ROS数据展示系统

一，用命令行窗口显示小车的IMU和里程计（odometry）数据

1. 显示IMU数据

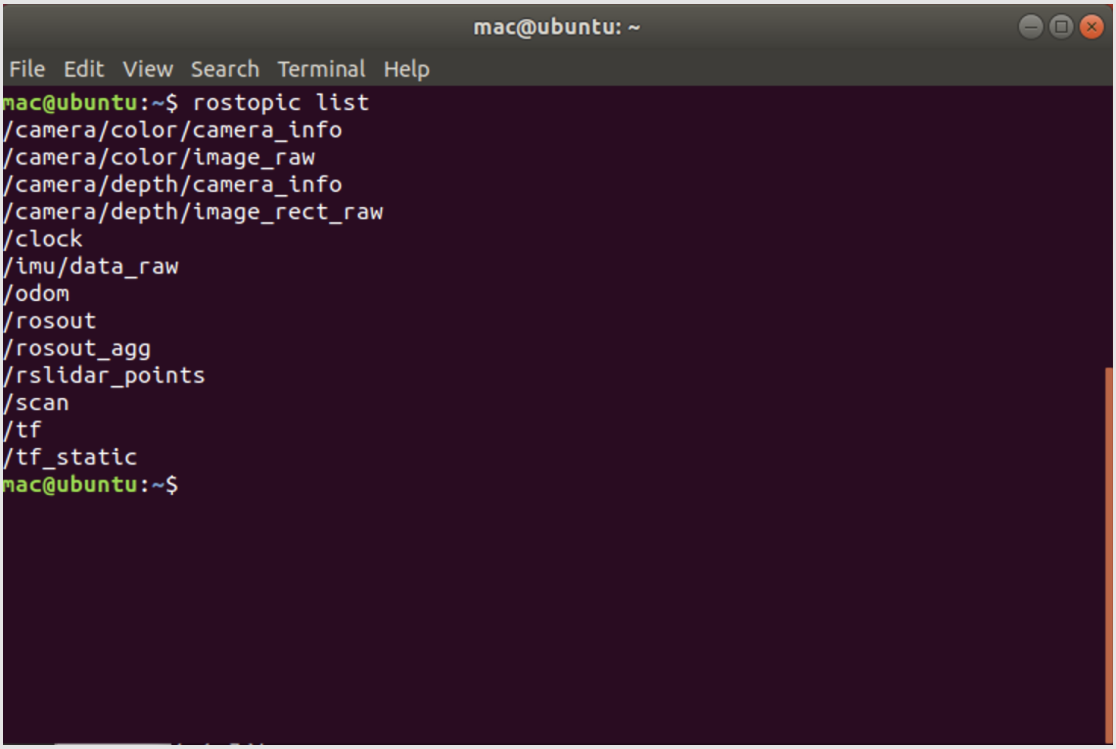
- 启动ROS

```
roscore http://ubuntu:11311/
File  Edit  View  Search  Terminal  Help
mac@ubuntu:~$ roscore
... logging to /home/mac/.ros/log/137e63aa-1295-11ee-9cf5-001c42c37cd1/roslaunch
-ubuntu-16279.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:43197/
ros_comm version 1.14.13


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.13

NODES

auto-starting new master
process[master]: started with pid [16289]
ROS_MASTER_URI=http://ubuntu:11311/
```

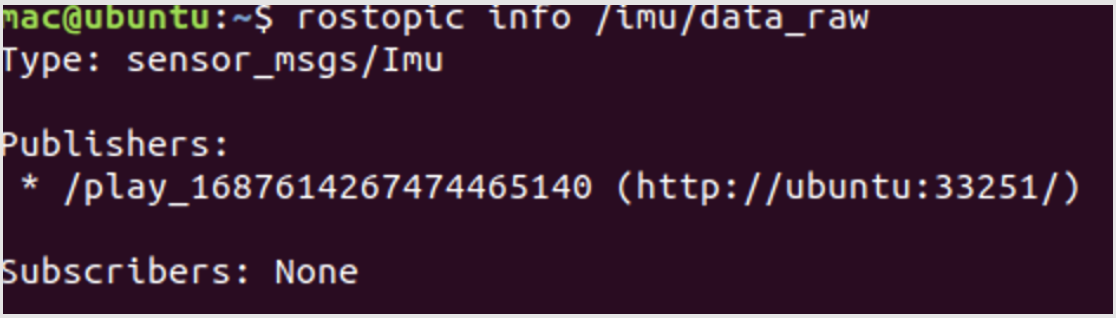
- 播放bag包数据

```
mac@ubuntu: ~/Desktop/ros_bag
File  Edit  View  Search  Terminal  Help
mac@ubuntu:~/Desktop/ros_bag$ rosbag play --pause all.bag
[ INFO] [1687614267.482243963]: Opening all.bag

Waiting 0.2 seconds after advertising topics... done.

Hit space to toggle paused, or 's' to step.
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
 [PAUSED ]  Bag Time: 1668162552.626376   Duration: 0.000000 / 79.848583
```

- 查看topic



- 查看话题消息的类型



- 显示消息具体内容

```
mac@ubuntu:~$ rosmsg show sensor_msgs/Imu
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
geometry_msgs/Quaternion orientation
  float64 x
  float64 y
  float64 z
  float64 w
float64[9] orientation_covariance
geometry_msgs/Vector3 angular_velocity
  float64 x
  float64 y
  float64 z
float64[9] angular_velocity_covariance
geometry_msgs/Vector3 linear_acceleration
  float64 x
  float64 y
  float64 z
float64[9] linear_acceleration_covariance
```

- 根据消息内容编写回调函数输出消息

```
void callback(const sensor_msgs::Imu::ConstPtr& ptr)
{
   std::cout << "std_msgs/Header header" << std::endl;
   std::cout << "    uint32 seq: " << ptr->header.seq <<
std::endl;
  std::cout << "    time stamp: " << ptr->header.stamp <<
std::endl;
  std::cout << "    string frame_id: " << ptr-
>header.frame_id << std::endl;
   std::cout << "geometry_msgs/Quaternion orientation" <<
std::endl;
   std::cout << "    float64 x: " << ptr->orientation.x <<
std::endl;
   std::cout << "    float64 y: " << ptr->orientation.y <<
std::endl;
   std::cout << "    float64 z: " << ptr->orientation.z <<
std::endl;
   std::cout << "    float64 w: " << ptr->orientation.w <<
std::endl;
   std::cout << "float64[9] orientation_covariance" <<
```

```
std::endl;
   std::cout << "    ";
   for (int i = 0; i < 9; ++i) {
       std::cout << ptr->orientation_covariance[i] << " ";
   }
   std::cout << std::endl;
   std::cout << "geometry_msgs/Vector3 angular_velocity" <<
std::endl;
   std::cout << "    float64 x: " << ptr-
>angular_velocity.x << std::endl;
   std::cout << "    float64 y: " << ptr-
>angular_velocity.y << std::endl;
   std::cout << "    float64 z: " << ptr-
>angular_velocity.z << std::endl;
   std::cout << "float64[9] angular_velocity_covariance" <<
std::endl;
   std::cout << "    ";
   for (int i = 0; i < 9; ++i) {
       std::cout << ptr->angular_velocity_covariance[i] <<
" ";
   }
   std::cout << std::endl;
   std::cout << "geometry_msgs/Vector3 linear_acceleration"
<< std::endl;
   std::cout << "    float64 x: " << ptr-
>linear_acceleration.x << std::endl;
   std::cout << "    float64 y: " << ptr-
>linear_acceleration.y << std::endl;
   std::cout << "    float64 z: " << ptr-
>linear_acceleration.z << std::endl;
   std::cout << "float64[9] linear_acceleration_covariance"
<< std::endl;
   std::cout << "    ";
   for (int i = 0; i < 9; ++i) {
       std::cout << ptr->linear_acceleration_covariance[i]
<< " ";
   }
   std::cout << std::endl;
   std::cout << "----------------------------------------
-------------------" >    >     << std::endl;
   return ;
}
```

- 订阅IMU话题

```
int main(int argc, char** argv)
{
  ros::init(argc, argv, "show_imu");
  ros::NodeHandle nodeHandle;
  ros::Subscriber subscriber =
nodeHandle.subscribe("/imu/data_raw", 1000, callback);
   ros::spin();
```

```
    return 0;
}
```

- 运行show_imu

```
mac@ubuntu:~/Desktop/ros_ws$ rosrun show_imu show_imu
std_msgs/Header header
    uint32 seq: 2180
    time stamp: 1668162552.625104072
    string frame_id: imu_link
geometry_msgs/Quaternion orientation
    float64 x: -0.00972918
    float64 y: 0.00102577
    float64 z: -0.00107328
    float64 w: 0.999952
float64[9] orientation_covariance
    0 0 0 0 0 0 0 0 0
geometry_msgs/Vector3 angular_velocity
    float64 x: 0
    float64 y: 0
    float64 z: 0
float64[9] angular_velocity_covariance
    0 0 0 0 0 0 0 0 0
geometry_msgs/Vector3 linear_acceleration
    float64 x: -0.0379779
```

2. **显示里程计数据**

- 查看话题消息的类型

```
mac@ubuntu:~$ rostopic info /odom
Type: nav_msgs/Odometry

Publishers:
 * /play_16876142674744651 40 (http://ubuntu:33251/)

Subscribers: None
```

- 显示消息具体内容

```
mac@ubuntu:~$ rosmsg show nav_msgs/Odometry
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
string child_frame_id
geometry_msgs/PoseWithCovariance pose
  geometry_msgs/Pose pose
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
  float64[36] covariance
geometry_msgs/TwistWithCovariance twist
  geometry_msgs/Twist twist
    geometry_msgs/Vector3 linear
      float64 x
```

- 根据消息内容编写回调函数输出消息

```
void callback(const nav_msgs::Odometry::ConstPtr& ptr)
{
   std::cout << "std_msgs/Header header" << std::endl;
   std::cout << "    uint32 seq: " << ptr->header.seq <<
std::endl;
   std::cout << "    time stamp: " << ptr->header.stamp <<
std::endl;
   std::cout << "    string frame_id: " << ptr-
>header.frame_id << std::endl;
   std::cout << "string child_frame_id: " << ptr-
>child_frame_id << std::endl;
   std::cout << "geometry_msgs/PoseWithCovariance pose" <<
std::endl;
   std::cout << "    geometry_msgs/Pose pose" << std::endl;
   std::cout << "        geometry_msgs/Point position" <<
std::endl;
   std::cout << "        float64 x: " << ptr-
>pose.pose.position.x << std::endl;
   std::cout << "        float64 y: " << ptr-
>pose.pose.position.y << std::endl;
   std::cout << "        float64 z: " << ptr-
>pose.pose.position.z << std::endl;
   std::cout << "    geometry_msgs/Quaternion orientation"
<< std::endl;
   std::cout << "        float64 x: " << ptr-
```

```
>pose.pose.orientation.x << std::endl;
   std::cout << "        float64 y: " << ptr-
>pose.pose.orientation.y << std::endl;
   std::cout << "        float64 z: " << ptr-
>pose.pose.orientation.z << std::endl;
   std::cout << "        float64 w: " << ptr-
>pose.pose.orientation.w << std::endl;
   std::cout << "    float64[36] covariance" << std::endl;
   std::cout << "        " << std::endl;
   for (int i = 0; i < 36; ++i) {
      std::cout << ptr->pose.covariance[i] << " ";
   }
   std::cout << std::endl;
   std::cout << "geometry_msgs/TwistWithCovariance twist"
<< std::endl;
   std::cout << "    geometry_msgs/Twist twist" <<
std::endl;
   std::cout << "        geometry_msgs/Vector3 linear" <<
std::endl;
   std::cout << "            float64 x: " << ptr-
>twist.twist.linear.x << std::endl;
   std::cout << "            float64 y: " << ptr-
>twist.twist.linear.y << std::endl;
   std::cout << "            float64 z: " << ptr-
>twist.twist.linear.z << std::endl;
   std::cout << "        geometry_msgs/Vector3 angular" <<
std::endl;
   std::cout << "            float64 x: " << ptr-
>twist.twist.angular.x << std::endl;
   std::cout << "            float64 y: " << ptr-
>twist.twist.angular.y << std::endl;
   std::cout << "            float64 z: " << ptr-
>twist.twist.angular.z << std::endl;
   std::cout << "    float64[36] covariance" << std::endl;
   std::cout << "        " << std::endl;
   for (int i = 0; i < 36; ++i) {
       std::cout << ptr->twist.covariance[i] << " ";
   }
   std::cout << std::endl;
   std::cout << "-----------------------------------------
------------------" << std::endl;
   return ;
}
```

- 订阅里程计话题

```
int main(int argc, char** argv)
{
    ros::init(argc, argv, "show_odometry");
    ros::NodeHandle nodeHandle;
    ros::Subscriber subscriber =
nodeHandle.subscribe("/odom", 1000, callback);
```

```
    ros::spin();
    return 0;
}
```

- 运行show_odometry

```
mac@ubuntu:~/Desktop/ros_ws$ rosrun show_odometry show_odometry
std_msgs/Header header
    uint32 seq: 84
    time stamp: 1668162553.054674248
    string frame_id: odom
string child_frame_id: base_footprint
geometry_msgs/PoseWithCovariance pose
    geometry_msgs/Pose pose
        geometry_msgs/Point position
        float64 x: -0.000926305
        float64 y: -0.000976497
        float64 z: 0
    geometry_msgs/Quaternion orientation
        float64 x: 0
        float64 y: 0
        float64 z: -0.000753429
        float64 w: 1
    float64[36] covariance
```

二，用图形界面显示颜色相机和深度相机的数据（利用OpenCV库）

1. **安装OpenCV库**

```
mac@ubuntu:~$ sudo apt-get update
mac@ubuntu:~$ sudo apt-get upgrade
mac@ubuntu:~$ sudo apt-get install libopencv-dev
mac@ubuntu:~$ sudo apt-get install libopencv-contrib-dev
```

2. **显示颜色相机数据**

- 查看话题消息的类型

```
mac@ubuntu:~$ rostopic info /camera/color/image_raw
Type: sensor_msgs/Image

Publishers:
 * /play_16876142674744651 40 (http://ubuntu:33251/)

Subscribers: None
```

- 显示消息具体内容

```
mac@ubuntu:~$ rosmsg show sensor_msgs/Image
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
uint32 height
uint32 width
string encoding
uint8 is_bigendian
uint32 step
uint8[] data
```

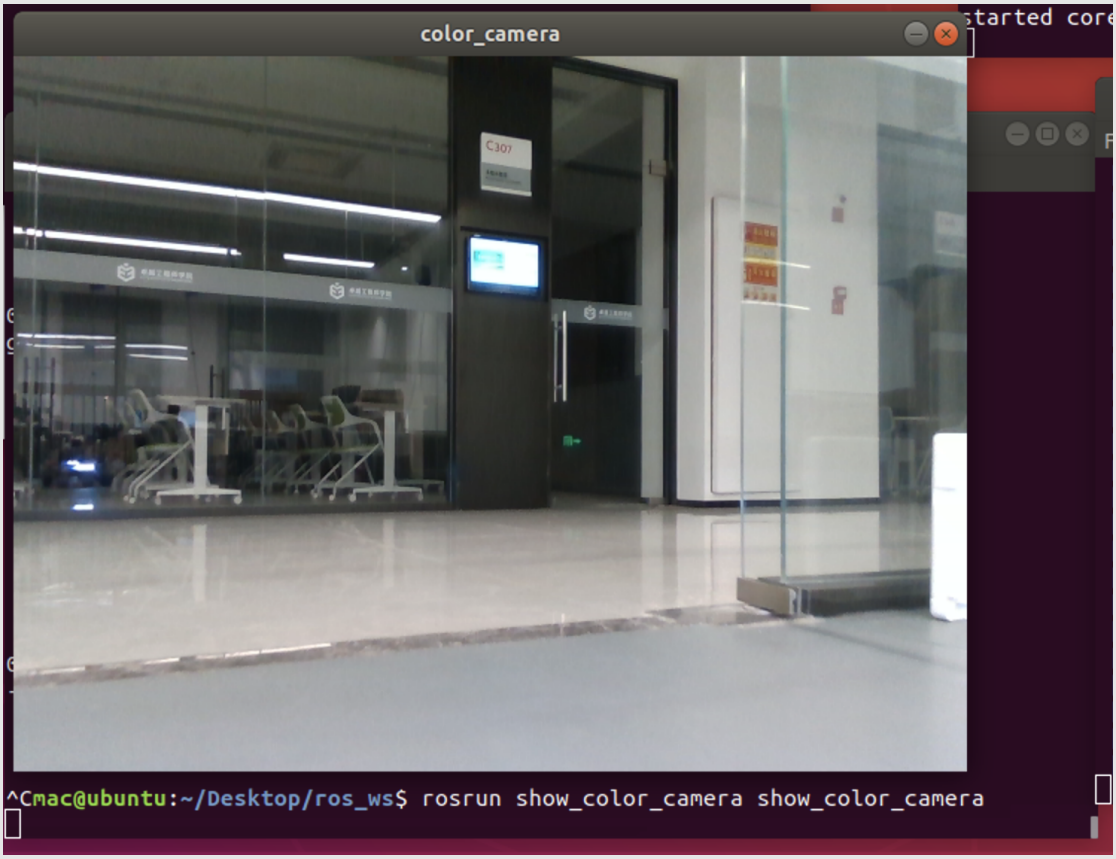
- 根据消息内容编写回调函数输出消息

```
void callback(const sensor_msgs::ImageConstPtr& ptr)
{
    cv_bridge::CvImagePtr cv_ptr;
    try {
        cv_ptr = cv_bridge::toCvCopy(ptr,
sensor_msgs::image_encodings::BGR8);
    } catch (cv_bridge::Exception& e) {
        ROS_ERROR("cv_bridge exception: %s", e.what());
        return ;
    }
    cv::imshow("color_camera", cv_ptr->image);
    cv::waitKey(1);
    return ;
}
```

- 订阅颜色相机话题

```
int main(int argc, char** argv)
{
    ros::init(argc, argv, "show_color_camera");
    ros::NodeHandle nodeHandle;
    image_transport::ImageTransport
imageTransport(nodeHandle);
    image_transport::Subscriber subscriber =
imageTransport.subscribe("/camera/color/image_raw", 1000,
callback);
    ros::spin();
    return 0;
}
```

- 运行show_color_camera

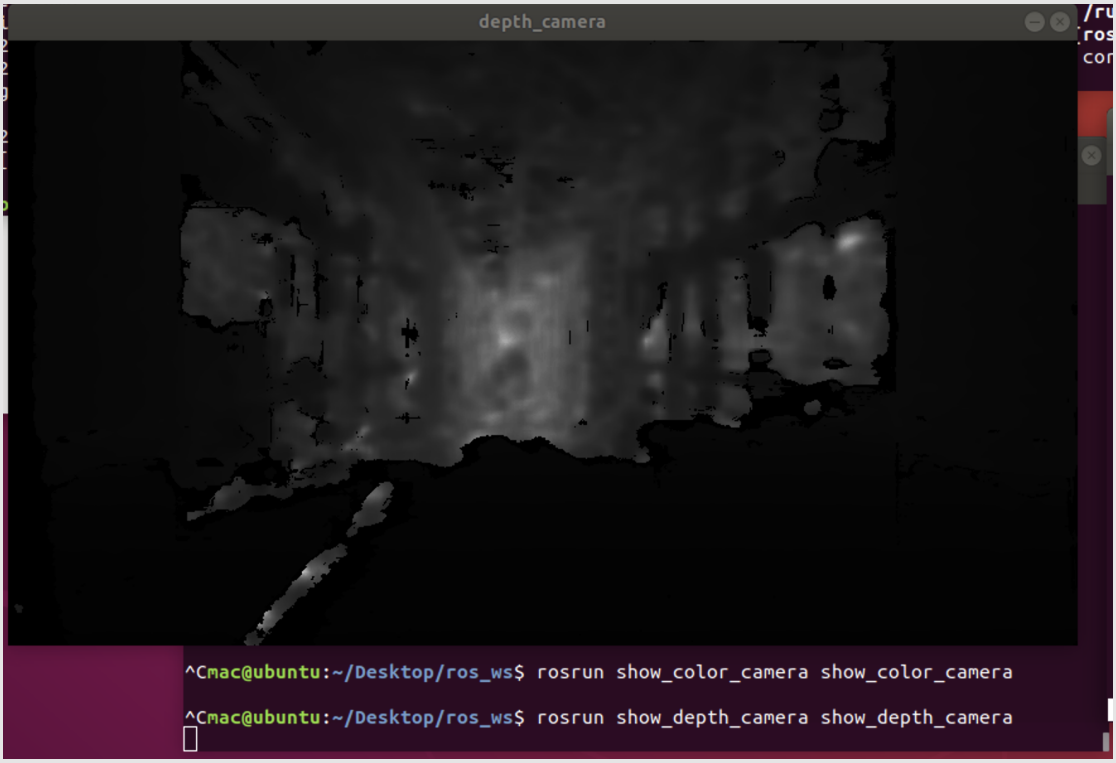


3. **显示深度相机数据**

显示深度相机数据的代码与显示颜色相机一样，只是图像编码不同而已。一个是BGR图像，一个是灰度图像。

```
cv_ptr = cv_bridge::toCvCopy(ptr,
sensor_msgs::image_encodings::TYPE_16UC1);
```

- 运行show_depth_camera

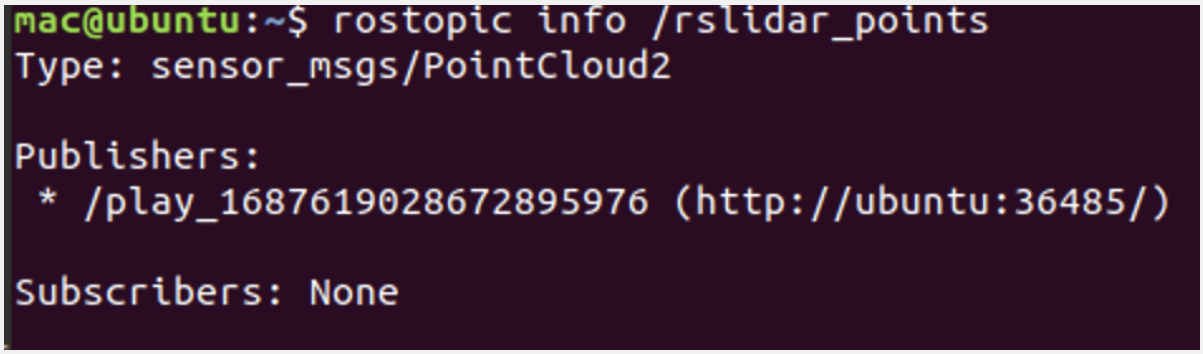


三，用图形界面显示激光雷达的点云数据（利用PCL库）

- 安装PCL库

```
mac@ubuntu:~$ sudo apt-get update
mac@ubuntu:~$ sudo apt-get upgrade
mac@ubuntu:~$ sudo apt-get install libpcl-dev
```

- 查看话题消息的类型

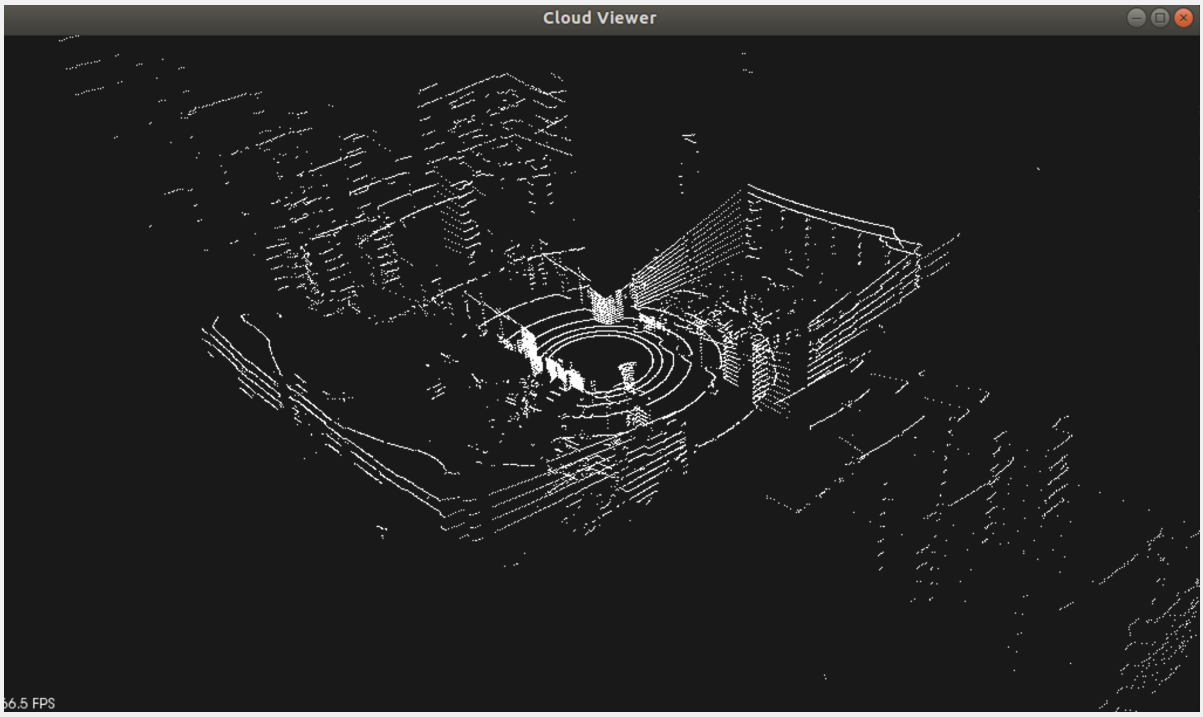


- 将ROS消息转换成点云显示出来

```
void callback(const sensor_msgs::PointCloud2::ConstPtr& msg)
{
    pcl::PointCloud<pcl::PointXYZ> cloud;
    pcl::PCLPointCloud2 pcl_pc;
    pcl_conversions::toPCL(*msg, pcl_pc);
    pcl::fromPCLPointCloud2(pcl_pc, cloud);
```

```
    viewer.showCloud(cloud.makeShared());
}
```

- 运行show_point_cloud



四，GMapping建图

- 安装gmapping

```
mac@ubuntu:~$ sudo apt-get update
mac@ubuntu:~$ sudo apt-get upgrade
mac@ubuntu:~$ sudo apt-get install ros-melodic-gmapping
```

- 查询nav_msgs/OccupancyGrid.msg消息定义

```
# This represents a 2-D grid map, in which each cell represents the
probability of
# occupancy.

Header header

#MetaData for the map
MapMetaData info

# The map data, in row-major order, starting with (0,0).  Occupancy
# probabilities are in the range [0,100].  Unknown is -1.
int8[] data
```

- 启动gmapping

```
void LaunchGMapping()
{
    system("rosrun gmapping slam_gmapping");
}
```

- 利用opencv绘制2D地图

```
void callback(const nav_msgs::OccupancyGrid::ConstPtr& ptr)
{
    std::cout << "resolution: " << ptr->info.resolution << std::endl;
    std::cout << "width: " << ptr->info.width << std::endl;
    std::cout << "height: " << ptr->info.height << std::endl;
    double scale = 1.0;
    int width = 1200;
    int height = 1200;
    cv::Point offset = {-1600, -1600};
    cv::Mat map = cv::Mat::zeros(cv::Size(width, height), CV_8UC3);
    for (int i = 0; i < ptr->info.width * ptr->info.height; ++i) {
        int x = (i % ptr->info.width + offset.x) * scale, y = (i /
ptr->info.width + offset.y) * scale;
        if (ptr->data[i] == -1) {
            cv::circle(map, cv::Point(x, y), 1, cv::Scalar(255, 255,
255), -1);
        } else if (ptr->data[i] >= 80) {
            cv::circle(map, cv::Point(x, y), 3, cv::Scalar(0, 0, 0),
-1);
        } else {
            cv::circle(map, cv::Point(x, y), 3, cv::Scalar(0, 255,
0), -1);
        }
    }
    cv::imshow("map", map);
    cv::waitKey(1000);
    return ;
}
```

- 建图结果