9 ORB_SLAM2_Octomap

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

9.1 Introduction

Octomap uses an octree data structure to store probabilistic occupancy maps of 3D environments. The OctoMap library implements a 3D occupancy grid mapping method, provides data structures and mapping algorithms in C++, and is especially suitable for robots. The map implementation is based on octrees.

It compresses, updates the map elegantly, and has adjustable resolution!It stores the map in the form of an octotree(described later), which saves a lot of space compared to point clouds. The map created by octomap looks like this:(different resolutions from left to right)

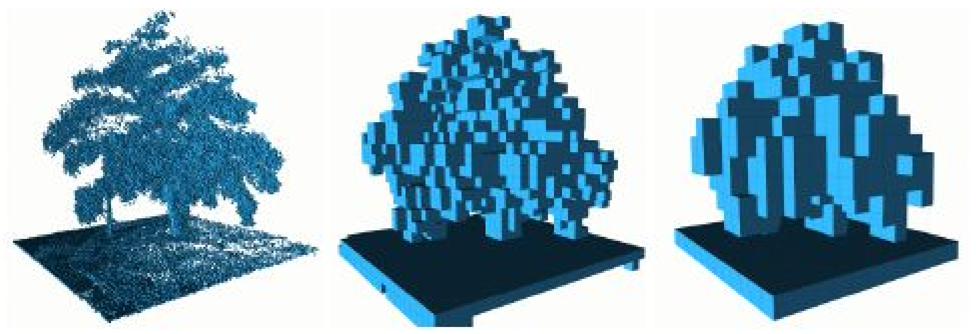


Fig. 3 By limiting the depth of a query, multiple resolutions of the same map can be obtained at any time. Occupied voxels are displayed in resolutions 0.08 m, 0.64 , and 1.28 m.

- Precautions

Note: When building a map, moving the robot slowly and losing keyframes may cause the map to fail.

According to different models, you only need to set the purchased model in [.bashrc], X1(ordinary four-wheel drive) X3(Mike wheel) X3plus(Mike wheel mechanical arm) R2(Ackerman differential) and so on. Section takes X3 as an example

Open the [.bashrc] file

```
sudo vim .bashrc
```

Find the [ROBOT_TYPE] parameter and modify the corresponding model

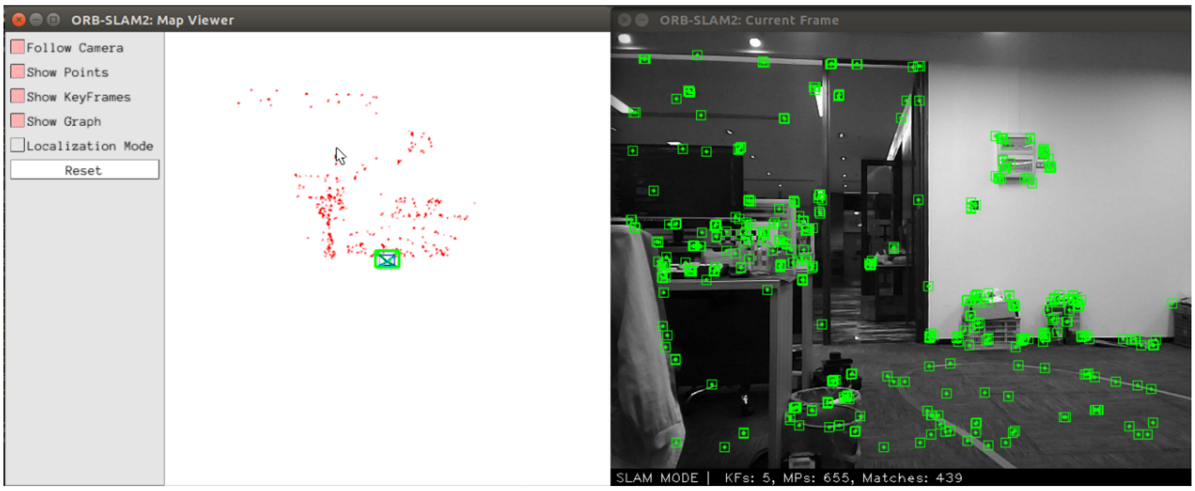
```
export  ROBOT_TYPE=X3    # ROBOT_TYPE: X1 X3 X3plus R2 X7
```

9.2 Use

Start orb_slam and the underlying driver(Robot side)

```
roslaunch yahboomcar_slam robot_orb_slam.launch bUseViewer:=true
```

- [bUseViewer] parameter: whether to open the visualization window of orbslam. If true, you can clearly view the key points. If the positioning is unsuccessful, you can reset the key points. Click [Reset] on the left side of the figure below.



SLAM MODE | KFs: 5, MPs: 655, Matches: 439

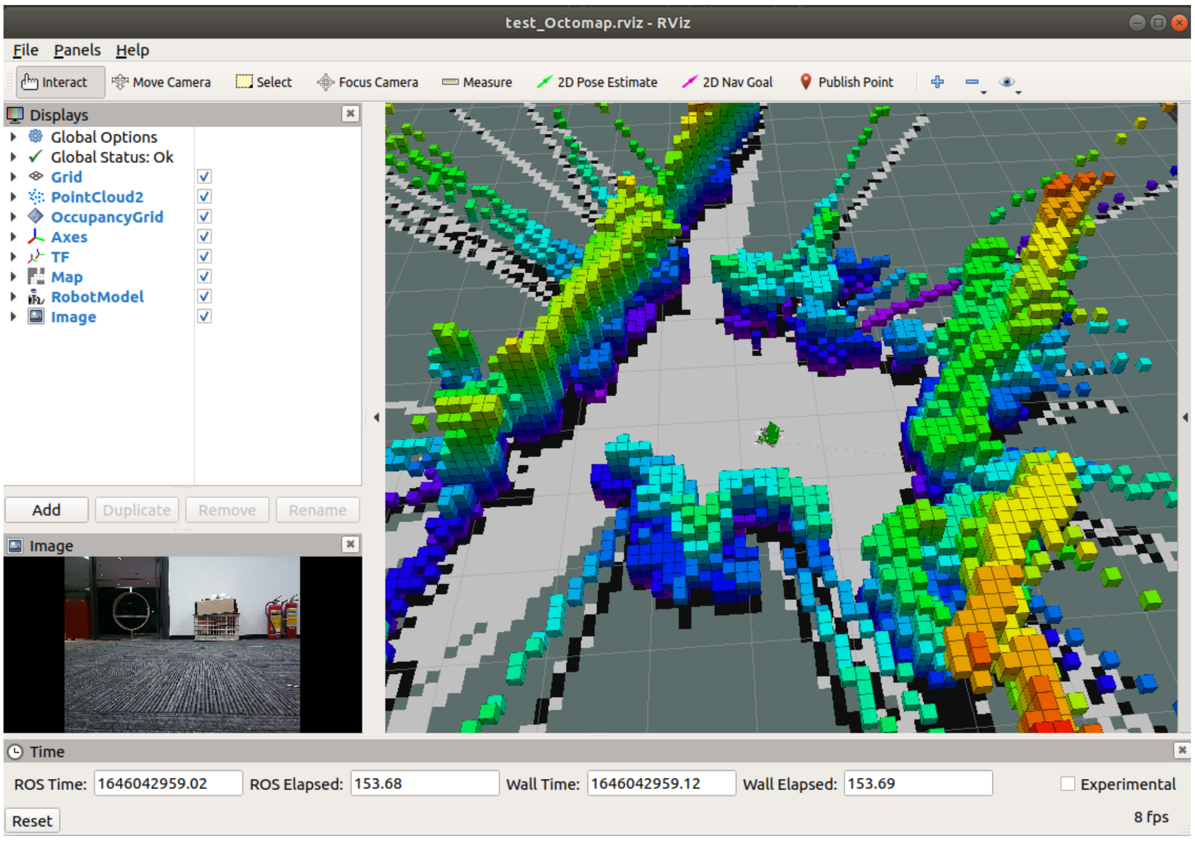
Start octree mapping(Robot side)

```
roslaunch yahboomcar_slam robot_orb_octomap.launch frame_id:=odom use_rviz:=false
```

- [frame_id] parameter: coordinate system name, available by default without setting.
- [use_rviz] parameter: whether to enable rviz.

Open the visual interface(virtual machine side)

```
roslaunch yahboomcar_slam view_orb_octomap.launch
```



ROS Time: 1646042959.02 ROS Elapsed: 153.68 Wall Time: 1646042959.12 Wall Elapsed: 153.69

Because of the octree, its map looks like a lot of small blocks(much like minecraft). When the resolution is high, the square is small; when the resolution is low, the square is large. Each square represents the probability that the square is occupied.

9.3 octomap_server

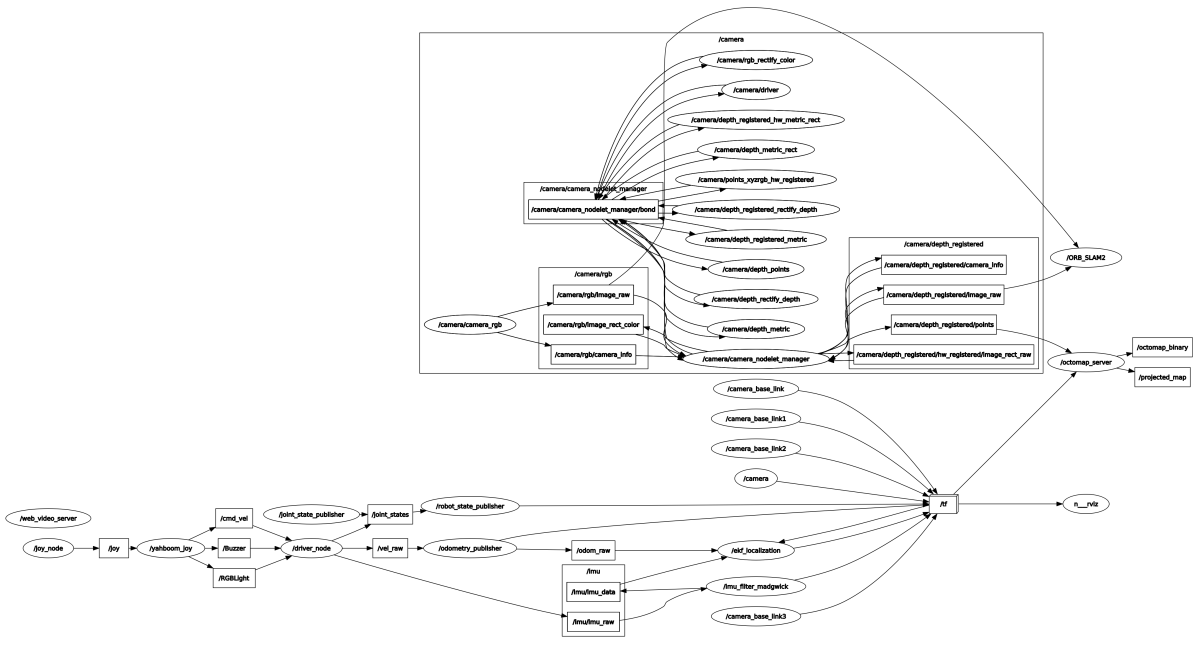
9.3.1 Topics and Services

Subscribe to topics	type	describe
cloud_in	sensor_msgs/PointCloud2	Incoming 3D point cloud for scan integration.
Post a topic	**type**	**describe**
octomap_binary	octomap_msgs/Octomap	The complete maximum-likelihood occupancy map is a compact octal-mapped binary stream, encoding free space and occupancy space. Binary messages only differentiate between free space and occupied space, but smaller.
octomap_full	octomap_msgs/Octomap	The complete maximum-likelihood occupancy map is a compact octal-mapped binary stream, encoding free space and occupancy space. The complete message contains the complete probability and all additional data stored in the tree.
occupied_cells_vis_array	visualization_msgs/MarkerArray	In RViz, all occupied voxels are marked as "boxes" for visualization

Subscribe to topics	type	describe
octomap_point_cloud_centers	sensor_msgs/PointCloud2	The centers of all occupied voxels serve as a point cloud, useful for visualization. Note that this will have gaps because points have no volume size and octagonal voxels can have different resolutions!
map	nav_msgs/OccupancyGrid	Project a 2D occupancy map down from a 3D map.
Serve	**type**	**describe**
octomap_binary	octomap_msgs/GetOctomap	The complete maximum-likelihood occupancy map is a compact octal-mapped binary stream, encoding free space and occupancy space.
clear_bbx	octomap_msgs/BoundingBoxQuery	Clear the area in the 3D occupancy map, set all voxels in that area to "free"
reset	std_srvs/Empty	reset the entire map

Node view

```
rqt_graph
```



9.3.2 Configuration parameters

parameter	type	Defaults	describe
frame_id	string	/map	The static global frame in which the map will be published. When building the map dynamically, the sensor data needs to be converted to this frame.
resolution	float	0.05	The resolution of the map(in meters) when starting from an empty map. Otherwise the resolution of the loaded file will be used.
base_frame_id	string	base_footprint	Robot pedestal for ground plane detection(if enabled)
height_map	bool	true	Should the visualization encode the height in a different color
color/[r/g/b/a]	float		When ~heigh_map=False, display the color of occupied cells in the [0:1] range
sensor_model/max_range	float	-1(unlimited)	The maximum extent(in meters) to insert point cloud data when building a map dynamically. Limiting the range to a useful range(e.g. 5 meters) prevents false false points away from the robot.
sensor_model/[hit | miss]	float	0.7/0.4	Hit and miss probabilities in sensor models when building maps dynamically
sensor_model/[min | max]	float	0.12/0.97	Clamping min and max probabilities when building maps dynamically
latch	bool	True for static mapping, false if no initial mapping is given	Whether the topic is locked for publication or only published once per change. For best performance when building maps(frequently updated), set this to false. When set to true, all changes on every map will create all themes and visualizations.
filter_ground	bool	false	Whether the ground plane should be detected and ignored from scan data when building the map dynamically using pcl::SACMODEL_vertical_plane. This clears everything on the ground, but doesn't insert the ground into the map as an obstacle. If this feature is enabled, the ~ground_filter/... parameter can be further configured.

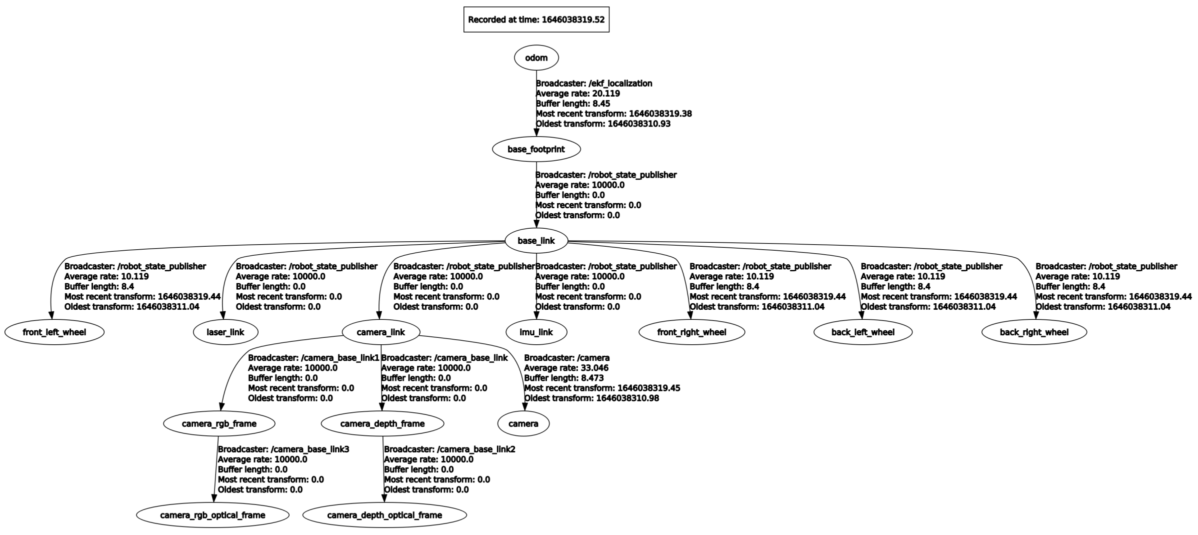
parameter	type	Defaults	describe
ground_filter/distance	float	0.04	Distance threshold of the point(z direction) to split to the ground plane
ground_filter/angle	float	0.15	The angle threshold between the detected plane and the horizontal plane detected as the ground
ground_filter/plane_distance	float	0.07	Distance threshold for distance z=0 for planes to be detected as ground(fourth coefficient of PCL plane equation)
pointcloud *[min | max]* z	float	-/+ infinity	Insert the minimum and maximum heights to be considered in the callback.
occupancy *[min | max]* z	float	-/+ infinity	Minimum and maximum heights to consider in the final map.

9.3.3 TF transformation

Required TF Transform	describe
sensor data frame --> /map	For scan integration, the sensor data needs to be converted into a global map frame. This information needs to be obtained from an external SLAM or localized node.

View tf tree

```
rosrun rqt_tf_tree rqt_tf_tree
```


Recorded at time: 1646038319.52
odom
Broadcaster: /ekf_localization
Average rate: 20.119
Buffer length: 8.45
Most recent transform: 1646038319.38
Oldest transform: 1646038310.93
base_footprint
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base_link
Broadcaster: /robot_state_publisher
Average rate: 10.119
Buffer length: 8.4
Most recent transform: 1646038319.44
Oldest transform: 1646038311.04
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 10.119
Buffer length: 8.4
Most recent transform: 1646038319.44
Oldest transform: 1646038311.04
Broadcaster: /robot_state_publisher
Average rate: 10.119
Buffer length: 8.4
Most recent transform: 1646038319.44
Oldest transform: 1646038311.04
Broadcaster: /robot_state_publisher
Average rate: 10.119
Buffer length: 8.4
Most recent transform: 1646038319.44
Oldest transform: 1646038311.04
front_left_wheel
laser_link
camera_link
imu_link
front_right_wheel
back_left_wheel
back_right_wheel
Broadcaster: /camera_base_link1
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /camera_base_link
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /camera
Average rate: 33.046
Buffer length: 8.473
Most recent transform: 1646038319.45
Oldest transform: 1646038310.98
camera_rgb_frame
camera_depth_frame
camera
Broadcaster: /camera_base_link3
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /camera_base_link2
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
camera_rgb_optical_frame
camera_depth_optical_frame

9.4 Extended test

Note: This case belongs to the test version and is related to the performance of the device. If the performance is too low, it may not start normally.

Start the underlying driver +orb_slam(virtual machine side)

```
roslaunch yahboomcar_slam test_orb_slam.launch
```

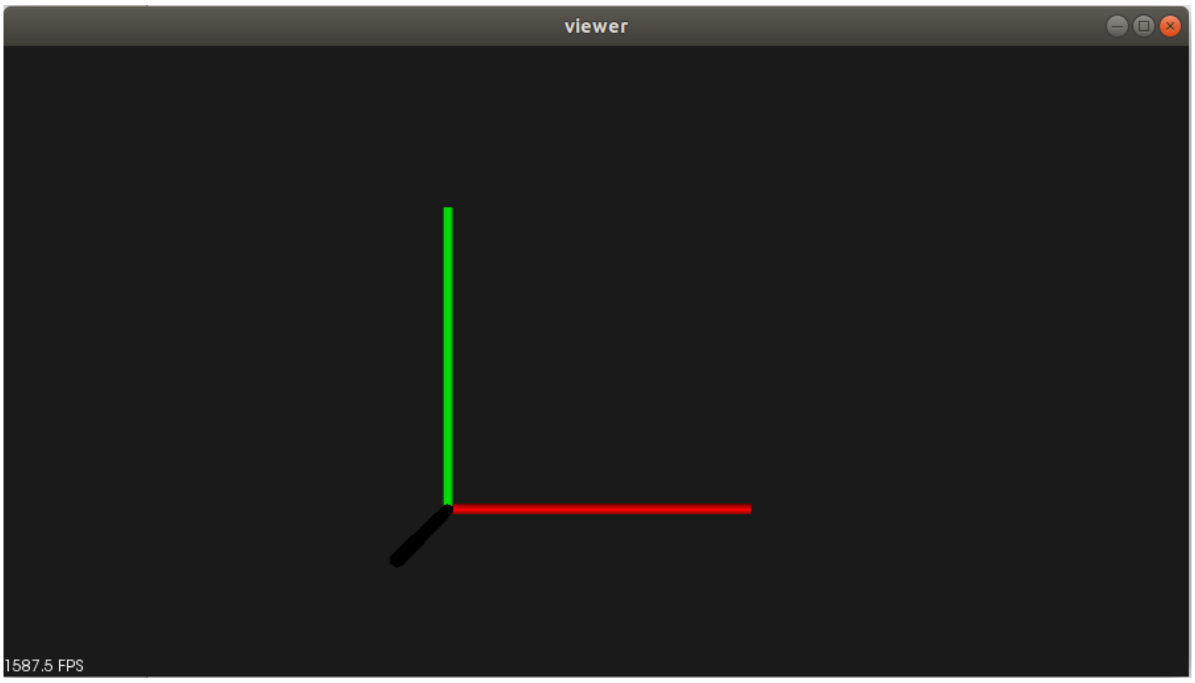
- [bUseViewer] parameter: whether to open the visualization window of orbslam.

Start octree mapping(virtual machine side)

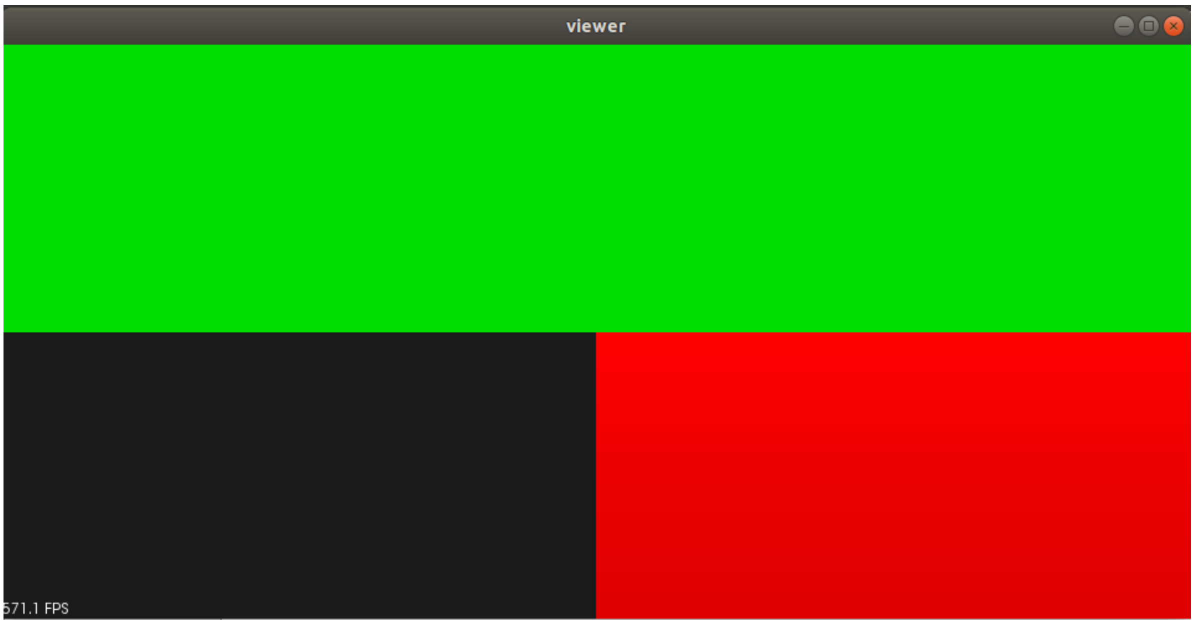
```
roslaunch yahboomcar_slam test_pcl_mapping.launch
```

- [use_viewer] parameter: whether to open the visualization window of pcl_viewer.
- [local_frame_id] parameter: local map coordinate system.
- [global_frame_id] parameter: global map coordinate system.

After opening, the [viewer] window will pop up, as shown below



Move the camera slowly, when a picture appears, as shown below



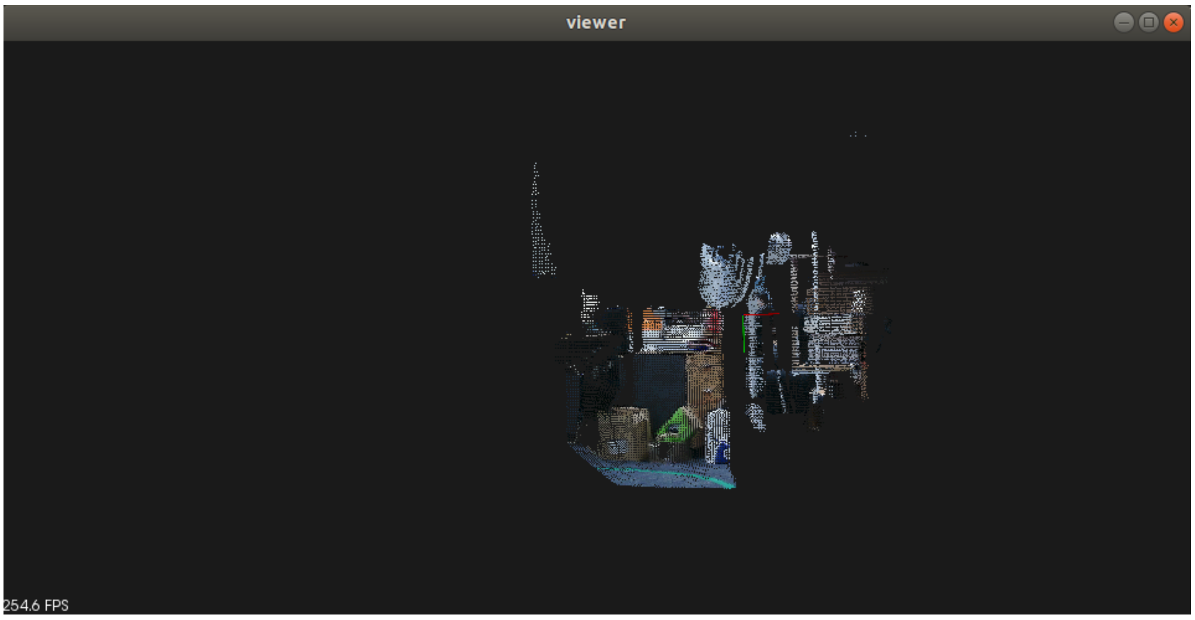
You need to scale and rotate the coordinate system as shown below

Scroll wheel: zoom in and out

Press and hold the scroll wheel: pan

Left mouse button: rotate

Right mouse button: zoom in and out



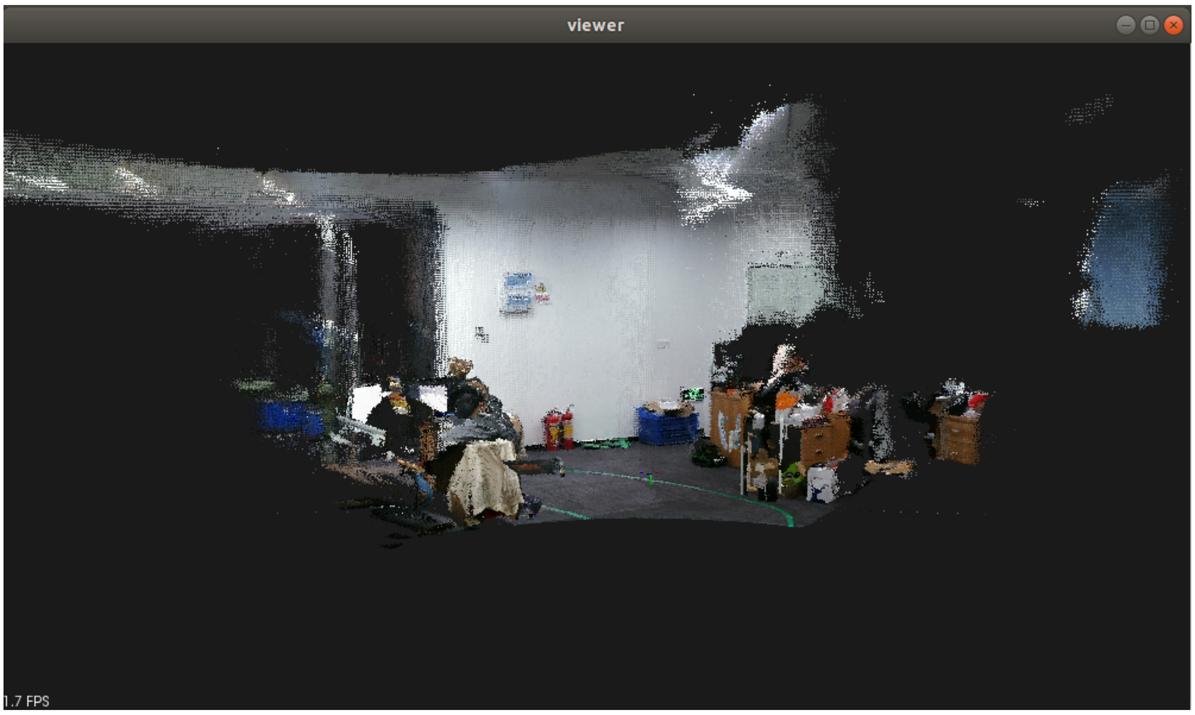
Slowly move the camera to build the map as shown below. After the construction is completed, [Ctrl+c] closes, and the pcd point cloud file is automatically saved. The file name of the path under this function package is [resultPointCloudFile.pcd]

pcl_viewer installation command

```
sudo apt-get install pcl-tools
```

View, enter the directory where the pcd file is located

```
pcl_viewer resultPointCloudFile.pcd
```


viewer
1.7 FPS