基于 PCL 和 ROS 的道路状况检测的项目总结

本项目主要是针对上汽公司提供的车辆行驶时所获取到的激光雷达点云数据进行处理，以检测道路上诸如车辆，行人，道路边沿等不同的目标。上汽公司那一边提供的技术参数主要如下：

技术指标	主要参数
车辆尺寸	5 * 2 * 2
速腾激光雷达安装位置	高度 1.2 米，距离左侧 0.75 米，下倾 13°

- 雷达点云数据主要以.bag 格式提供。
- 我们主要处理了两类数据，一类是城市道路，另一类是高架。
- 本项目用到的工具主要是 PCL 和 ROS，关于这两个工具的具体使用方法将在另外一份文档中进行说明。
- 本项目的主要流程包括四个方面：第一步是滤波；第二步是分割；第三步是目标框定；第四步是重构。
- 所有代码可见压缩包。

下面我们将对每一步所用到的算法，算法特点以及具体参数进行详细的说明。

一、 滤波

本项目滤波主要采用了条件滤除（ConditionalRemoval）算法。

算法特点：它可以一次删除满足对输入的点云设定的一个或者多个条件指标的所有数据点。

算法原理：算法原理很简单，主要根据具体的数据特征（比如说 X,Y,Z 的坐标值），我们设定条件，将不符合条件的点滤除。

算法参数：主要以下表形式给出。

坐标	范围
x	[-7.5,7.5]
y	[-10,5]
z	[-0.2, ∞]

二、 分割

本项目分割主要采用了基于法线的区域生长（RegionGrowing）算法。

算法特点：区域生长算法直观感觉上和欧几里德算法相差不大，本质上都是用区分邻里关系远近来完成的。都是从一个点出发，最终占领整个被分割区域。

一般欧几里德算法是使用距离作为判定依据。对于点云数据来说，因为点云数据提供了更高维度的数据，故有很多信息可以提取获得。区域生长算法则利用了法线，曲率，颜色等信息来判断点云是否应该聚成一类。该算法是针对小曲率变化面设计的，尤其适合对连续阶梯平面进行分割。

算法原理：首先依据点的曲率值对点进行排序，之所以排序是因为，区域生长算法是从曲率最小的点开始生长的，这个点就是初始种子点，初始种子点所在的区域即为最平滑的区域，从最平滑的区域开始生长可以减少分割片段的总数，提高效率，设置一空的种子点序列和空的聚类区域，选好初始种子后，将其加入到种子点序列中，并搜索邻域点，对每一个邻域点，比较邻域点的法线与当前种子点的法线之间的夹角，小于平滑阀值的将当前点加入到当前区域，然后检测每一个邻域点的曲率值，小于曲率阀值的加入到种子点序列中，删除当前的种子点，循环执行以上步骤，直到种子序列为空。其算法可以总结为：

1. 种子周围的临近点和种子点云相比较
2. 法线的方向是否足够相近
3. 曲率是否足够小
4. 如果满足 2，3 则该点可用做种子点
5. 如果只满足 2，则归类而不做种
6. 从某个种子出发，其“子种子”不再出现则一类聚集完成
7. 类的规模既不能太大也不能太小

算法参数：

主要参数	参数值
最小的聚类点数	10
最大的聚类点数	10000
搜索方式	KdTree
搜索的领域点的个数	200
区域生长方式	基于法线
平滑度	3.0/180.0*M_PI
曲率阈值	1.0
计算法线时的搜索方式	KdTree
计算法线时搜索半径	1.0

三、 目标框定

本项目的目标框定算法主要是对分割步骤得到的各个 cluster 用长方体进行框出，然后实时显示。

算法特点：算法的思想比较简单，易于编程实现，而且能够做到实时显示。

算法原理：针对分割出来的每一个类，找出 x，y，z 三个方向上的最小最大值（x_{min}，x_{max}，y_{min}，y_{max}，z_{min}，z_{max}），然后根据立体几何知识，画出长方体的 8 条边。显示时对每个类用不同颜色标识，由于每一幅图上的类的个数和类型是不断发生变化的，所以我们在选择颜色的时候是采用随机生成颜色的方式，但是确保能在每一幅图上对不同的类用不同的颜色予以标识。

算法参数：

主要参数	参数值
长方体每条边 x 方向的采样间隔	0.1
长方体每条边 y 方向的采样间隔	0.1
长方体每条边 z 方向的采样间隔	0.1

四、 重构

PCL 中实现曲面重构的算法主要有三个：泊松（Possion）重构算法，贪婪三角形投影算法和移动立方体算法。本项目主要采用了泊松重构算法。

算法特点：Possion 重构的输入是点云及其法向量，输出是三维网格。Possion 重构是一个非常直观的方法，Possion 重构主要有以下几个特点：

- Poisson 在边缘处的锐度比 VRIP 要好。这是因为 VRIP 在大的边缘处 TSDF 的累加会有平滑效应，而 Poisson 依据的是法向量，不会引入额外的平滑。
- VRIP 是局部方法，每次只更新当前深度图对应的 TSDF。Poisson 是全局方法。
- Poisson 对于噪声更加鲁棒一些。点云法向量估计的精度不能太差。
- 如果重建出奇怪的形状（分层、分块），请查看原始点云是否平滑，是否有噪声，调整生成网格的分辨率以适应点云。

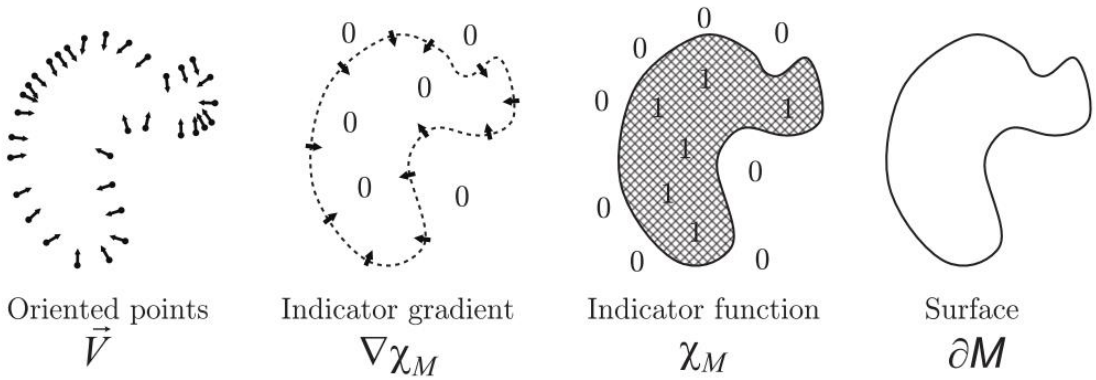
算法原理：

Possion 算法的核心思想是点云代表了物体表面的位置，其法向量代表了内外的方向。通过隐式地拟合一个由物体派生的指示函数，可以给出一个平滑的物体表面的估计。

给定一个区域 M 及其边界 ∂M，指示函数 χ_M 定义为：

$$\chi_M(x)=\begin{cases}1 & x\in M\\ 0 & x\in M\end{cases}$$

这样，把重构 $S=\partial M$ 的问题转换为重构 χ_M 的问题。下图非常形象地描述了这二者的联系。



具体算法的数学推导很多，在此不做详细的说明。

采用 Possion 重构的一般过程：

1. 定义八叉树。使用八叉树结构存储点集，根据采样点集的位置定义八叉树，然后细分八叉树使每个采样点都落在深度为 D 的叶节点；
2. 设置函数空间：对八叉树的每个节点设置空间函数 F，所有节点函数 F 的线性和可以表示向量场 V，基函数 F 采用了盒滤波的 n 维卷积；
3. 创建向量场：均匀采样的情况下，假设划分的块是常量，通过向量场 V 逼近指示函数的梯度。采用三次条样插值（三线插值）；
4. 求解泊松方程：方程的解采用拉普拉斯矩阵迭代求出；
5. 提取等值面：为得到重构表面，需要选择阈值获得等值面；先估计采样点的位置，然后用其平均值进行等值面提取，然后用移动立方体算法得到等值面。

算法参数：

主要参数	参数值
pn.setConfidence(false)	false（如果 false，所有法向量均归一化）
pn.setDegree(2)	2 设置参数 degree[1,5],值越大越精细，耗时越久
pn.setDepth(8)	树的最大深度，求解 2^d x 2^d x 2^d 立方体元。由于八叉树自适应采样密度，指定值仅为最大深度。
pn.setIsoDivide(8)	用于提取 ISO 等值面的算法的深度
pn.setManifold(false)	是否添加多边形的重心，当多边形三角化时。设置流行标志，如果设置为 true，则对多边形进行细分三角话时添加重心，设置 false 则不添加

pn.setOutputPolygons(false)	是否输出多边形网格（而不是三角化移动立方体的结果）
pn.setSamplesPerNode(3.0)	设置落入一个八叉树结点中的样本点的最小数量。无噪声，[1.0-5.0],有噪声[15.-20.]平滑
pn.setScale(1.25)	设置用于重构的立方体直径和样本边界立方体直径的比率。
pn.setSolverDivide(8)	设置求解线性方程组的Gauss-Seidel迭代方法的深度
pn.setSearchMethod	设置搜索方法，我们采用的是Kdtree