

# 7、ORB\_SLAM2 Octomap mapping

---

## 7、ORB\_SLAM2 Octomap mapping

- 7.1、brief introduction
- 7.2、Camera based octomap mapping
- 7.3、Based on orbslam and pointcloud\_ Octomap mapping for mapping
- 7.4、Node resolution
  - 7.3.1、Display Calculation Chart
  - 7.3.2、Details of each node
  - 7.3.3、TF transformation

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](http://wiki.ros.org/octomap_server)

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

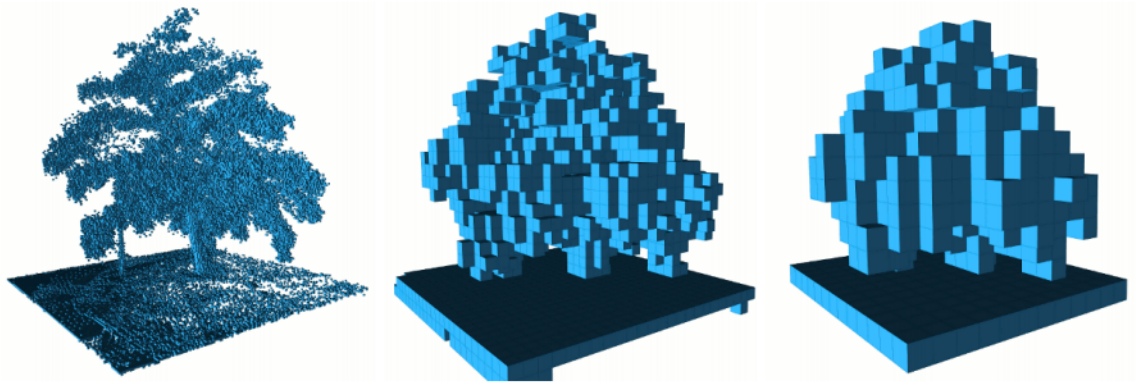
- Reference model: ROSMASTER X3
- Robot hardware configuration: Arm series main control, Silan A1 LiDAR, AstraPro Plus depth camera
- Robot system: Ubuntu (version not required)+Docker (version 20.10.21 and above)
- PC virtual machine: Ubuntu (20.04)+ROS2 (Foxy)
- Usage scenario: Use on a relatively clean 2D plane

## 7.1、brief introduction

---

Octomap is a 3D map creation tool based on Octree, which can display complete 3D graphics including accessible areas and unmapped areas, and sensor data based on occupancy grid can be fused and updated in multiple measurements; Maps can provide multiple resolutions, data can be compressed, and storage is compact In fact, the code for octomap mainly consists of two modules: the 3D map creation tool octomap and the visualization tool octovis.

Compared to point clouds, it can save a lot of space. The map created by octomap is roughly like this: (different resolutions from left to right)



**Fig. 3** By limiting the depth of a query, multiple resolutions of the same map can be obtained at any time. Occupied voxels are displayed in resolutions 0.08 m, 0.64 , and 1.28 m.

## 7.2 、 Camera based octomap mapping

**Note:** The commands or locations mentioned below are all those in the Docker container unless otherwise specified.

Enter the Docker container, refer to **【Docker Course -5. Entering the Robot Docker Container】** , and execute the following launch file on the terminal:

```
#Raspberry Pi 5 master user
~/run_docker.sh
#Orin master user
~/ros2_orbslam2.sh
```

Here, take orin master as an example and modify the corresponding car model

```
[System Information]
IP_Address_1: 192.168.2.102
IP_Address_2: 172.17.0.1
-----
ROS_DOMAIN_ID: 28 | ROS: humble
my_robot_type: x3 | my_lidar: s2 | my_camera: astraplus
-----
jetson@yahboom:~$ ./ros2_orbslam2.sh
access control disabled, clients can connect from any host
WARNING: Published ports are discarded when using host network mode
-----
ROS_DOMAIN_ID: 28
ros-foxy: OrbSlam2 | my_robot_type: x3 | my_camera: astraplus
-----
root@yahboom:/#
```

1、 Start camera code

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

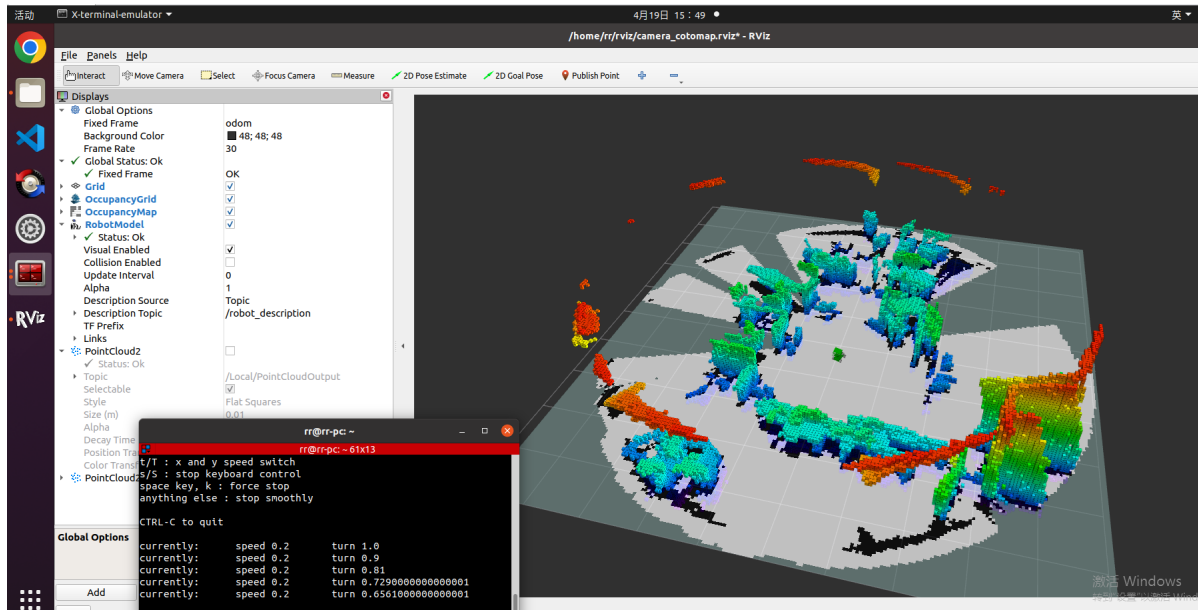
## 2、 Start octomap\_ Server mapping

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

## 3、 Enable rviz in Docker or Virtual Machine **[Recommended]** :

```
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.



## 7.3、 Based on orbslam and pointcloud\_ Octomap mapping for mapping

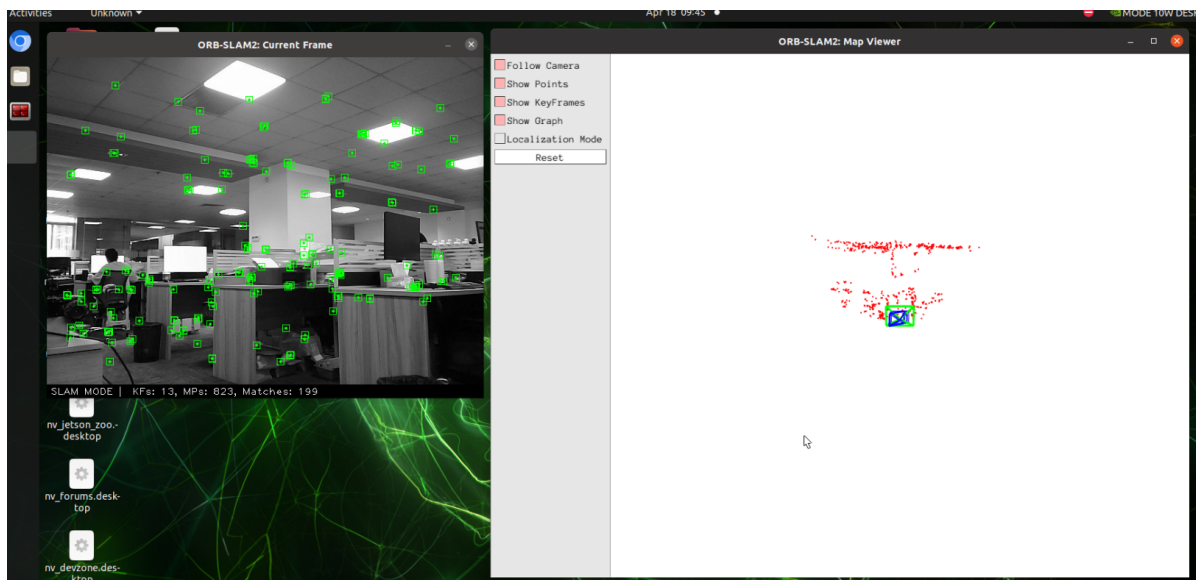
Enter the Docker container, refer to **[Docker Course -5. Entering the Docker Container for Robot]** , and execute the following launch file on each terminal:

### 1、 Start camera code

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

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
```



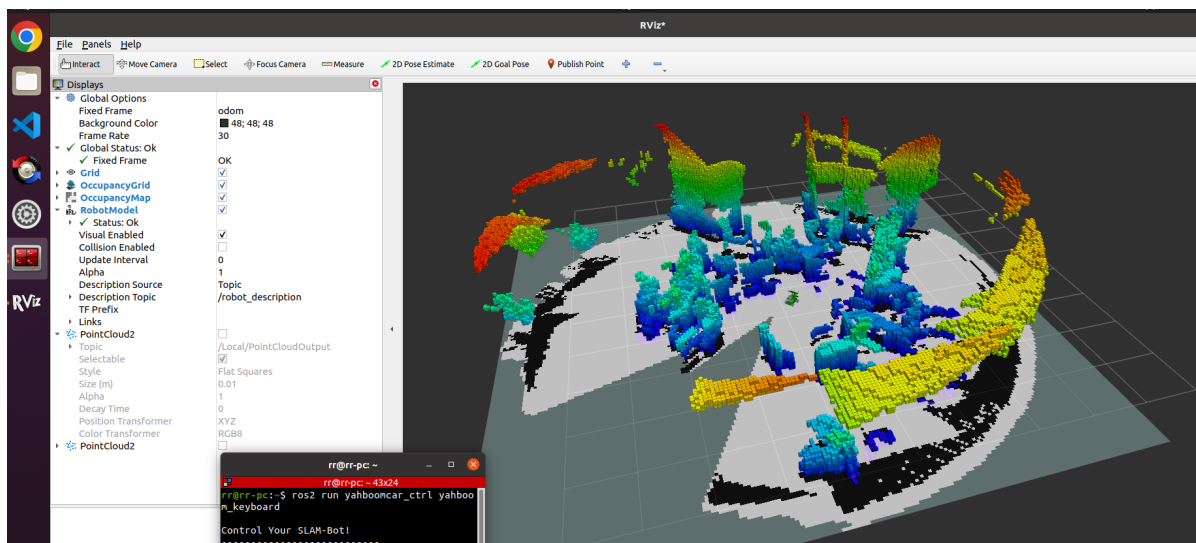
### 3、 Start octomap\_Server mapping

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

### 4、 Enable rviz in Docker or Virtual Machine **[Recommended]**:

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

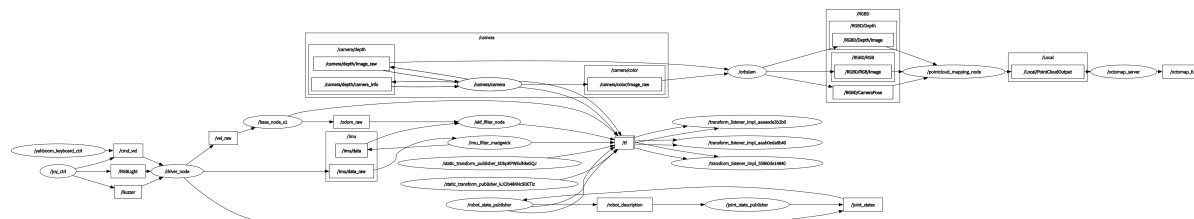
5、 Using a remote control or keyboard to control the node to slowly move the machine to create a map, losing keyframes may cause the map to fail.



## 7.4、 Node resolution

### 7.3.1、 Display Calculation Chart

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
```

