6、Data conversion and point cloud

6.1、scan to image

Start up lidar

```
roslaunch rplidar_ros rplidar.launch
```

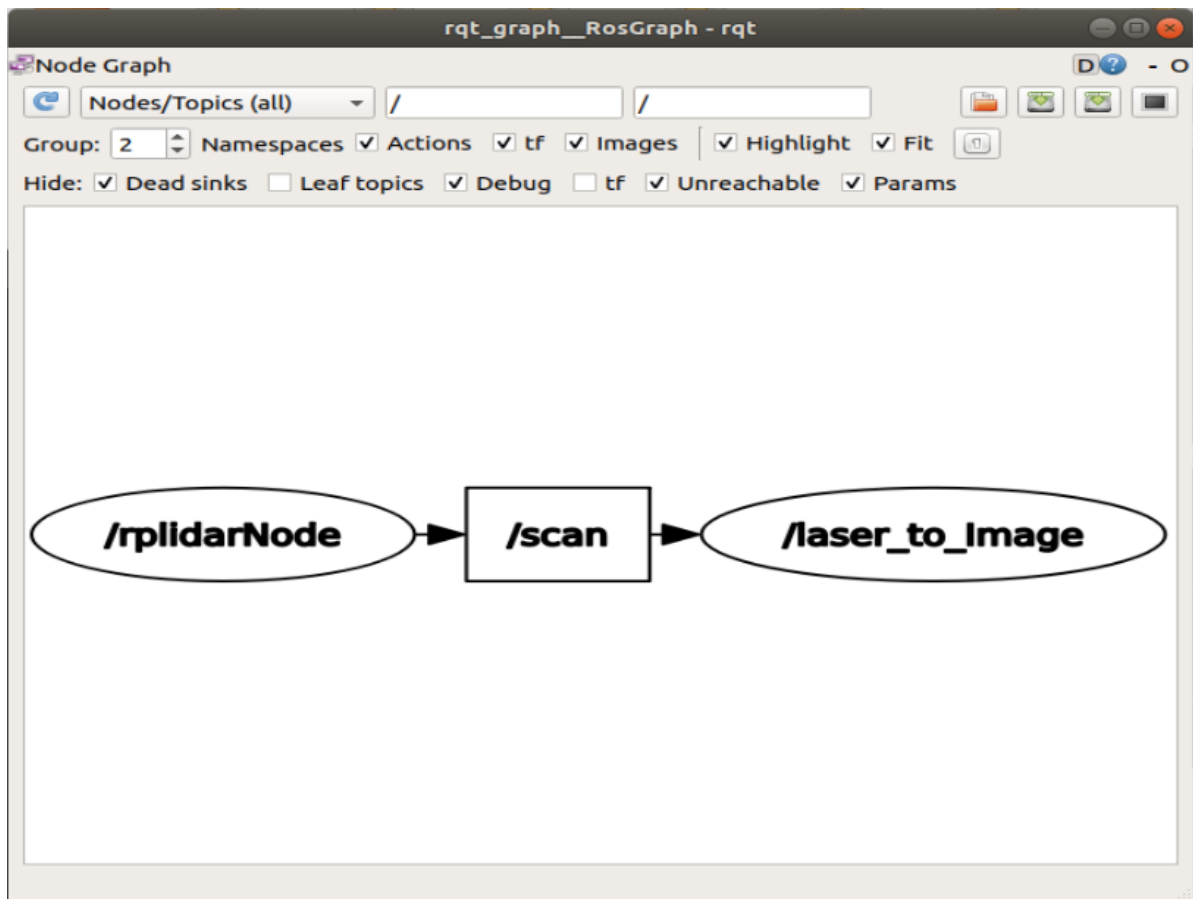- Start up

```
roslaunch transbot_visual laser_to_image.launch
rosrun transbot_visual laser_to_image.py
```


- View node graph

```
rqt_graph
```

- py Code analysis

Create subscribers and publishers

```
self.laserSub = rospy.Subscriber("/scan", LaserScan, self.laserCallback)  #
Receiving scan node
self.image_pub = rospy.Publisher('/laserImage', Image, queue_size=1) # Publish
the converted image information
```

Process the data in the callback function [self.laserCallback()] and publish it

```
def laserCallback(self, data):
    # Extract the received lidar data and convert it into point cloud data
    cloud_out = self.laserProj.projectLaser(data)
    lidar = point_cloud2.read_points(cloud_out)
    points = np.array(list(lidar))
    # Convert point cloud data into image data
    img = self.pointcloud_to_laserImage(points)
    # Convert image data into ROS image information and publish
    self.image_pub.publish(self.bridge.cv2_to_imgmsg(img))
    img=cv.resize(img,(640,480))
    cv.imshow("img", img)
    cv.waitKey(10)
    ROS_INFO("Published ... ");
```

6.2、ROS and PCD

Introduce several tools running on several ROS nodes. Their role is to convert between format point cloud or package and point cloud data (PCD file format).

Start the Astra camera

```
roslaunch astra_camera astraproplus.launch
```

Point cloud display: rviz (start the rviz command, select the corresponding topic, modify the parameters, and present different effects); pcl_visualization tool.

```
roslaunch transbot_visual pointCloud_visualize.launch
cloud_topic:=/camera/depth_registered/points
```

(1) pointcloud_to_pcd

```
rosrun pcl_ros pointcloud_to_pcd input:=/camera/depth/points
rosrun pcl_ros pointcloud_to_pcd input:=/camera/depth_registered/points
```

Save the ROS point cloud message in the specified PCD file.

(2) convert_pcd_to_image

```
rosrun pcl_ros convert_pcd_to_image <cloud.pcd>
```

Load a PCD file and publish it as a ROS image message five times per second.

(3) convert_pointcloud_to_image

```
rosrun pcl_ros convert_pointcloud_to_image
input:=/camera/depth_registered/points  output:=/my_image
View image: rosrun image_view image_view image:=/my_image
```

Subscribe to a topic of ROS point cloud and publish it with image information.

(4) pcd_to_pointcloud

```
rosrun pcl_ros pcd_to_pointcloud <file.pcd> [ <interval> ]
```

Load a PCD file and publish one or more times as a ROS point cloud message.

- file.pcd: The name of the (required) file to be read.
- interval: (Optional) The number of seconds to sleep between messages. If the parameter [interval] is zero or not specified, the message will be published once.

```
roslaunch transbot_visual pointCloud_visualize.launch cloud_topic:=/cloud_pcd
```

(5) bag_to_pcd

rosbag recording

Order: rosbag record topic1 [topic2 topic3 ...]

```
rosbag record /camera/depth_registered/points
```

bag_to_pcd

```
rosrun pcl_ros bag_to_pcd <input_file.bag> <topic> <output_directory>
#Eg:
rosrun pcl_ros bag_to_pcd 2021-09-09-11-41-56.bag
/camera/depth_registered/points my_pcd
```
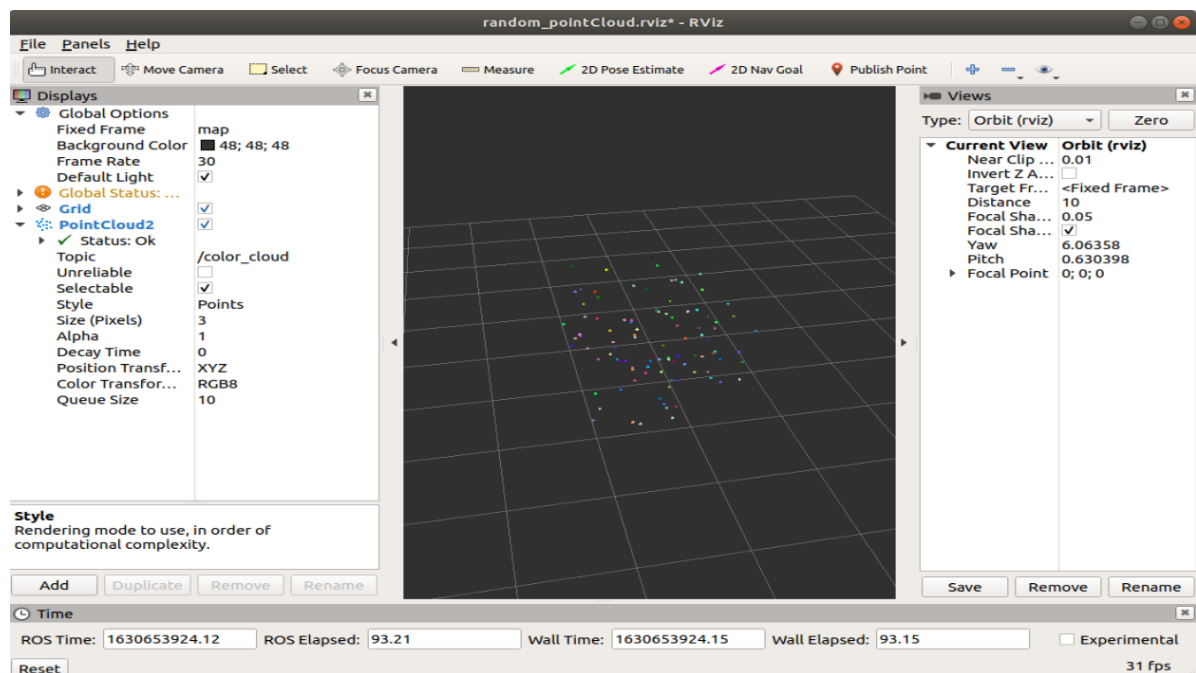
Read a package file and save the ROS point cloud message in the specified PCD file. This requires a bag file.

6.3、PCL 3D point cloud.

6.3.1、Start up

Release point cloud, the launch file contains the launch of rviz. So I can clearly see a cloud of dots flashing in the middle of rviz.

```
roslaunch transbot_visual pointCloud_pub.launch
```


Another way to start, this way you need to manually start [rviz], and add the component [PointCloud2] to select the topic [/color_cloud].

```
roscore
rosrun transbot_visual pointCloud_pub
```

- Code analysis

Code path: ~/transbot_ws/src/transbot_visual/src/pub_pointCloud.cpp

6.2.3、Point cloud visualization

- rviz

```
rviz
```

- pcl_visualization

Start up command

```
roslaunch transbot_visual pointCloud_visualize.launch
rosrun transbot_visual pointCloud_visualize
```


- shortcut key

【Ctrl】+【-】

【Shift】+【+】

【Alt】+【-】

【Alt】+【+】

- Code analysis

Code path: ~/transbot_ws/src/transbot_visual/src/pcl_visualize.cpp