基于Voronoi Diagram的车道中心线生成

1. 问题

传统的车道检测方法通常在光线不足、遮挡和车道磨损等各种条件下难以实现，导致结果不可靠、断断续续。通过将Voronoi Diagram应用于已检测到的车道线精准生成车道中心线。主要目标是实现可靠的数据预处理与后处理，应用Voronoi Diagram进行准确的中心线计算，优化算法以确保实时计算的效率。

1. 实现
   1. DBSCAN Clustering

聚类算法，用于将乱序的数据点分类为不同的线。

* 1. merge\_line\_on\_side() – 可选

将最靠近左上角、左下角，右上角、右下角的线默认是一条线但因为检测的问题断了，将其分别合并。

* 1. 车道线平滑 – 可选

使用B-Spline Interpolation对车道线进行平滑处理，能解决一些数据点存在杂点的问题。

* + 1. Moving Average – 可选

可解决一些少数数据点噪音的情况。

* 1. Alpha Shape

找出包围整个图的多边形，用于清除Voronoi Diagram在车道线外围生成的杂点。

* 1. Voronoi Diagram Generate

使用jcv\_voronoi库进行生成，使用了[Fortune's sweep algorithm.](https://en.wikipedia.org/wiki/Fortune%27s_algorithm)其运行速度相较于网上各种开源库更快且使用简单可靠占用内存少。<https://github.com/JCash/voronoi>

* 1. 后处理

Voronoi Diagram生成出来的结果是很多条小线段，其中很多是杂点。因此需要对其进行后处理以删除多余部分。

根据Voronoi Diagram的理论。只需要通过寻找不在车道线点的外包围多边形的线段，我们就可以找出在车道外需要删掉的部分。然后寻找与每一根车道线相交的线段删掉。剩下的部分便是干净的中心线了（如果初始数据很杂乱且车道线不进行平滑处理的话还是会留下一些小杂点）。

基于这个需求我选择了r-tree数据结构，r-tree是空间数据索引的方法，可以快速的基于各个对象的空间关系进行查找(query)。

1. 优化1

从r-tree中查找出来的结果回分为两个vector，一个是在外包围多边形内的index（过滤掉车道外的线段后要保留的），一个是与车道线相交的index（要删掉的）。这里的index指的是包含voronoi diagram中生成的所有小线段的vector的index。所以这里需要找出两个vector都纯在的index然后删掉。

一个个进行遍历或查找十分慢，所以我想出了一个更优雅的方法来完成这个计算，也就是创建两个主vector长度的binary number，将这些index按照index存进去，0表示删掉，1表示保留。然后只需要让这两个binary number通过一个AND gate就能得出最后要保留的线段的index。

1. 优化2

当车道线不够平滑时，生成的中心线即便经过后处理还是会有一些杂线，这个时候就需要通过将离车道线太近的中心线剔除，这样可以尽可能的使生成的中心线干净，而且保证了可靠性，并且可以顺便使我们知道某条道是否过窄无法通过。

Pipeline：

A screenshot of a phone

Description automatically generated