



基于点云的车辆多目标跟踪算法研究

院系：广州研究院 专业：通信工程

汇报人：刘智文



目录

CONTENTS

- 01 选题背景与意义
- 02 研究内容与技术路线
- 03 目标检测算法研究
- 04 目标跟踪算法研究
- 05 总结与展望

01
PART

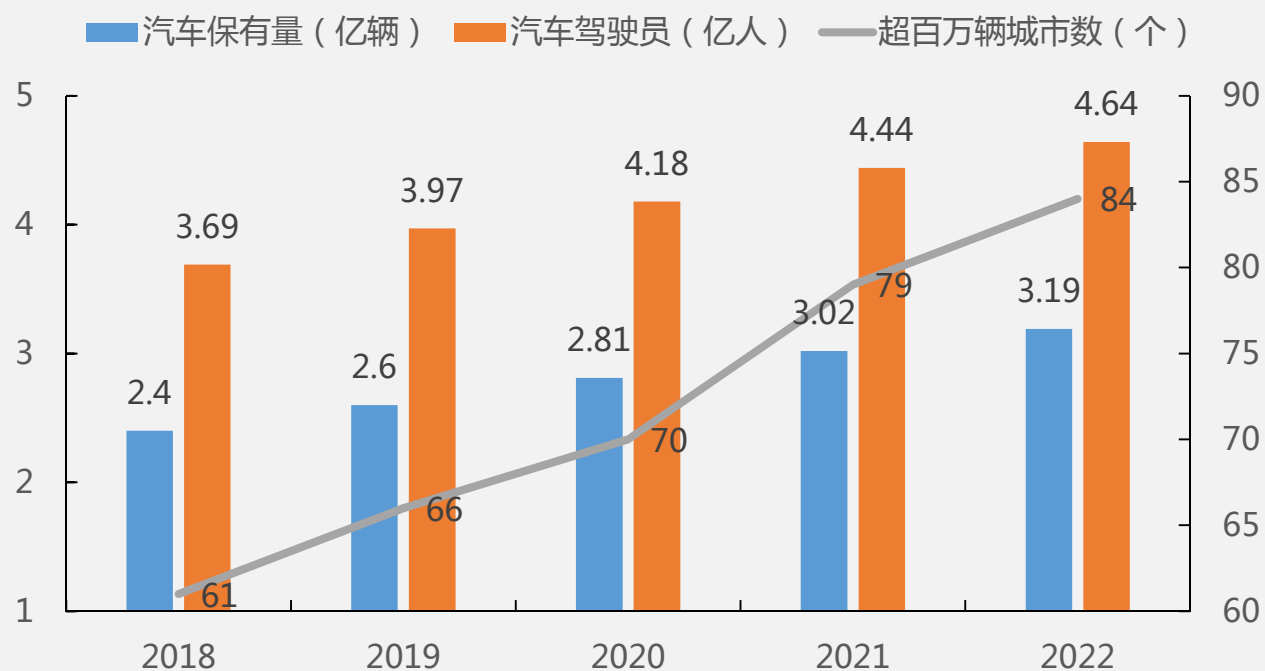
选题背景与意义



选题背景

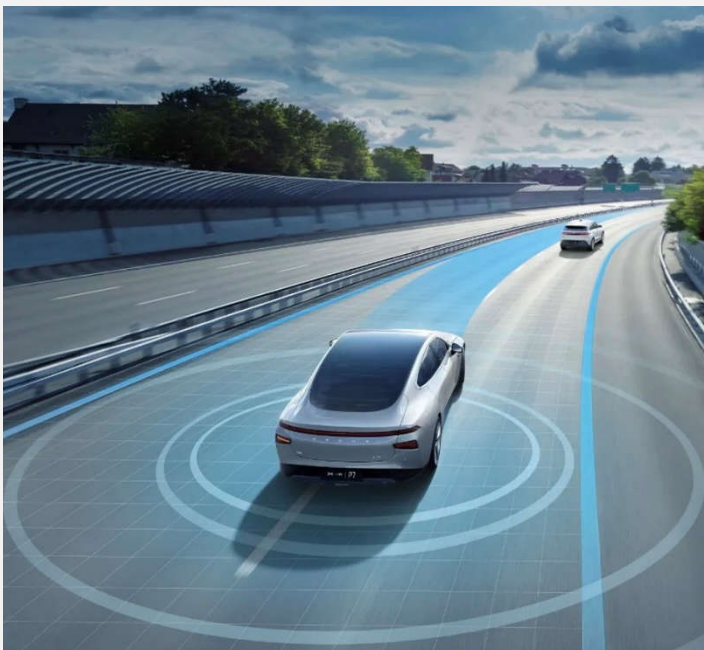


近五年全国汽车保有量统计





选题背景



环境感知



决策规划



执行控制



选题意义



目标跟踪算法

目标跟踪作为环境感知的重要组成部分，不仅能加速推进自动驾驶技术的实现与落地，有效缓解由于驾驶员技术或者疲劳引发的巨大安全隐患，还能解决目标检测中由于漏检导致的不连续的问题。

单目标跟踪和多目标跟踪

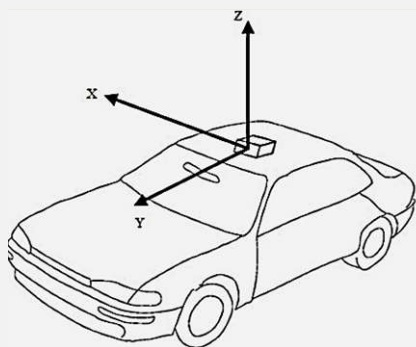
目标跟踪按研究对象可分为单目标跟踪和多目标跟踪，按研究方法可分为跟踪前检测和检测前跟踪。

02
PART

研究内容与技术路线



数据采集



激光雷达产品参数表

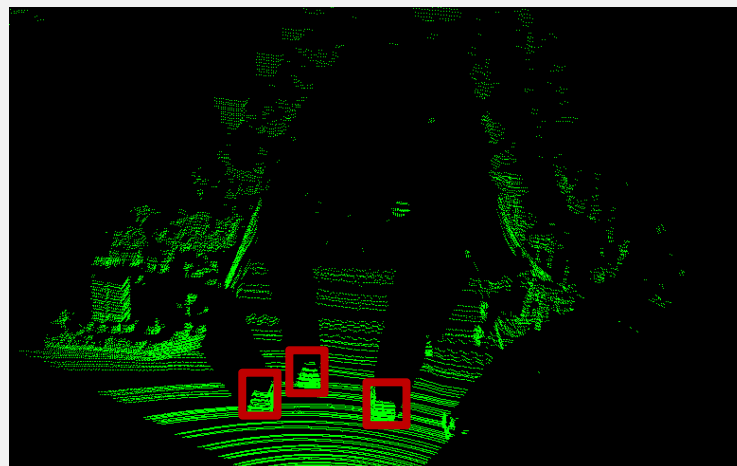
型号		CH128X1
扫描通道		128 路
测量方式		脉冲式
激光波长		905nm
测量精度		±3cm
测点速率		82 万点/秒
扫描频率		5 ~ 25Hz
视场角	垂直	-18° ~ 7°
	水平	120°
角度分辨率	垂直	0.125°
	水平	10Hz: 0.2° (20Hz: 0.4°)
尺寸 (长 * 宽 * 高)		118 * 90 * 75mm



研究对象



相机图像



点云图像

数据格式的不同

$$\begin{bmatrix} f_{11} & f_{12} & \dots & f_{1n} \\ f_{21} & f_{22} & \dots & f_{2n} \\ \dots & \dots & \dots & \dots \\ f_{m1} & f_{m2} & \dots & f_{mn} \end{bmatrix} \neq \begin{bmatrix} f_{21} & f_{22} & \dots & f_{2n} \\ f_{11} & f_{12} & \dots & f_{1n} \\ \dots & \dots & \dots & \dots \\ f_{k1} & f_{k2} & \dots & f_{kn} \end{bmatrix}$$

跟踪方法的不同

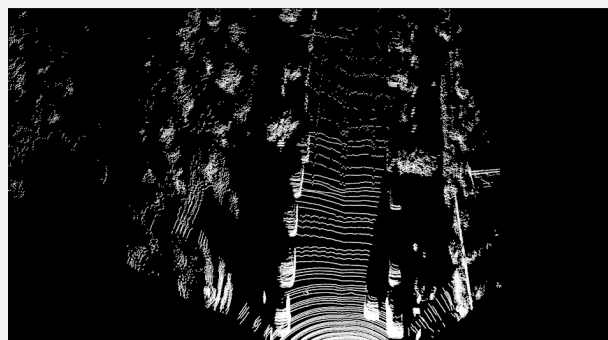
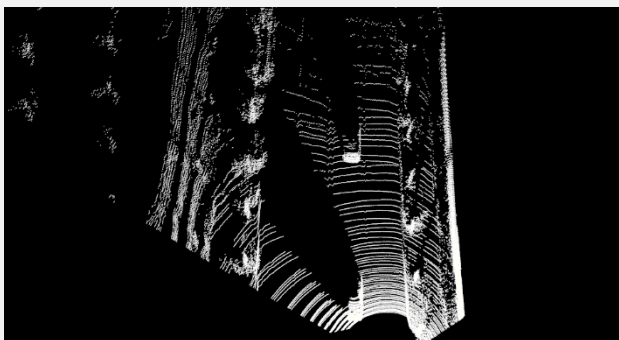
$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix} = \begin{bmatrix} x_2 & y_2 & z_2 \\ x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_k & y_k & z_k \end{bmatrix}$$



研究对象



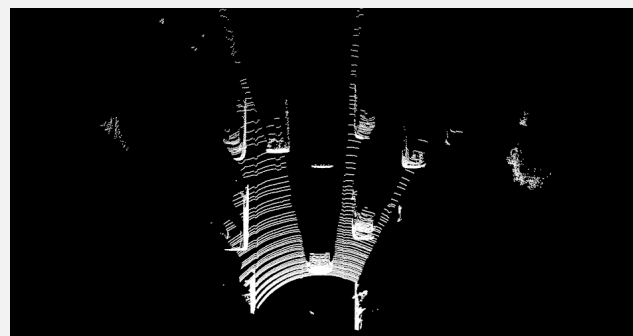
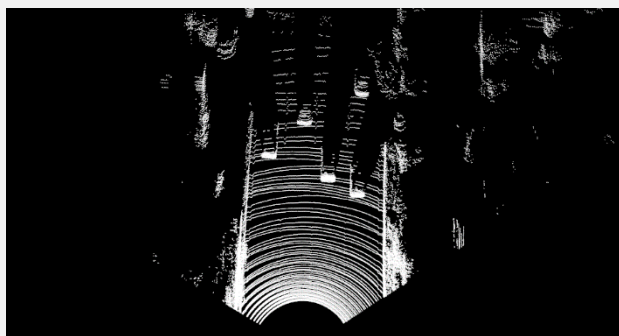
研究对象为在道路上正常行驶的车辆。采集的数据中包括在车辆行驶过程中遇到的不同场景。



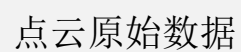
上图展示了车辆少以及路旁停放车辆密集的情况。



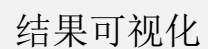
研究对象



上图展示了车辆较多以及红绿灯口车辆静止时的情况。



目标检测模块



目标跟踪模块



现存问题

一、聚类算法耗时长

点云数据中点的个数多，且点云聚类算法常通过自定义阈值进行迭代计算，耗时严重。

三、运动状况难描述

在道路上行驶的车辆，常伴随着改变车道，改变速度的情况发生。在对车辆速度进行估计时，难以用一种确定的运动模型描述目标运动状况。

二、姿态估计不准确

对三维目标的跟踪需要更多的状态量，指明目标运动方向。对受遮挡，点云数量少，点云不完整目标的姿态估计与实际存在偏差。

四、数据关联情况多

在多目标跟踪中，需要确定量测来自杂波还是某条轨迹。当量测和轨迹数量多时，关联情况多样，导致计算复杂，甚至出现组合爆炸，出现切换错误。





主要工作

数据采集和格式转换

通过车载激光雷达采集点云数据；熟悉其他数据集中数据格式以及之间的格式转换。

目标检测模块

尽可能彻底滤除地面，缩短聚类时间，能检测到被遮挡和点云数量少的目标。

跟踪系统搭建

使用不同的运动模型进行建模，结合数据关联和匈牙利算法，避免组合爆炸，添加相似度度量方法，解决使用交并比导致的小目标和速度快目标无法跟踪的问题。

结果可视化

直接将跟踪结果利用点云库中的可视化工具显示，方便在不同的视角观察跟踪结果。

03
PART

目标检测算法研究



目标检测



目标检测

地面滤除

去除地面点，减少对聚
类算法和检测目标高度的
影响。

点云聚类

初步确定检测目标

姿态估计

设置过滤规则，确定检测目标和
航向角。返回目标位置信息方便后续
对目标的跟踪。

目标跟踪

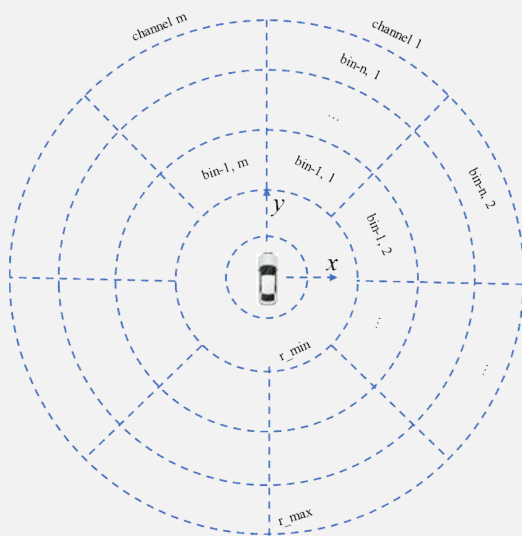
.....



地面滤除

基于坡度的地面分割算法

直接处理点云原始数据，将点云数据分成不同的栅格，从点云数据中的高度信息出发，结合坡度和传感器本身高度，滤除地面。



计算坡度前进行高斯滤波

线性滤波，对栅格中的高度信息进行加权平均，去除数据中的噪点，在保证去噪质量的前提下保留住点云数据特征信息。

计算坡度后进行均值滤波

非线性滤波，对栅格地面点周围的非地面点进行平滑，取中值代替原始值，可以尽可能的找到地面点，同时能很好的保护数据边缘特征信息。





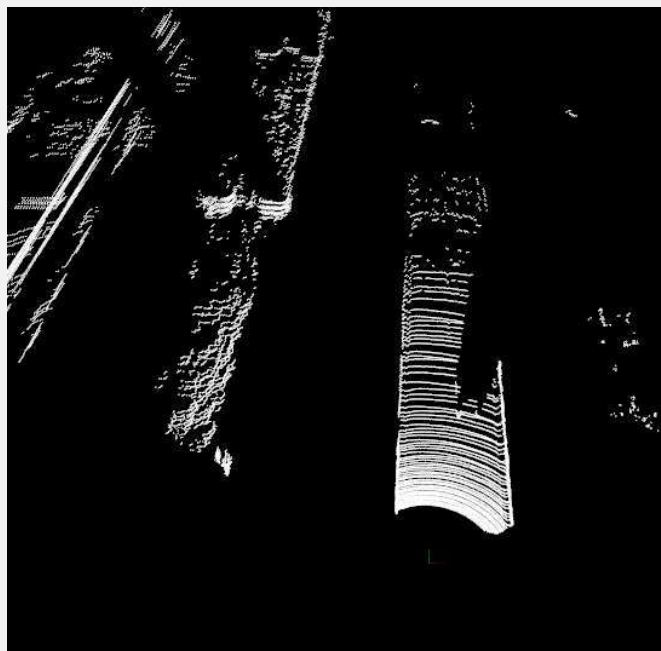
地面滤除



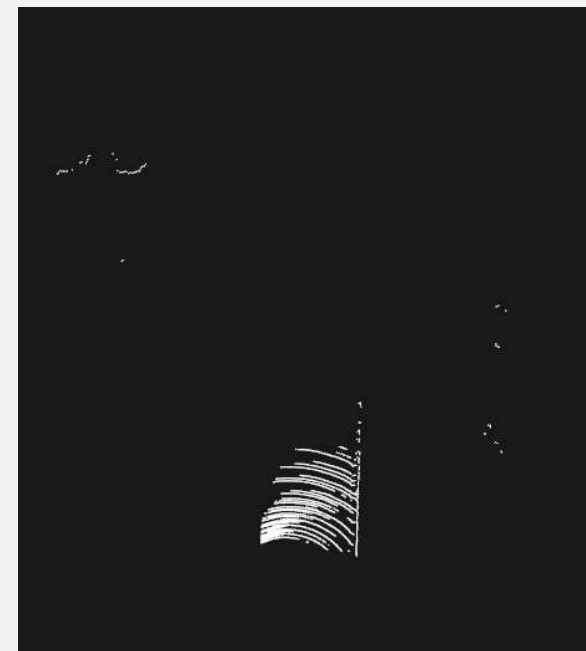
地面滤除效果



原始点云



本文算法



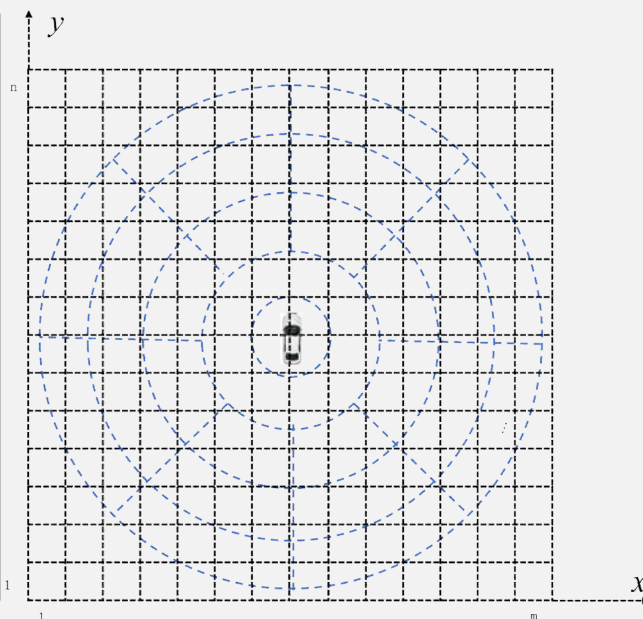
随机抽样一致性平面拟合算法



点云聚类

点云聚类

考虑到点云目标在聚类时不受高度的限制，可以直接将点云数据投影在平面上，依据扫描范围，划分不同网格区域。以网格中的点云数量为判断条件，确定类别在二维平面投影后的区域。



改进方法

在车辆密集的情况下，网格划分大小会导致欠分割，选取更小的网格大小可以一定程度上避免欠分割，但是会增大计算量。且网格大小不是越小越好。

先采取大网格进行聚类，再对其中过大的类别结合更小的网格和网格内点云数量聚类。

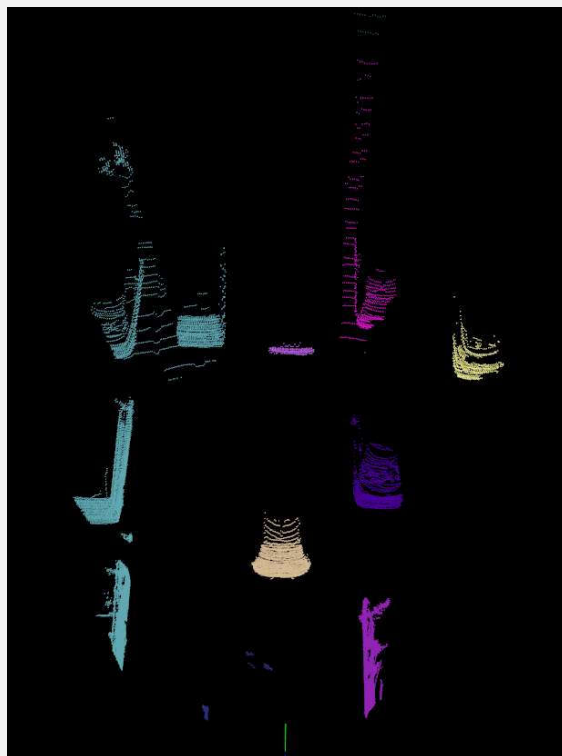


欠分割

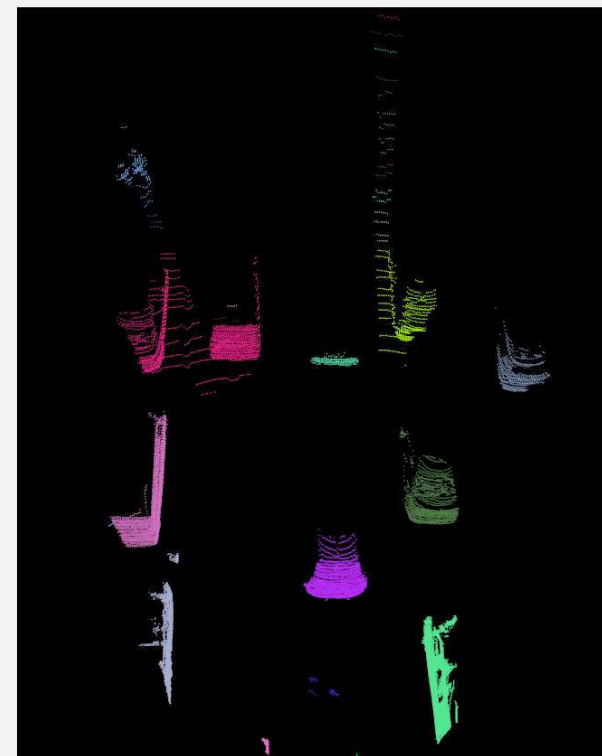
红绿灯前车辆密集情况



原图



欠分割



改进后

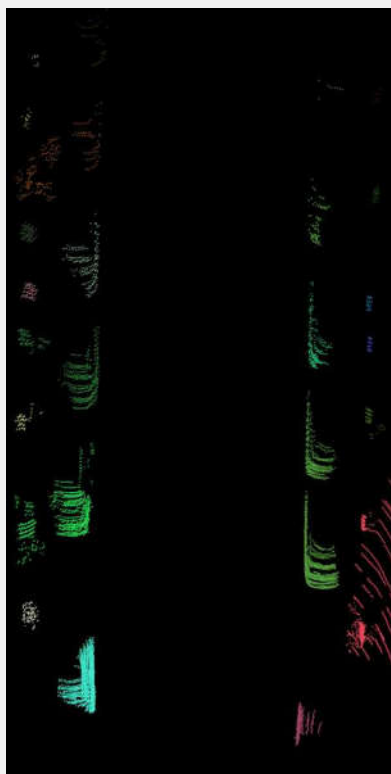


欠分割

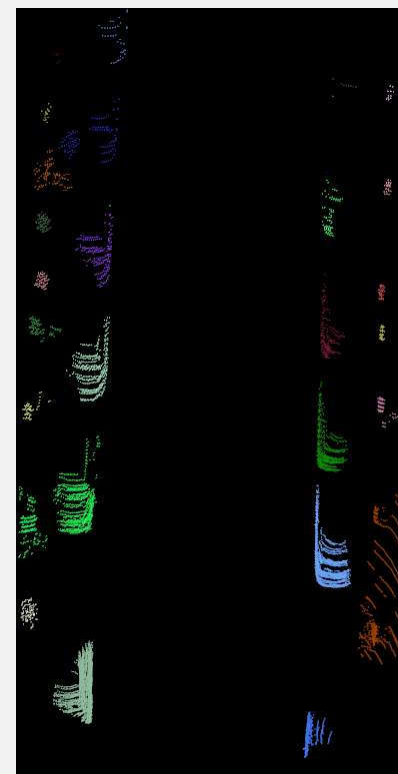
道路两边车辆停放距离近



原图



欠分割



改进后



点云聚类



聚类算法耗时对比

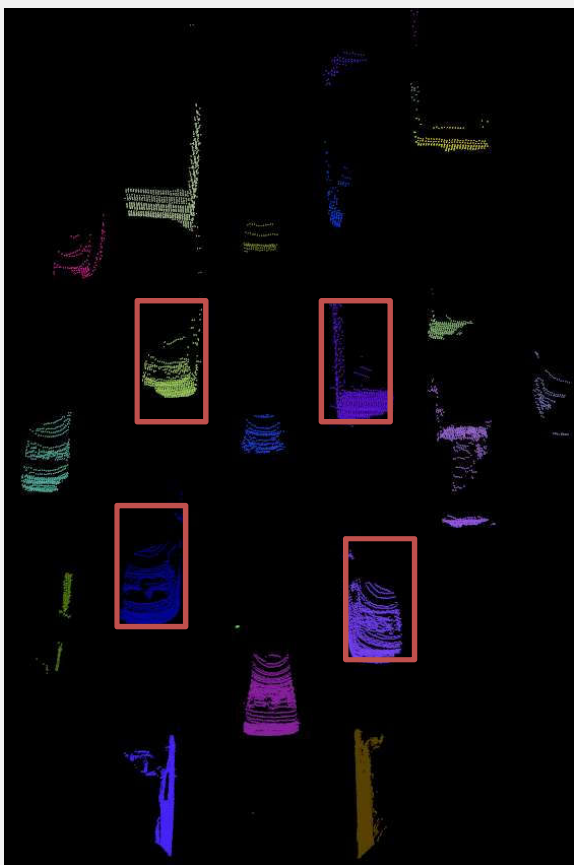
数据	算法	
	连通组件聚类	KD-Tree加速的DBSCAN聚类
实地采集/1000帧	3.9235s	36.4319s
KITTI数据/1000帧	9.6262s	97.7457s

对比分析

利用连通组件聚类，在聚类时不考虑高度信息，不用进行迭代计算，耗时比采用基于密度聚类算法缩短十倍左右。而KITTI数据整体比采集到的数据耗时的原因在于KITTI数据是由机械式激光雷达采集的，水平视角范围为 360° ，实地采集的数据是由混合固态激光雷达采集的，水平视角范围为 120° 。



姿态估计



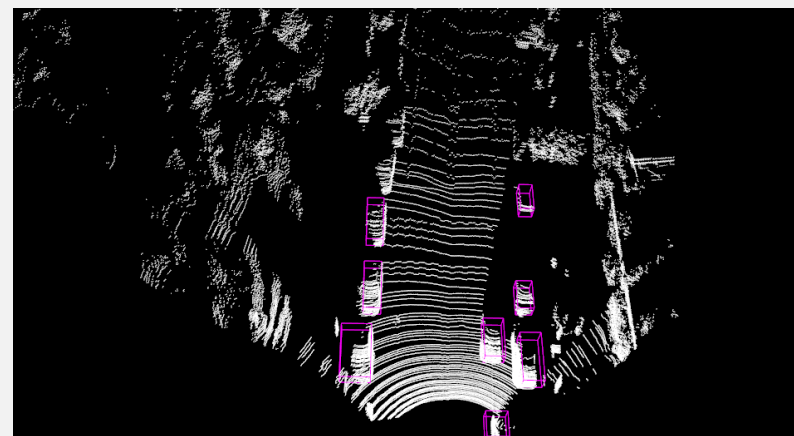
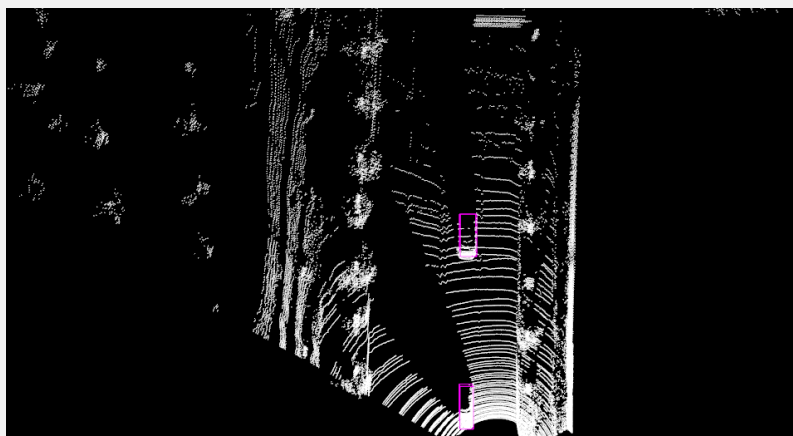
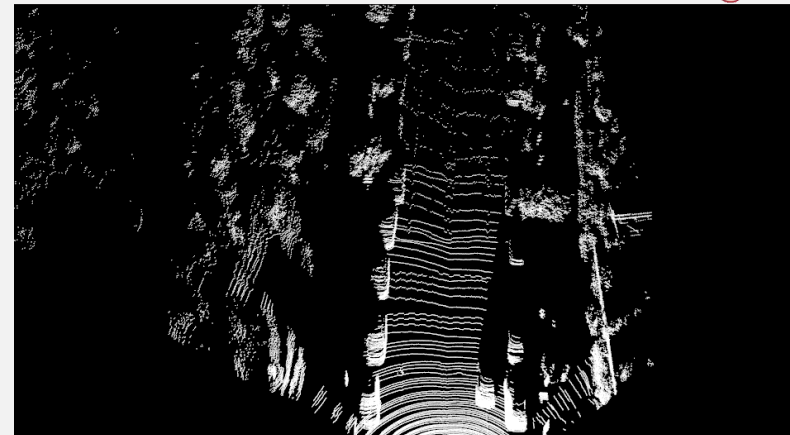
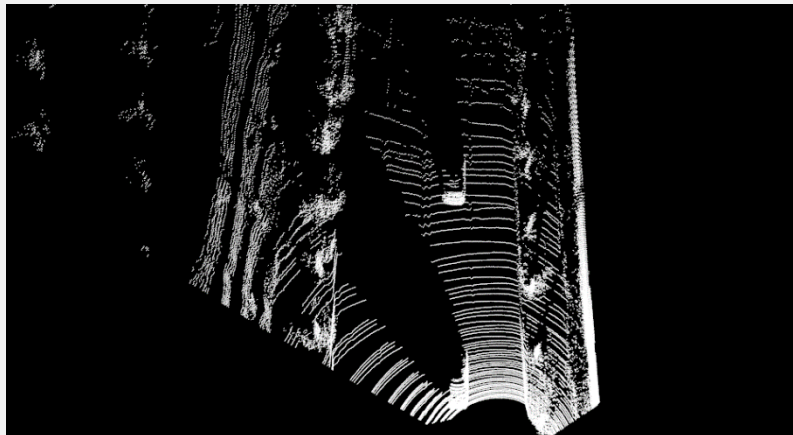
场景分析

对于两侧的车辆，可以得到后面和侧边的点云，使用L型算法可以很好的估计航向角。对于正前方，以及远处的目标，只能获得后方的部分点云。L型算法失效。

为了解决正前方以及远处的目标姿态估计不准的问题，首先将聚类目标通过坐标转换转到坐标系中间位置，分别计算聚类中各点的斜率，求出最值。通过最值绝对值差值判断目标是否位于正前方，并求出目标在各轴上的最值，确定其位置信息。

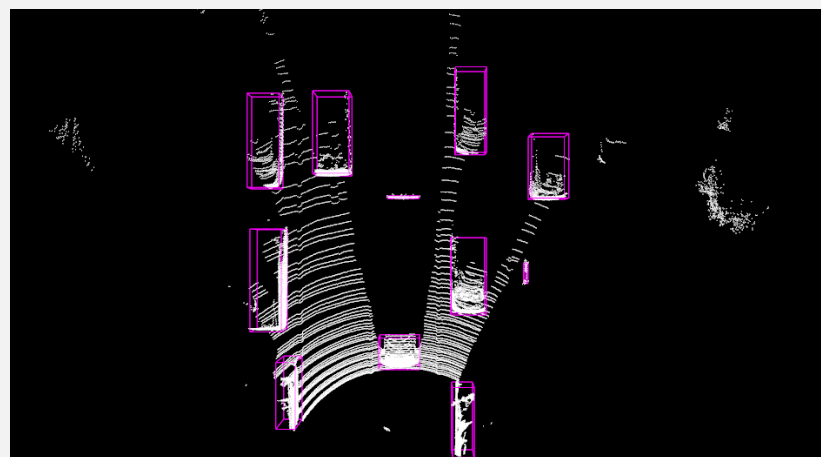
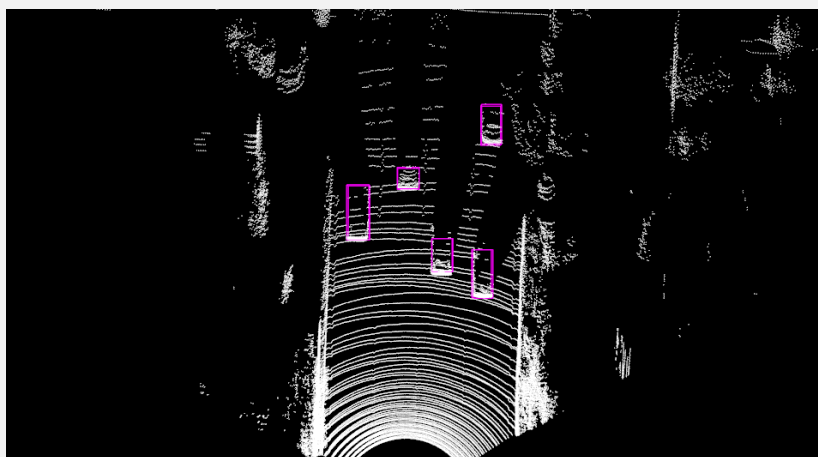
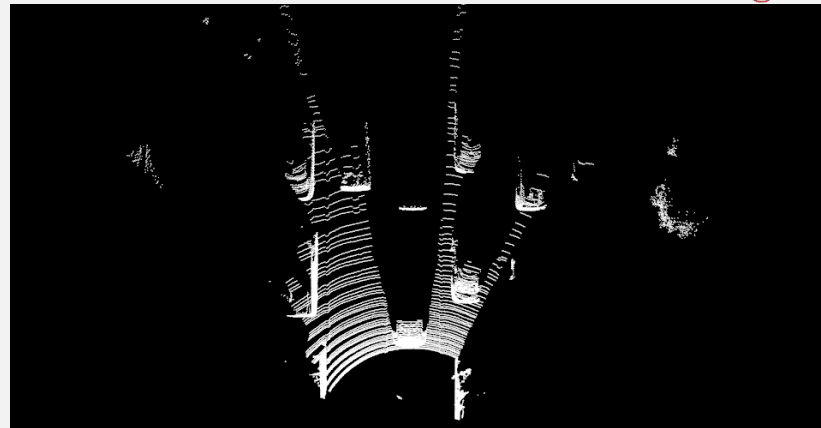


检测结果





检测结果



04
PART

目标跟踪算法研究



目标跟踪



目标检测

.....

目标跟踪

运动建模

采用交互式多模型算法，使用匀速，
恒定转弯率和速度，恒定转弯率和加速度
模型描述目标运动情况。

滤波更新

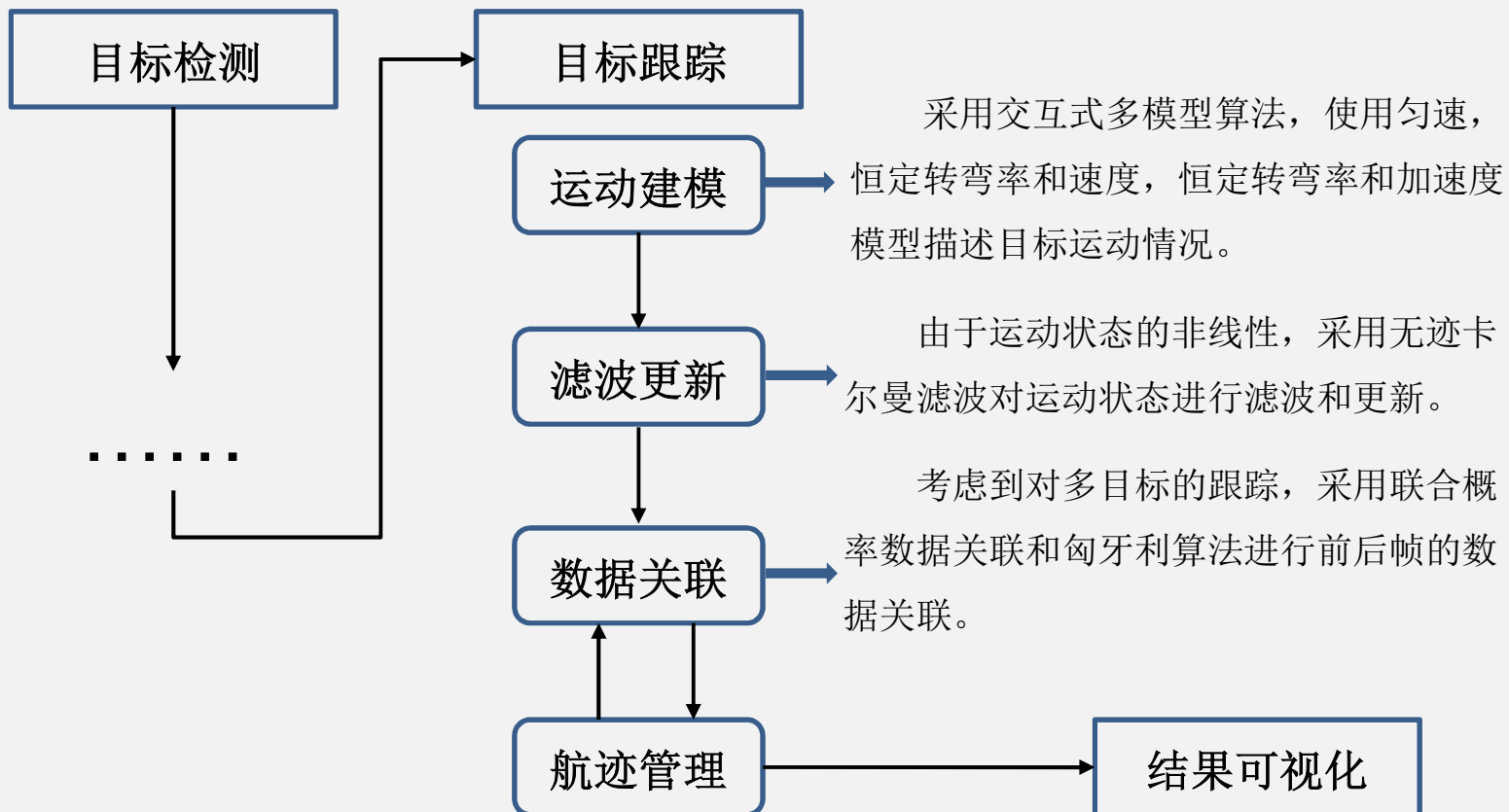
由于运动状态的非线性，采用无迹卡
尔曼滤波对运动状态进行滤波和更新。

数据关联

考虑到对多目标的跟踪，采用联合概
率数据关联和匈牙利算法进行前后帧的数
据关联。

航迹管理

结果可视化





运动建模

CV运动模型

$$x_{k+1} = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix}_k$$

CTRV运动模型

$$x_{k+1} = x_k + \begin{bmatrix} \frac{v_k}{\omega_k} (\sin(\theta_k + \omega_k \Delta t) - \sin(\theta_k)) \\ \frac{v_k}{\omega_k} (-\cos(\theta_k + \omega_k \Delta t) + \cos(\theta_k)) \\ 0 \\ \omega_k \Delta t \\ 0 \end{bmatrix}$$

$$\begin{cases} p_{x_{k+1}} = p_{x_k} + v_k \cos(\theta_k) \Delta t \\ p_{y_{k+1}} = p_{y_k} + v_k \sin(\theta_k) \Delta t \end{cases}$$

CTRA运动模型

$$x_{k+1} = x_k + \begin{bmatrix} g_x(x_k, \Delta t) \\ g_y(x_k, \Delta t) \\ \omega \Delta t \\ a \Delta t \\ 0 \end{bmatrix}$$

$$\begin{cases} p_{x_{k+1}} = p_{x_k} + (v_k \Delta t + \frac{1}{2} a_k (\Delta t)^2) \cos(\theta_k) \\ p_{y_{k+1}} = p_{y_k} + (v_k \Delta t + \frac{1}{2} a_k (\Delta t)^2) \sin(\theta_k) \end{cases}$$

$$g_x = \frac{a}{\omega^2} (\cos(\theta_k + \omega \Delta t) - \cos(\theta_k)) + \frac{1}{\omega} ((v_k + a \Delta t) \sin(\theta_k + \omega \Delta t) - v_k \sin(\theta_k))$$

$$g_y = \frac{a}{\omega^2} (\sin(\theta_k + \omega \Delta t) - \sin(\theta_k)) - \frac{1}{\omega} ((v_k + a \Delta t) \cos(\theta_k + \omega \Delta t) - v_k \cos(\theta_k))$$

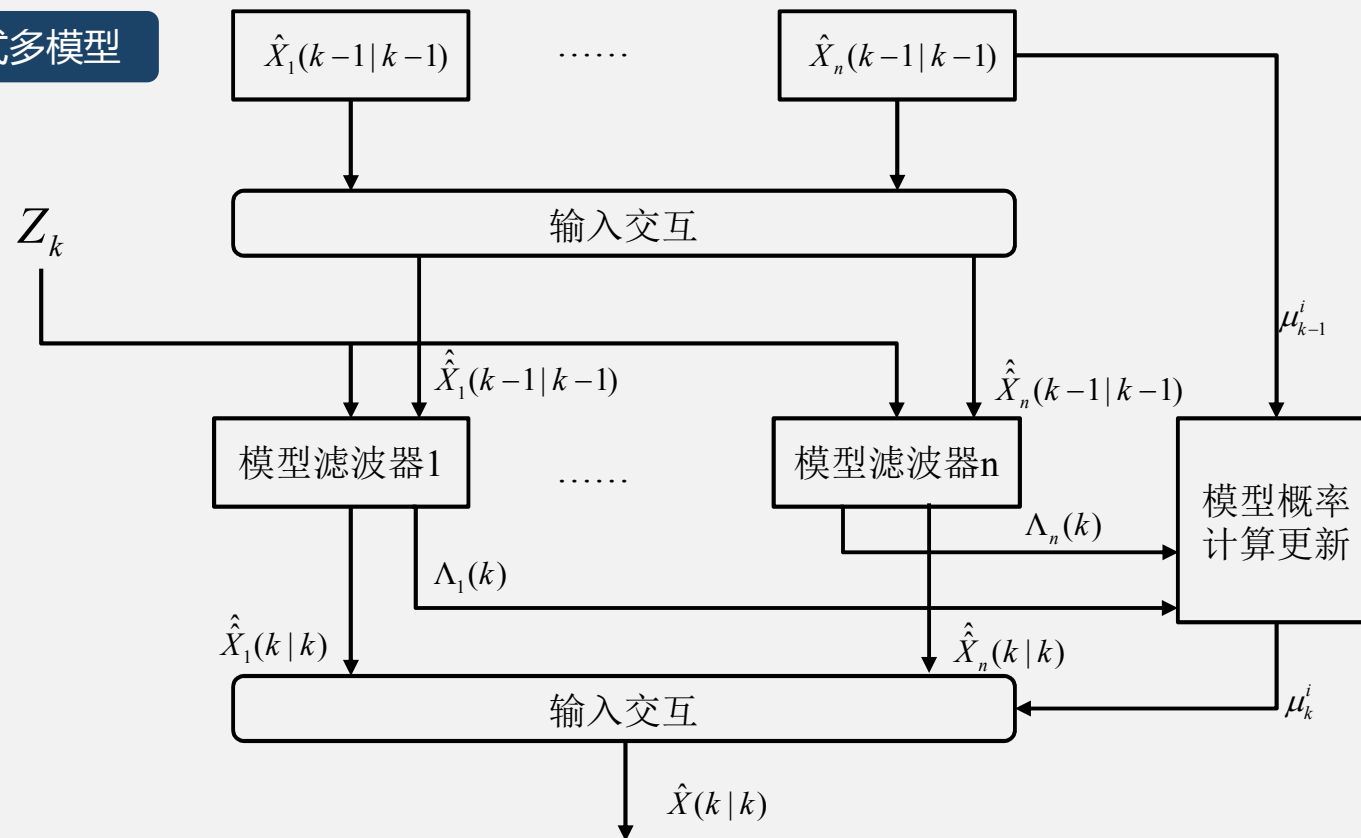
$$x_k = [px, py, v, \theta, \omega, a]^T$$



滤波更新



交互式多模型





滤波更新

UKF

UT(Unscented Transform)



KF(Kalman Filtering)

确定性采样

加权计算近似均值与近似协方差

步骤一

步骤二

步骤三

步骤四

确定参数与计算权重

对采样点非线性变换

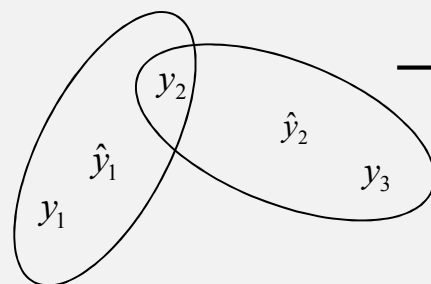


数据关联



跟踪门

联合概率数据关联



确认矩阵

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

生成可行矩阵

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



改进方法



联合概率数据关联

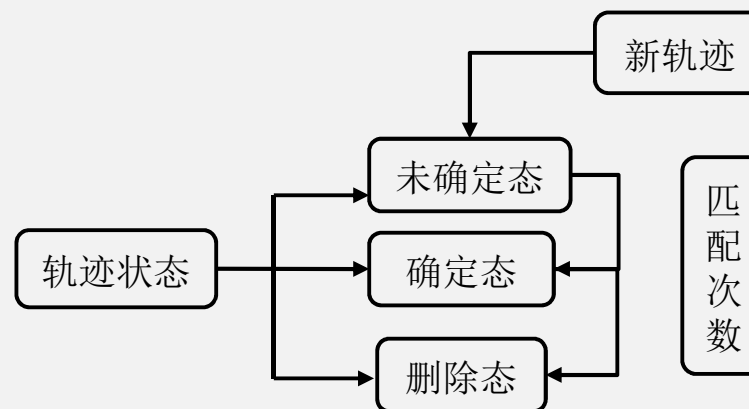
确认
矩阵

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

可行
矩阵

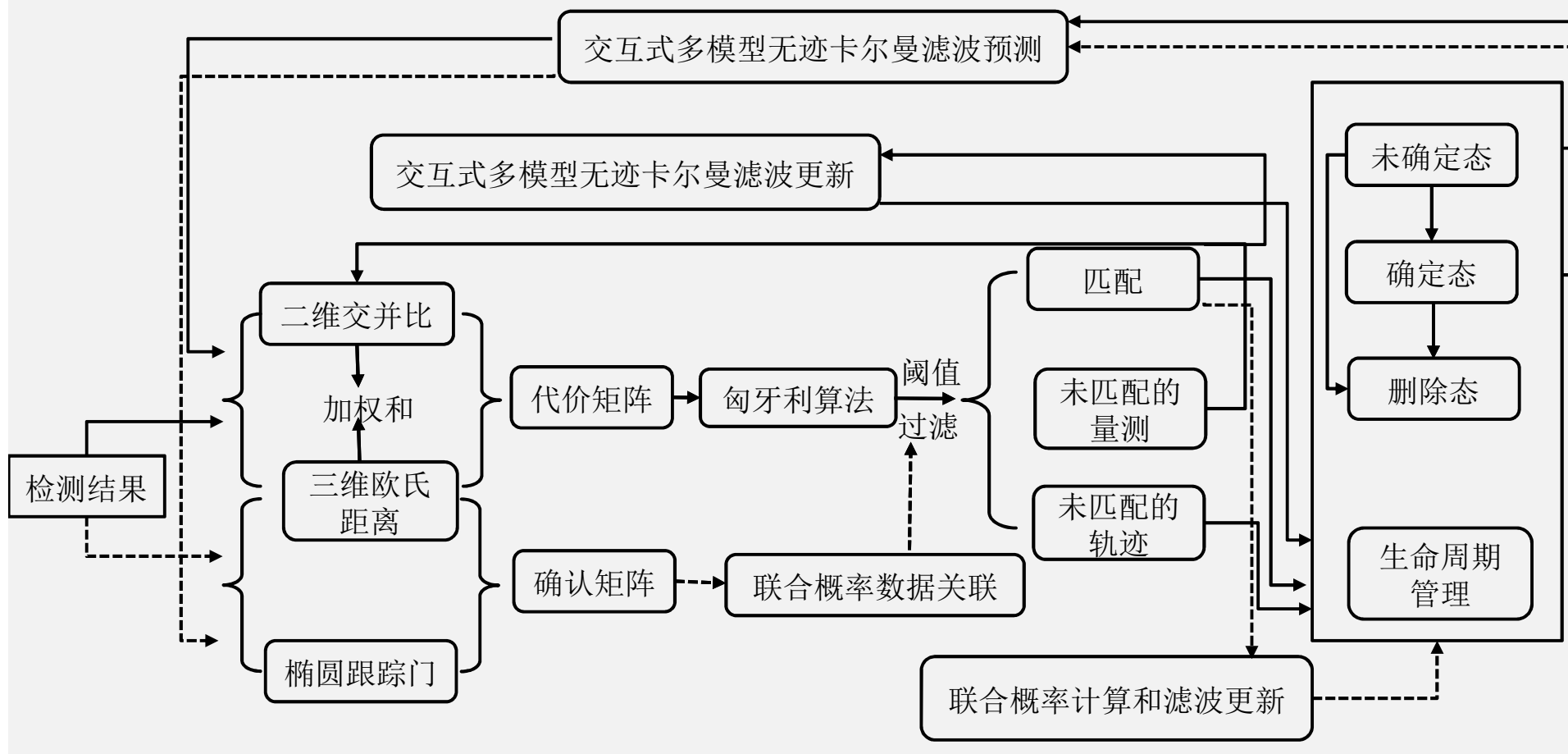
$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$





数据关联





主要评价指标



Multiple Object Tracking Accuracy

$$MOTA = 1 - \frac{FN + FP + IDS}{GT}$$

False Positive Rate

$$FAR = \frac{FP}{FP + TN}$$

Multiple Object Tracking Precision

$$MOTP = \frac{\sum_{t,i} d_{t,j}}{\sum_t c_t}$$

ID Switch

$$IDS = \sum_t id_t$$

False Positive

$$FP = \sum_t fp_t$$

ID F-Score

$$IDF1 = \frac{2TP}{2TP + FP + FN}$$



实验环境



环境配置表

所需环境	对应版本
操作系统	Ubuntu 20.04
编程语言	C++ 20, Python3.6
CPU	Inter(R) Core(TM) i7-10700@2.90Hz
Clion	2021.1.3
OpenCv	3.4.1
PCL	1.8.1
VTK	7.1.1
camke	3.19.2
g++	9.4.0
Eigen	3.3.7



结果分析

各序列实验结果对比表



数据	算法	MOTA	MOTP	FP	FAR	IDS
序列0000	AB3DMOT	53.02%	86.98%	93	0.600	0
	本文算法	72.56%	87.07%	46	0.297	1
序列0003	AB3DMOT	76.35%	86.01%	56	0.386	0
	本文算法	84.43%	86.67%	19	0.131	0
序列0007	AB3DMOT	73.78%	87.75%	464	0.579	0
	本文算法	79.21%	88.22%	281	0.351	0
序列0010	AB3DMOT	64.48%	88.35%	122	0.414	0
	本文算法	61.38%	88.80%	32	0.108	0
序列0015	AB3DMOT	69.98%	86.62%	128	0.340	0
	本文算法	73.18%	87.07%	82	0.218	0
序列0016	AB3DMOT	75.96%	85.52%	189	0.900	0
	本文算法	80.62%	85.73%	120	0.571	0
序列0019	AB3DMOT	25.52%	84.50%	591	0.556	0
	本文算法	55.92%	84.83%	316	0.298	0



结果分析



总实验结果对比表

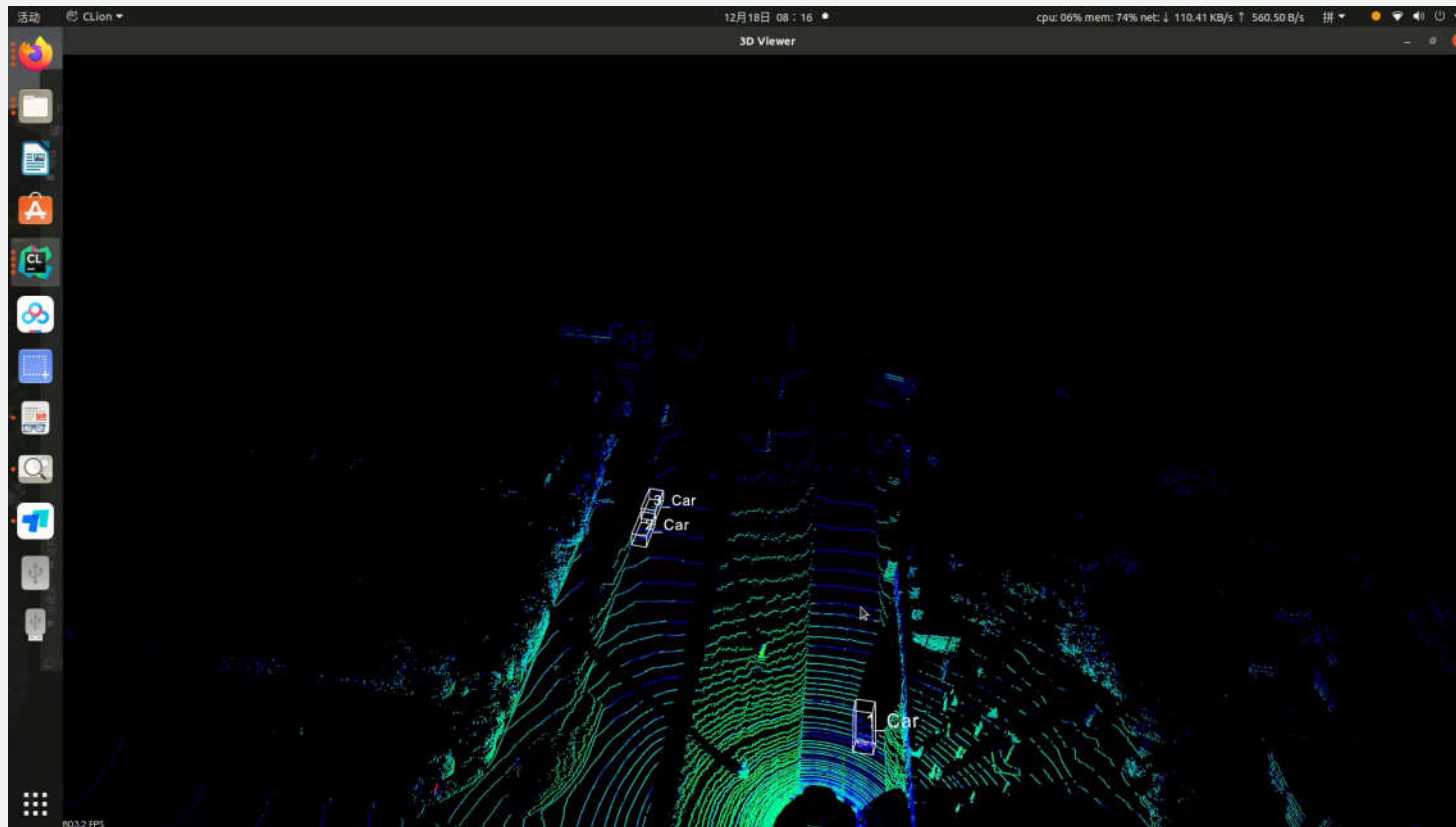
算法	MOTA	MOTP	FP	FAR	IDS	IDF1
AB3DMOT	64.58%	86.63%	1643	0.540	0	0.877
本文算法	73.31%	86.99%	896	0.294	1	0.899

测试序列实验结果对比表

数据	算法	MOTA	MOTP	FPS	IDS	IDF1
测试序列1	AB3DMOT	94.37%	90.80%	85.2	0	0.971
	本文算法	91.80%	93.08%	45.5	0	0.957
测试序列2	AB3DMOT	45.71%	63.83%	64.0	2	0.629
	本文算法	51.66%	65.52%	30.4	2	0.683



结果可视化



05
PART

总结与展望



研究总结



地面滤除

将高斯滤波和中值滤波融合在基于坡度的地面滤除算法中，滤除地面点的范围比RANSAC算法大70%左右。

点云聚类

使用改进的算法避免迭代计算，耗时比采用基于KD-Tree的DBSCAN聚类算法缩短10%左右。提出划分网格策略解决在目标聚类过程中出现的欠分割的问题。

目标跟踪

提出结合跟踪门，距离特征和匈牙利算法建立航迹的方法，避免组合爆炸，使用经过剪枝后的确认矩阵进一步减少计算量，采用轨迹状态的分级管理和匹配，有效减少ID切换错误。跟踪准确率提升8%左右，误跟踪数量和虚警率减少50%左右。



工作展望



一、在检测阶段通过聚类算法实现，并未区分车辆类别。后续研究可以结合分类算法，对目标类别做细分。



二、车载激光雷达的环境下，通过跟踪算法对目标速度的估计是相对的。后续研究可以增加速度补偿算法，实现绝对速度的计算。



三、本文通过固定帧数的方法设置轨迹的生命周期。后续研究可以采用基于深度学习的方法判断轨迹是否需要删除。