripherals.

7.ORB_SLAM2 Octomap mapping icon

- 1.octomap mapping based camera
2. octomap mapping based on orbslam andpointcloud_mapping
3. Node analysis

octomap website: <http://octomap.github.io/>

octomap 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 configurations for software and hardware are as follows:

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

1.octomap mapping based camera

Enter the Docker container and execute the following launch file on the sub terminal:

1)Start camera

```
ros2 launch orbbec_camera orbbec_camera.launch.py
```

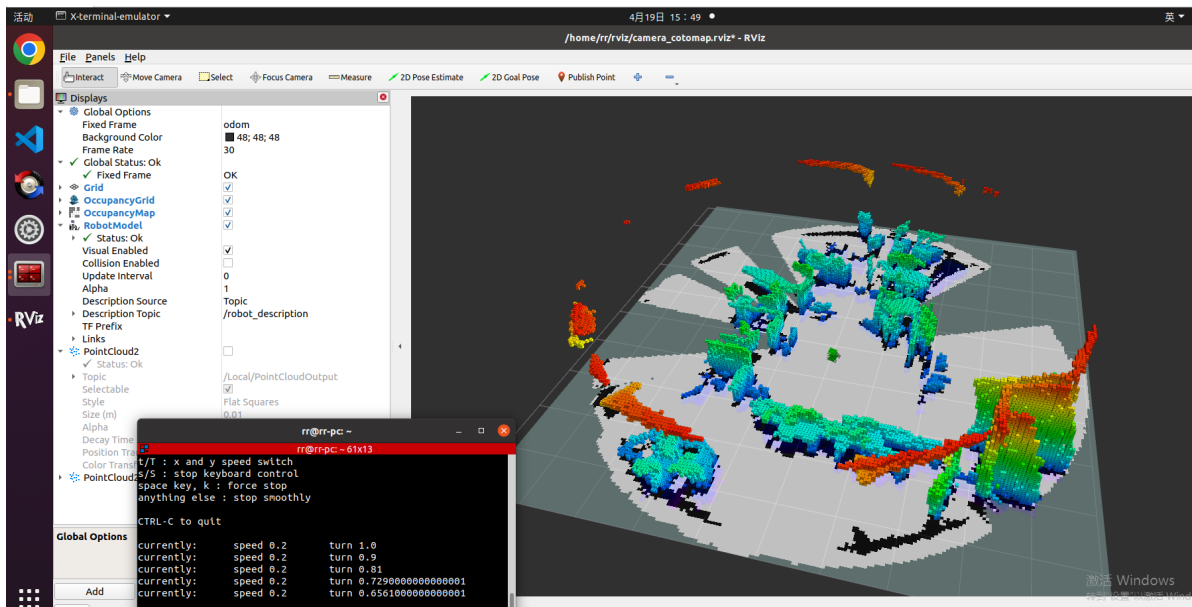
2)Start octomap_ Server mapping

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

3)Enable rviz in Docker or Virtual Machine:

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

4)Using a remote control or keyboard to control the node to slowly move the machine to create a map, losing keyframes may cause the map to fail.



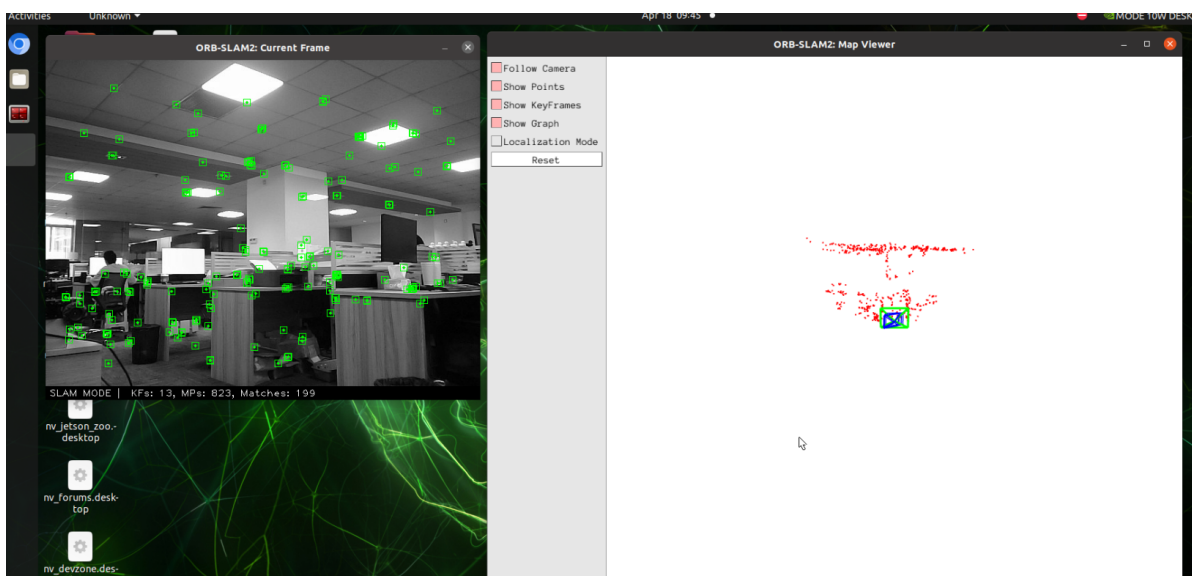
2. octomap mapping based on orbslam and pointcloud_mapping

1) Start camera

```
ros2 launch orbbec_camera orbbec_camera.launch.py
```

2) Launch orbslam to release camera pose, color and depth maps. Depending on the performance of different controllers, the waiting time here is approximately within 10 seconds

```
ros2 launch yahboomcar_slam orbslam_base.launch.py
```



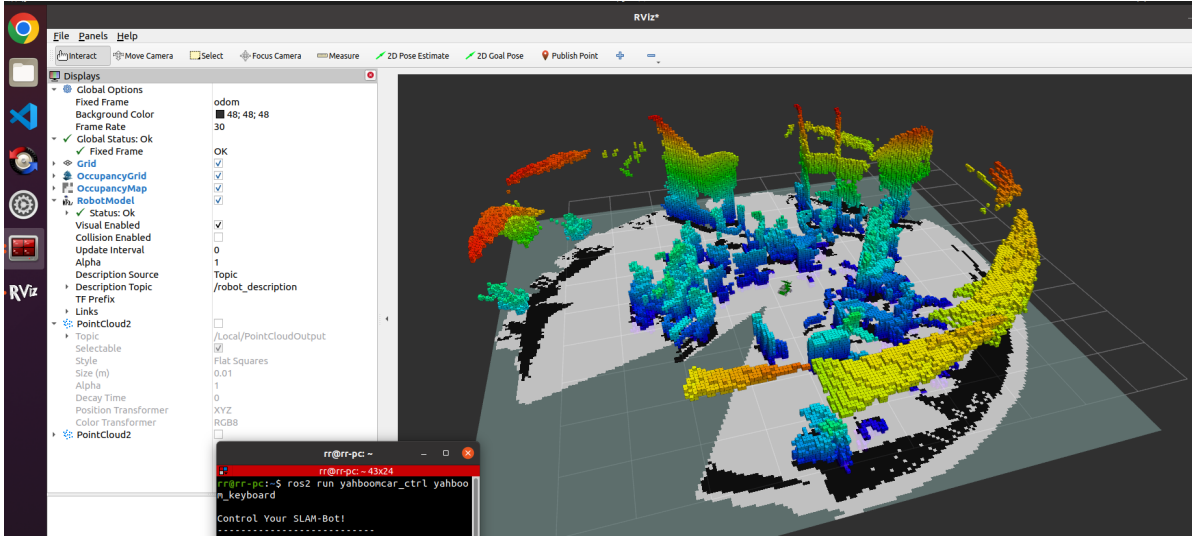
3) Start octomap_server mapping

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

4) Start rviz

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

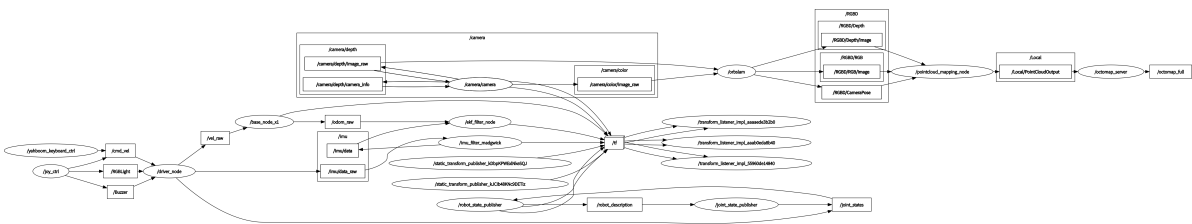
5) Using a remote control or keyboard to control the node to slowly move the machine to create a map, losing keyframes may cause the map to fail.



3. Node analysis

1) Display calculation chart

rqt_graph



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:

```

3)TF transformation

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

