

# 自动驾驶 - 激光雷达避障 Project

---

## Project 要求

---

总体思路是编写一个 ROS Node，订阅激光雷达传感器数据，进行处理得到 2D 的障碍及自由区域。

### 第 1 档：对点云数据进行投影

点云包含了周围环境的三维信息，而自动驾驶一般仅关注无人车在 x-y 平面的运动，所以需要将 3D 点云先下采样和投影到 x-y 平面上，同时能够大大降低后续的计算量。

ROS 中点云的数据类型是 `sensor_msgs/PointCloud2`，主要有一个包含所有点的无序集合，在处理时，一般需要先把 msg 转换成 PCL 的 `pcl::PointCloud<pcl::PointXYZI>` 类型。

PCL (Point Cloud Library) 是针对点云数据的算法库，包含对点及点云完整的数据格式定义，及一些常见的算法封装。ROS 中使用 PCL，及点云类型的转换，可以参考这篇文章：

<http://www.cnblogs.com/li-yao7758258/p/6651326.html>

第一档要求将点云下采样和投影至 x-y 平面，以 `nav_msgs/OccupancyGrid` 消息类型进行输出，并能够在 ROS 的可视化工具 rviz 中显示。

请大家自行查阅了解该消息类型的具体定义。rviz 是 ROS 自带的可视化工具，具体的使用可以参考官方的教程。

### 第 2 档：障碍物检测

上一档的最终结果中，会包含点云中路面点投影的占用网格，这些路面上的点实际上是可通行的区域，划为障碍物会影响后续的路径规划。

这一档的输出类型与上一档一致，但需要大家预先对障碍物和自由区域进行检测，输出的栅格地图，最终应只包含自动驾驶行车时实际会造成阻碍的障碍物（显示为黑色网格）。

障碍物的检测需要考虑点云在空间表现出中的一些特征，有非常多的方法，具体的思路在课上有过简单的提示，大家可以多思考多调研，方法没有局限。

## 第 3 档：路径规划

在得到环境平面的空闲和障碍区域后，需要规划出一条路径，输出给下层控制器驱动无人车前行。

这一档需要大家实现能够在 rviz 通过 2D Pose Estimate 设置起点，2D Nav Goal 设置终点之后，计算出起点至终点之间的一条路径。计算出的路径应是没有触及栅格地图中的障碍区域的。

起点和终点的获取可以通过订阅 /initialpose 和 /move\_base\_simple/goal 话题的数据，消息类型分别为： geometry\_msgs/PoseWithCovarianceStamped ， geometry\_msgs/PoseStamped 。

路径规划的思路主要是基于一些图搜索的算法，将前面栅格图的自由区域当做一个有向联通图，使用最短路径算法计算路径，例如像 Dijkstra 和 A\* 这样的广度优先搜索算法。

## 数据包

给定的数据包包含了需要的一帧点云数据，用于 Project 的验证和测试。

数据包以 ROS Bag 形式提供，具体信息如下：

```
path:      point_cloud_projection.bag
version:    2.0
duration:   0.0s
start:      Apr 30 2018 23:47:48.46 (1525103268.46)
end:        Apr 30 2018 23:47:48.46 (1525103268.46)
size:       767.1 KB
messages:   1
compression: none [1/1 chunks]
types:      sensor_msgs/PointCloud2 [1158d486dd51d683ce2f1be655c3c181]
topics:     /point_cloud  1 msg      : sensor_msgs/PointCloud2
```

只需按照所列出的话题名和类型订阅传感器消息即可。

ROS Bag 的使用具体见教程：[Recording and playing back data](#)

**注意** bag 中只包含了一帧点云，大家需要先运行编写的节点，之后再播放数据包。