

# 模拟激光雷达点云在路侧感知算法中的应用<sup>①</sup>



邹 凯, 郭云鹏, 陈升东, 袁 峰

(广州中科院软件应用技术研究所, 广州 511466)

通讯作者: 邹 凯, E-mail: zoukai@gz.iscas.ac.cn

**摘 要:** 路侧感知算法融合车载感知算法实现了超视距感知, 基于深度学习的感知算法性能取决于激光雷达点云标签标注的质量, 而点云标签相对于二维图像更难标注, 需要大量时间人力成本进行标注, 且现行感知算法都是针对车载激光雷达. 针对这些问题, 本文提出了一种基于路侧激光雷达栅格特征聚类的感知算法, 该算法首先对路侧激光雷达点云栅格化并提取特征, 再构建深度学习模型学习栅格的初级感知信息, 最后根据初级感知信息进行聚类完成感知检测. 本文还利用仿真平台模拟路侧激光雷达点云, 并研究混合数据集在感知算法训练上的应用, 基于模拟数据预训练模型微调 (Fine-tune) 在感知算法上的应用. 实验结果表明, 本文提出的路侧感知算法具有较高的实时性与可靠性, 模拟路侧激光雷达点云有助于路侧感知算法训练, 减少路侧感知算法对标注工作的依赖, 提高感知算法性能.

**关键词:** 感知算法; 激光雷达; 模拟仿真; 深度学习

引用格式: 邹凯, 郭云鹏, 陈升东, 袁峰. 模拟激光雷达点云在路侧感知算法中的应用. 计算机系统应用, 2021, 30(6):246–254. <http://www.c-s-a.org.cn/1003-3254/7964.html>

## Application of Simulated LiDAR Point Cloud in Roadside Perception Algorithm

ZOU Kai, GUO Yun-Peng, CHEN Sheng-Dong, YUAN Feng

(Guangzhou Institute of Software Application Technology, Chinese Academy of Sciences, Guangzhou 511466, China)

**Abstract:** The roadside perception algorithm is integrated with the on-board perception algorithm to achieve over-the-horizon perception. The performance of the perception algorithm based on deep learning depends on the quality of the point cloud annotation of lidar which is harder than the annotation of 2D images because it takes longer time and calls for much manpower. In addition, existing perception algorithms based mainly on the on-board lidar. In this study, we proposes a perception algorithm based on the feature clustering of roadside lidar grids. This algorithm rasterizes the point cloud of roadside lidar and extract the features, then learn the primary perception information of the grids by creating a deep learning model for clustering on this basis. We also simulate the point cloud of roadside lidar via a simulation platform, and studies the application of the hybrid data set in training perception algorithm, which is fine-tuned by the pre-training model of simulation data. Experimental results show that the proposed perception algorithm is reliable with real-time service. Besides, simulating the point cloud of roadside lidar helps with the training of this algorithm and reduces its dependence on annotation, improving its performance.

**Key words:** perception algorithm; lidar; simulation; deep learning

① 基金项目: 广东省重点领域研发计划 (2019B010154004, 2019B090912002); 广州市科技计划 (201807010049, 201802010006)

Foundation item: Research and Development Plan of Key Areas of Guangdong Province (2019B010154004, 2019B090912002); Science and Technology Plan of Guangzhou Municipality (201807010049, 201802010006)

收稿时间: 2020-09-30; 修改时间: 2020-10-28; 采用时间: 2020-12-01; csa 在线出版时间: 2021-06-01

近年来,自动驾驶引起了研究者的广泛关注,自动驾驶技术包括感知与理解、定位定向、自主规划、决策控制等部分<sup>[1]</sup>,感知与理解是自动驾驶中关键技术之一,算法通过传感器对障碍物进行检测,按照传感器所处环境不同分为车载感知算法与路侧感知算法,路侧传感器部署在道路侧边灯杆等设施上,通过汽车到基础设施 (Vehicle-to-Infrastructure, V2I) 融合车载感知算法结果完成超视距感知<sup>[2]</sup>. 路侧感知算法使用的传感器有激光雷达、摄像头等,激光雷达因其不受光照条件影响,直接获取物体三维信息等特点,使得基于激光雷达的感知算法成为主要的方向之一<sup>[3-5]</sup>.

深度学习近年来在计算机视觉任务上获得了一系列的成果<sup>[6-8]</sup>,基于深度学习的感知算法同样也吸引了广泛的关注. 基于激光雷达的深度学习感知算法需要对激光雷达点云进行标注,而由于点云数据是三维数据且具有稀疏性,相对于二维图像更难标注,标注工作需要耗费大量的人力物力成本<sup>[9]</sup>,且现行的主流感知算法都是基于车载激光雷达<sup>[10]</sup>,而路侧激光雷达由于所处环境、放置方式不同,使得车载感知算法泛化到路侧感知算法上鲁棒性较差,所以如何构建性能良好的基于深度学习的路侧感知算法,并减少对点云标注的需求,使得算法能在实际应用中快速部署,是目前路侧感知算法的重点与难点.

面对该问题,本文提出了一种基于栅格 (grid) 聚类的路侧感知算法,并通过 LGSVL 仿真平台 (LGSVL Simulator) 模拟路侧激光雷达获得点云,研究模拟点云在算法训练中的应用. 基于栅格聚类的路侧感知算法首先将点云进行栅格化,并将栅格内点云转化为统计特征,再构建全卷积网络对栅格特征进行多任务学习,训练后的网络获得栅格的分类、聚类、高度等初级语义信息,最后采用聚类算法将初级语义信息进行聚类,完成端到端感知输出. 模拟点云是为了能在路侧感知算法训练中替代真实点云,减少点云标注数量,所以本文设计多组应用模拟点云的实验,验证在网络训练中模拟点云对算法的帮助.

实验结果表明,本文提出的路侧感知算法兼具实时性与准确性,能有效的识别障碍物,并已部署在路侧边缘计算设备实地使用,模拟点云的不同方案实验结果也证明模拟点云有助于路侧感知算法的训练,减少了真实数据的标注依赖,节约了算法实际应用中的标注成本.

本文主要的贡献体现在以下 3 个方面:

(1) 提出一种基于栅格聚类的路侧感知算法,以路侧激光雷达点云为输入,对点云进行栅格特征表示,构建深度学习网络进行多任务学习,通过栅格聚类完成端到端感知. 实验结果表明该算法具有较好的实时性与准确性,且已实际应用于路侧感知平台.

(2) 利用仿真平台模拟路侧激光雷达环境并获取模拟点云,通过不同的实验研究模拟点云在感知算法上的应用. 实验结果表明,模拟点云通过混合数据集、预训练模型可以增强感知算法的鲁棒性,减少对真实标注点云的依赖.

(3) 通过已部署的路侧激光雷达采集点云数据,并通过人工标注的方式进行标注,构建了路侧激光雷达点云数据集并开源,有效解决了路侧激光雷达点云数据集的缺失.

## 1 相关工作

点云数据是在欧式空间下的点的一个子集,点云数据具有无序性、点与点的连接性、旋转不变性 3 个特征<sup>[3]</sup>,所以根据点云特征的代表方式可将感知算法模型分为:基于体素模型 (voxel-based models) 方法、基于原始点云模型 (points clouds based models) 方法、基于图模型 (graph-based models) 方法、基于投影栅格模型 (view grids based models).

基于体素模型的方法是将三维点云划分到体素 (voxel) 中,文献 [3] 中提出 VoxNet 网络,对体素进行随机采样归一化,利用三维卷积模型学习全局特征张量. 基于原始点云模型的方法是将原始点云作为特征,文献 [4] 提出 PointNet 网络,模型通过多层感知机获取每个点云的空间特征,再通过空间转化网络 (spatial transform network) 级联为局部特征. 基于图模型的方法是将点云转化为图结构数据,以点作为图中的节点 (node),以点之间的关系作为图中的边 (graph),文献 [11] 提出 EdgeConv 模型,利用动态图卷积学习局部特征. 基于投影栅格模型是将点云进行鸟瞰视角 (Bird's Eye View, BEV) 投影或前视图 (Front View, FV) 投影再栅格化进行特征表示,文献 [12] 提出了一种 PIOXR 网络,利用一步检测器对 BEV 投影的栅格进行学习.

基于体素模型的方法运算速度较快,但是会损失体素内点的信息<sup>[13]</sup>;基于原始点云模型的方法保留了

点的几何特征,但是运算效率较低;基于图模型的方法增加了点之间的相互信息,但是运算效率较低.本文采用基于体素模型,具有较好的运算精度与效率,同时针对点云栅格化损失部分信息,本文提出一种栅格统计特征表示,即兼具了该方法的运算效率也保留了栅格几何信息.

深度学习中使用模拟数据的应用也越来越多,文献[14]利用域随机化模拟真实图像,并在真实图像上进行 fine-tune 实验.文献[15]通过增强现实技术,将虚拟引擎中的物体放置到真实环境图像中,并利用混合数据进行对比实验.文献[16]通过 pointNet 模型将真实点云的前景与背景分离,再将模拟物体的点云设置

到真实点云的背景中生成模拟点云,通过感知算法结果对比了两种点云的相似性.本文采用 LGSVL 仿真平台构建路侧激光雷达并获取雷达点云,并通过实验探索模拟点云在感知算法上的应用.

## 2 基于栅格聚类的路侧感知算法

本文提出的路侧感知算法对交通参与者进行检测,输入路侧激光雷达点云,输出交通参与者的三维边界框(bounding boxes),实现端到端的交通参与者目标分类与检测.算法流程如图1所示,本节将介绍如何进行点云特征表示,网络结构,聚类,以及在模型训练与验证中的细节.

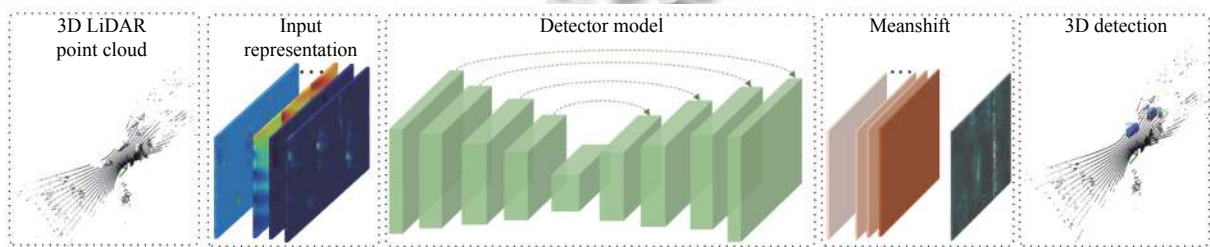


图1 基于栅格聚类感知算法示意图

### 2.1 点云表示

路侧激光雷达获取到的点云是三维、非结构化数据,而标准卷积需要结构化数据,所以本文采用将三维点云栅格化,再结合每个栅格的物理特性,通过统计来获得每个栅格的特征.激光雷达根据回波时间(Time Of Flight, TOF)原理扫描检测范围内的所有物体,而路侧感知算法主要关注道路及交通参与者,所以限定每帧点云以雷达为原点范围 $L \times W \times H$ 为检测区域,检测区域离散化为栅格,每个栅格的分辨率为 $d_L \times d_W \times d_H$ ,即将点云转化为一个 $L/d_L \times W/d_W \times H/d_H$ 的张量.可视化结果见图2.

点云被离散到栅格中,每个栅格所含的点云数目与点云的空间分布相关,根据感知算法对点云物理特性需求,本文将每个栅格转化为一个8通道的特征向量.特征向量各通道含义分别为:最高点高度、平均高度、栅格朝向、栅格距离、最大反射强度、平均反射强度、点云数量、非空栅格标志.最高点高度表示栅格内最高点云的高度,平均高度表示栅格内所有点云的高度平均值,这两个特征表征了点云高度信息,由于等交通参与者在高度上具有区分性,高度特征可以排除树木、楼栋等干扰也可以区分交通参与者类别.栅

格朝向表示栅格中心在雷达坐标系中的角度,栅格距离表示栅格中心距离雷达距离,这两个特征表征了激光雷达扫描时近密远疏的特性,栅格与雷达的几何关系.最大反射强度表示指栅格内最大反射强度,平均反射强度表示栅格内平均反射强度,这两个特征表征了点云反射强度信息,不同材质的物体反射强度不同,反射强度信息有助于物体的分类.点云数据表示栅格内点云的总数,非空栅格标志表示标识该栅格是否有点云的标识,这两个特征表示了点云内点的分布情况,由于点云具有稀疏性,所以具有没有点云的空栅格,空栅格与有点栅格的排布表征了点云在空间中的排布情况.经以上栅格化与统计特征表示后,每帧点云可以转化为维度为 $L/d_L \times W/d_W \times H/d_H \times 8$ 的特征张量.

### 2.2 深度学习网络结构

路侧感知算法构建基于全卷积的网络进行多任务学习,对每个栅格进行分割,获取物体检测的初级语义信息.如图1所示,输入栅格特征,深度学习网络通过下采样(down sample)对栅格特征进行卷积,4次下采样后分辨率分别为原始栅格特征的1/2、1/4、1/8、1/16、1/32,网络无需采用待选框(proposal),再通过转

置卷积<sup>[17]</sup>上采样 (up sample) 对特征的分辨率进行还原, 最终学习到一个 8 通道 的热图 (heat map).

如图 3 所示为深度学习网络结构示意图, 其中输入即为 2.1 节中的点云 8 通道特征, 图中 {C1,C2,C3,C4,C5} 表示下采样中的各特征层, 以特征图的分辨率可以分为 5 层, {D1,D2,D3,D4,D5} 表示上采样部分, 中间横向箭头表示跳层连接.



图 2 Offset 向量在栅格内的可视化

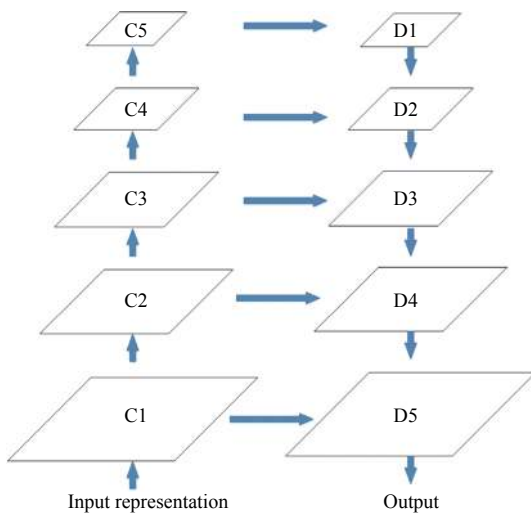


图 3 深度学习网络结构示意图

下采样的网络参数如表 1 所示, 表中 input size 与 output size 分别表示输入与输出该特征层的张量大小,  $8 \times N \times M$  即为 8 通道的点云特征, layer 列中大括号表示了该特征层中还有的卷积层, 每层卷积由 4 个数字表示, 分别表示卷积核大小、卷积步长、填充、卷积核通道数, 例如 C1 特征层中的第一个“ $1 \times 1, 1, 0, 24$ ”分别

表示该层卷积的卷积核为  $1 \times 1$  大小, 卷积时的步长为 1, 填充 (padding) 为 0, 卷积核的通道为 24. 特征图经过下采样后, 获得的输出为  $192 \times N/2 \times M/2$  的高语义低分辨率的特征图. 每一层卷积后都采用 ReLU 非线性激活函数.

表 1 下采样网络各层参数表

名称	Input size	Output size	Layer
C1	$8 \times N \times M$	$48 \times \frac{N}{2} \times \frac{M}{2}$	$\left[ \begin{array}{l} 1 \times 1, 1, 0, 24 \\ 3 \times 3, 1, 1, 24 \\ 3 \times 3, 2, 1, 48 \\ 3 \times 3, 1, 1, 48 \end{array} \right] \times \text{Conv}$
C2	$48 \times \frac{N}{2} \times \frac{M}{2}$	$64 \times \frac{N}{4} \times \frac{M}{4}$	$\left[ \begin{array}{l} 3 \times 3, 2, 1, 64 \\ 3 \times 3, 1, 1, 64 \\ 3 \times 3, 1, 1, 64 \end{array} \right] \times \text{Conv}$
C3	$64 \times \frac{N}{4} \times \frac{M}{4}$	$96 \times \frac{N}{8} \times \frac{M}{8}$	$\left[ \begin{array}{l} 3 \times 3, 2, 1, 96 \\ 3 \times 3, 1, 1, 96 \\ 3 \times 3, 1, 1, 96 \end{array} \right] \times \text{Conv}$
C4	$96 \times \frac{N}{8} \times \frac{M}{8}$	$128 \times \frac{N}{16} \times \frac{M}{16}$	$\left[ \begin{array}{l} 3 \times 3, 2, 1, 128 \\ 3 \times 3, 1, 1, 128 \\ 3 \times 3, 1, 1, 128 \end{array} \right] \times \text{Conv}$
C5	$128 \times \frac{N}{16} \times \frac{M}{16}$	$192 \times \frac{N}{32} \times \frac{M}{32}$	$\left[ \begin{array}{l} 3 \times 3, 2, 1, 192 \\ 3 \times 3, 1, 1, 192 \end{array} \right] \times \text{Conv}$

如图 3 所示, 上采样的阶段, 使用转置卷积对高语义特征进行转置, 再通过跳层连接将下采样中同一分辨率的特征图进行拼接, 如表 2 所示上采样中各层的参数, 其中 TConv 表示为转置卷积.

表 2 上采样网络各层参数表

名称	Input size	Output size	Layer
D1	$192 \times \frac{N}{32} \times \frac{M}{32}$	$128 \times \frac{N}{16} \times \frac{M}{16}$	$[3 \times 3, 1, 1, 192] \times \text{Conv}$ $[4 \times 4, 2, 1, 128] \times \text{TConv}$
D2	$128 \times \frac{N}{16} \times \frac{M}{16}$	$96 \times \frac{N}{8} \times \frac{M}{8}$	$[3 \times 3, 1, 1, 128] \times \text{Conv}$ $[4 \times 4, 2, 1, 96] \times \text{TConv}$
D3	$96 \times \frac{N}{8} \times \frac{M}{8}$	$64 \times \frac{N}{4} \times \frac{M}{4}$	$[3 \times 3, 1, 1, 96] \times \text{Conv}$ $[4 \times 4, 2, 1, 64] \times \text{TConv}$
D4	$64 \times \frac{N}{4} \times \frac{M}{4}$	$48 \times \frac{N}{2} \times \frac{M}{2}$	$[3 \times 3, 1, 1, 64] \times \text{Conv}$ $[4 \times 4, 2, 1, 48] \times \text{TConv}$
D5	$48 \times \frac{N}{2} \times \frac{M}{2}$	$8 \times N \times M$	$[3 \times 3, 1, 1, 48] \times \text{Conv}$ $[4 \times 4, 2, 1, 14] \times \text{TConv}$

网络输出热图 8 通道分别表示为: x 轴偏移量 (offset) 热图、y 轴偏移量热图、背景点热图、地面点热图、交通参与者点热图、高度热图、x 轴朝向 (heading) 热图、y 轴朝向热图. x 轴热图偏移量、y 轴偏移量热图分别表示栅格在 x、y 轴上距物体重心的距离, 将两个

热图中的值以箭头的形式表示在原始点云上,在雷达坐标上  $x$ 、 $y$  轴的偏移量表示为一个向量,其值显示了朝向物体中心点聚集的趋势.背景点热图表示该栅格是否是背景点云栅格,点云栅格中还有树木等背景点云,背景点热图用于背景栅格过滤.地面点热图表示该栅格是否是地面点栅格,地面点热图用于过滤地面点栅格对检测的干扰.交通参与者热图表示栅格为交通参与者的标志,交通参与者为大车、小车、非机动车、行人,交通参与者热图用于物体分类.高度热图表示栅格的高度,用于异常高度栅格的过滤. $x$  轴朝向热图  $x$ 、 $y$  轴朝向热图分别表示栅格在  $x$ 、 $y$  轴的朝向,用于聚类中向量的计算.

感知算法网络对分类、检测进行多任务学习,在网络训练时采用多任务损失函数 (multi-task loss)<sup>[18]</sup> 对类别、高度、朝向、偏移量进行学习.本文采用 Focal loss<sup>[19]</sup> 对类别进行分类, Smooth L1 loss 对高度、朝向进行回归,对数比率损失函数对 offset 进行回归.

$$L_{\text{total}} = \frac{1}{2} \sum_{i=1}^N (\lambda^2 L_i + \log(\lambda^2 + 1)) \quad (1)$$

网络总损失函数由 4 个函数加权求和,如式 (1) 所示,其中  $N$  表示多任务学习中损失函数的个数,本文中  $N=4$ ,  $\lambda$  表示加权系数,加权系数与损失函数个数有关  $\lambda = 1/N$ ,  $L_0$  到  $L_4$  分别为分类损失函数  $L_{\text{cls}}$ 、高度损失函数  $L_{\text{hei}}$ 、朝向损失函数  $L_{\text{head}}$ 、偏移量损失函数  $L_{\text{offset}}$ :

$$L_{\text{cls}} = \frac{1}{P} \sum_{i=1}^N -\alpha(1 - p_i)^\gamma \log(p_i) \quad (2)$$

由于点云栅格中有背景点干扰,且交通参与者中各类别出现的数目不一致,所以在分类任务中有类别不平衡问题,所以本文采用如式 (2) 所示 focal loss 作为分类损失函数,其中  $P$  表示含有所有点云的栅格总数,  $\alpha$  表示各种不同类别点云的权重,其中背景点栅格权重为 0.1,地面点栅格权重为 0.75,交通参与者栅格权重为 0.9,  $p_i$  表示网络输出该栅格的分类概率,  $\gamma$  为 2.

$$L_{\text{hei}}(g, \hat{g}) = \frac{M}{P} \sum_{i=1}^P \text{Smooth } L_1(g_i - \hat{g}_i) \quad (3)$$

$$L_{\text{head}}(h, \hat{h}) = \frac{M}{P} \sum_{i=1}^P \text{Smooth } L_1(h_i - \hat{h}_i) \quad (4)$$

$$\text{Smooth}_{L_1}(x) = \begin{cases} 0.5\beta^2, & \text{if } |x| \leq \frac{1}{\beta^2} \\ |x| - \frac{0.5}{\beta}, & \text{otherwise} \end{cases} \quad (5)$$

高度值、朝向值回归时,由于点云的稀疏性以及背景点的干扰,在回归任务时候容易出现离群点,离群点会使得损失函数变大,反向传播过程中梯度增大,从而影响其它任务的收敛,本文采用 Smooth L1 损失函数如式 (3)、式 (4) 所示,其中式 (3) 表示高度回归损失函数,  $g$  表示预测的高度值,  $\hat{g}$  表示真实的高度值,  $P$  表示有点栅格的掩模 (mask) 矩阵,矩阵由布尔数值组成,有点云的栅格为 1,没点云的栅格为 0. Smooth L1 函数如式 (5) 所示,其中  $\beta$  默认设置为 10. 式 (4) 表示朝向回归损失函数,  $h$  表示预测朝向,  $\hat{h}$  表示真实朝向.

$$L_{\text{offset}} = L_{\text{offset}_x} + L_{\text{offset}_y} \quad (6)$$

$$L_{\text{offset}_x}(o_x, \hat{o}_x) = \frac{M}{P} \sum_i \log\left(\frac{|o_{x_i} - \hat{o}_{x_i}|}{\hat{o}_{x_i}} + 1\right) \quad (7)$$

$$L_{\text{offset}_y}(o_y, \hat{o}_y) = \frac{M}{P} \sum_i \log\left(\frac{|o_{y_i} - \hat{o}_{y_i}|}{\hat{o}_{y_i}} + 1\right) \quad (8)$$

Offset 的回归损失函数如式 (6) 所示,由式 (7) 所示  $x$  轴的 offset 回归函数与式 (8) 所示  $y$  轴的 offset 回归函数求和组成.

### 2.3 聚类

深度学习网络通过多任务损失函数学习到每个栅格学习的类别、偏移量等信息,每个目标物体是由空间中的多个栅格组成,属于同一物体的栅格具有相似的物理特性,所以本节通过均值漂移聚类算法将栅格聚类成物体.

如图 4 所示为聚类初始点的计算,通过深度学习网络学到了 8 通道热图,遍历栅格,先通过背景点热图过滤背景点栅格,剩下前景点栅格,深度学习网络学习到的  $x$ 、 $y$  轴偏移量热图表示的是该栅格对目标的偏移量,根据该信息构建该栅格的并查集,并将不相交集合并进行合并,再通过高度热图对并查集过滤,过滤掉不符合高度的栅格,其剩下的即作为聚类的初始点.

均值漂移公式,如式 (9) 为漂移向量计算公式,  $x$  为初始聚类原点,聚类原点的选取是根据网络学习到的在该原点半径为  $h$  的范围内共有  $n$  个点云,遍历每个点云  $x_i$  与  $x$  的 offset 向量距离并求和,其中  $c_{k,d}/nh^d$  表示单位密度,  $k$  为高斯核函数.

$$f_{h,K(x)} = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k \left( \left\| \frac{(x-x_i)}{h} \right\|^2 \right) \quad (9)$$

迭代式 (9), 每次使得式 (9) 获得最大偏移向量, 即将如式 (10) 其求导

$$\hat{\nabla} f_{h,K(x)} = \hat{f}_{h,G(x)} \frac{2c_{k,d}}{h^2 c_{g,d}} m_{h,G}(x) \quad (10)$$

其中,  $\hat{f}_{h,K(x)}$  如式 (11) 所示, 令  $g(x) = -k'(x)$ :

$$\hat{f}_{h,K(x)} = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n g \left( \left\| \frac{(x-x_i)}{h} \right\|^2 \right) \quad (11)$$

$m_{h,G(x)}$  表示 Meanshift 向量, 新迭代的原点即为式 (12) 为零时候  $x$  的值.

$$m_{h,G(x)} = \frac{\sum_{i=1}^n x_i g \left( \frac{\|x-x_i\|^2}{h} \right)}{\sum_{i=1}^n g \left( \frac{\|x-x_i\|^2}{h} \right)} - x \quad (12)$$

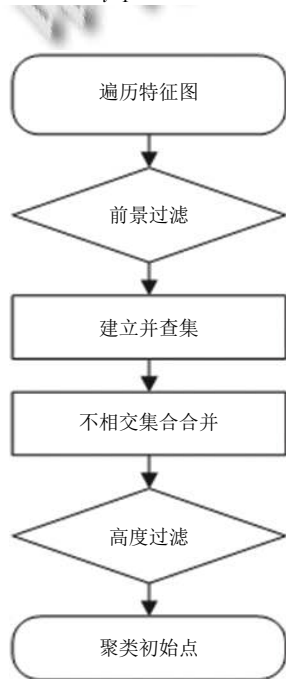


图4 聚类初始点选取

通过漂移聚类, 将一个物体的栅格聚类到一个范围内, 再对聚类中同一物体栅格类别进行统计, 置信高的即为该物体的类别. 再通过栅格高度回归值对异常高度进行过滤, 输出端到端物体.

### 3 模拟路侧激光雷达点云

机械式激光雷达通过激光束的旋转扫描物体, 本

文模拟 16 线机械式激光雷达, 水平测角为  $360^\circ$ , 垂直测角为  $-15^\circ \sim 15^\circ$ , 垂直测角分辨率为  $2^\circ$ , 垂直范围内共 16 线激光光束, 激光雷达扫描结果以点云的形式储存, 点云反应了被扫描物体的距离与反射强度, 距离值转化为激光雷达为原点的  $x, y, z$  坐标, 与反射强度共同构成点云的 4 维信息.

本文采用 LGSVL 仿真平台<sup>[20]</sup> 模拟路侧激光雷达环境, 并设置动态交通参与者生成模拟点云. LGSVL 是一款开源的专注于自动驾驶的仿真平台, 基于 Unity 虚拟引擎, 提供环境模拟与各种交通参与者的模型, 同时可以在虚拟环境中模拟传感器的收发. LGSVL 仿真平台中的激光雷达可配置参数, LGSVL 仿真平台构建路侧激光雷达设置的部署, 在路侧端设置 3.6 米高, 雷达朝向道路来车方向, 为了扫描道路中的物体, 将雷达 pitch 角度偏转  $31.25$  度, 并设置雷达模型中测距参数为 150 米.

如图 5 所示模拟点云与真实点云对比图, 图 5(a) 为经过 LGSVL 模拟的模拟点云图, 图 5(b) 为真实点云图, 图中蓝色框为小车, 红色框为小车, 从图中可以看出, 经过 LGSVL 仿真平台模拟真实环境与路侧雷达放置方式, 在雷达近端呈现的点云与模拟的相似, 只是在道路周边建筑物, 远端点云的稀疏后的噪声有所不同, 但是作为路侧激光雷达还是对道路上的机动车数据进行了大量的补充.

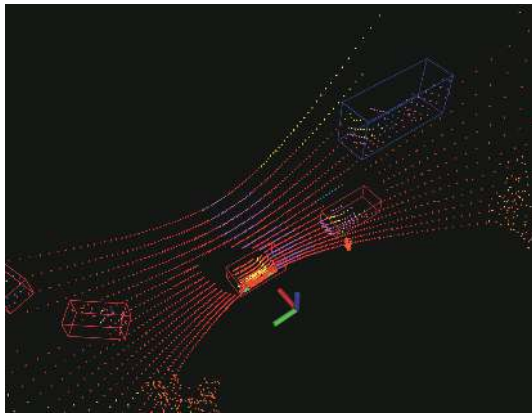
## 4 实验结果与分析

文利用真实的路侧激光雷达点云数据对感知算法进行训练与验证, 通过端到端的交通参与者类别、位置的输出来验证算法的性能. 通过 LGSVL 仿真平台模拟点云, 利用感知算法在模拟点云与真实点的混合数据集上训练, 在模拟数据集上预训练模型的结果来验证模拟点云对路侧感知算法的帮助.

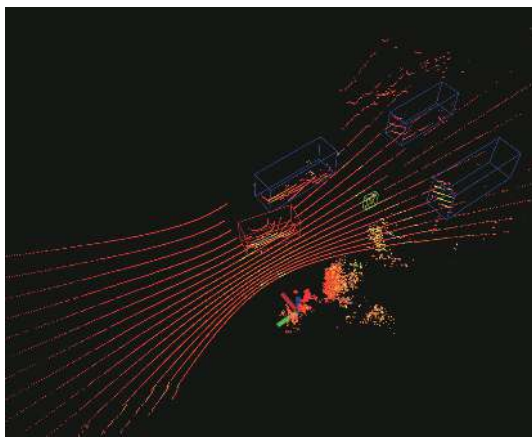
### 4.1 实验数据

用于自动驾驶上的点云数据大多数是自动驾驶汽车利用车载激光雷达采集, 本文提出的路侧激光雷达应用于路侧激光雷达点云, 所以本文利用部署在路侧的激光雷达采集了 1700 帧点云, 其中 1300 帧作为训练集, 400 帧作为验证集. 点云通过人工进行标注, 所使用的雷达为速腾聚创 16 线机械式激光雷达, 雷达所处环境如图 6 所示园区内主干道, 激光雷达 pitch 角倾斜  $31.25$  度. LGSVL 仿真平台构建如图 7 所示的环境, 并

按照路侧激光雷达一样的参数设置仿真雷达,通过设置交通参与者进行点云的采集.本次模拟点云 32045 帧,其中 27304 帧为训练集,4741 帧为验证集.



(a) 模拟点云



(b) 真实点云

图5 模拟点云与真实点云



图6 路侧激光雷达部署场景

## 4.2 实验结果

本文的数据集分为模拟点云数据集与真实点云数据集,感知算法在真实点云上进行感知算法的训练与测试.路侧感知算法只需要检测道路上交通参与者,所

以训练选择点云范围为,栅格分辨率为 0.1875,使用平移、旋转、旋转对点云进行数据增强,offset 采用指向凸包重心的向量.

如表 3 所示,训练数据集采用真实点云 (Real),模拟点云 (Sim),训练后的模型分别在两者验证集上进行验证.感知算法在真实点云上的实验反应了感知算法的性能,其中感知算法端到端检测 precision 为 84.67%, recall 为 98.27%,结果证明该算法端到端检测性能具有良好的性能,满足路需求.在两者数据集交叉实验中,基于模拟点云数据训练的算法泛化在真实点云数据上性能较差,证明模拟点云与真实点云还具有一定差异,不能直接在感知算法中代替真实点云进行训练.

如表 4 所示,采用真实点云与模拟点云混合数据集进行如表 4 所示,采用真实点云与模拟点云混合数据集进行测试.实验结果证明,利用混合数据时 precision 高于 2.46%, F1 高于 1.13%, mAP 高于 1.94%,证明了模拟数据增强了感知算法的准确性.

如表 5 所示,采用模拟点云的预训练模型,在真实点云上进行训练.实验结果证明,当利用预训练模型再利用所有真实数据训练时,其各项评价指标相差不大,但是若用预训练模型在 20% 真实点云上训练时,precision 提高 1.49%, recall、F1、mAP 有所下降,但是减少了 80% 点云数据训练,由此可以看出,在牺牲算法一定程度上该方法可以减少许多数据标注工作.

如表 6 所示,将本文的算法分布部署在 Nvidia GTX 1070 与 Nvidia Jetson AGX Xavier 显卡上进行端到端模型评估时各部分所用时间, feature 列表示点云特征处理所用时间, inference 表示深度学习模型推理所用时间, cluster & box 表示聚类及画物体框输出所用时间.实验点云特征处理由 C++ 编写,利用 PCL 库对点云进行过滤.深度学习模型转为 onnx 格式,并利用 TensorRT 进行加速.利用 ros 画物体的框并输出,如图 7 第 2 行所示,输出物体的类别信息,概率值,框体与距离雷达的距离等.本次实验是通过实时测量雷达点云数据时所求得平均值,从表 4 可知各部分的耗时,所使用 16 线机械式激光雷达频率为 10 Hz,工程上要求算法处理每帧点云小于 50 ms 即可,本文的算法在本地 GPU 设备 GTX 1070 上约为 8 ms,在边缘计算设备 Jetson AGX 上约 20 ms,均小于 50 ms,符合实时性与工程性要求.

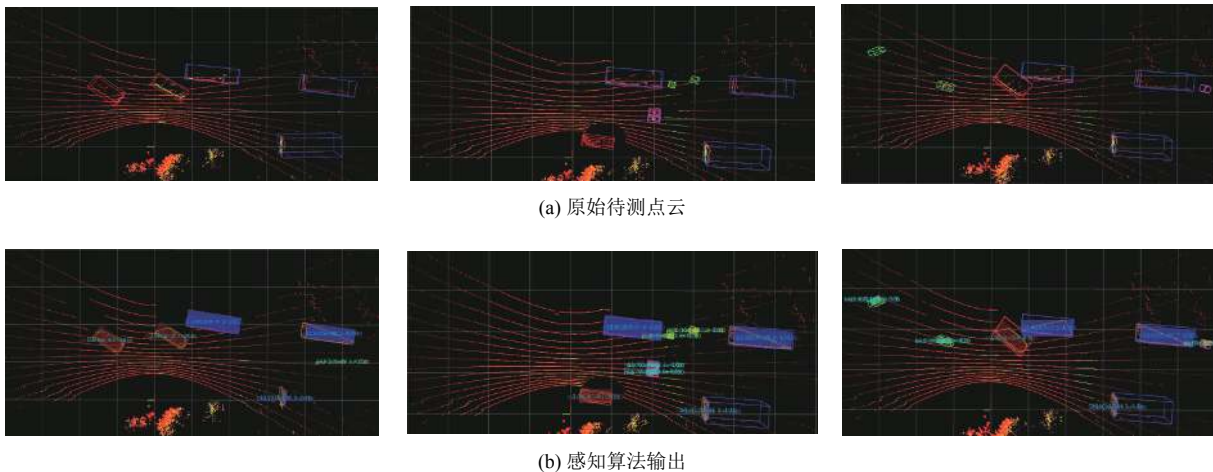


图7 感知算法输出

表3 感知算法单一数据集算法性能

训练集	测试集	Prec	Recall	F1	mAP
Real	Real	0.8476	0.9827	0.9096	0.8321
	Sim	0.7426	0.6515	0.6940	0.4838
Sim	Real	0.6233	0.7364	0.6751	0.4590
	Sim	0.9844	0.9017	0.9412	0.8877

表4 感知算法混合数据集训练算法性能

训练集	测试集	Prec	Recall	F1	mAP
Real 100%+Sim100%	Real	0.8713	0.9766	0.9209	0.8515
	Sim	0.9836	0.9006	0.9402	0.8859
Real 100%+Sim 20%	Real	0.8689	0.9784	0.9204	0.8502
	Sim	0.9812	0.9011	0.9394	0.9394

表5 感知算法在真实数据 fine-tune 性能

训练集	测试集	Prec	Recall	F1	mAP
Real 20%	Real	0.8616	0.9131	0.8866	0.7867
	Sim	0.7641	0.6013	0.6729	0.4595
Real 100%	Real	0.8519	0.9655	0.9051	0.8226
	Sim	0.7686	0.6640	0.7124	0.5104

表6 感知算法计算效率

Devices	Feature	Inference	Cluster&box	Total
GTX 1070	1.483	3.283	3.381	8.147
Jetson AGX	4.261	7.826	8.031	20.118

## 5 结论与展望

本文提出的一种适用于路侧激光雷达的路侧感知算法,对交通参与者进行分类与检测,检测性能良好,算法部署在边缘计算设备上并达到每帧约 20 ms 的实时性能.此外本文利用仿真平台模拟路侧激光雷达并采集点云,利用模拟点云在感知算法训练上的应用,证明了模拟点云通过混合数据、预训练模型上可以帮助感知算法训练增强性能,在一定程度下可以减少算法对

标注数据的依赖,减少标注工作量.

由于本文研究的路侧感知算法应用于路侧激光雷达,本文感知算法的泛化性、鲁棒性还需要在更大样本的实验数据中进行进一步验证和提升,在后续的研究中,关注实验样本库的构建,进一步对算法性能进行改进.

## 参考文献

- 蔡云飞,石庭敏,唐振民.基于双多线激光雷达的非结构化环境负障碍感知技术.自动化学报,2018,44(3):569-576.
- Krämer A, Schöller, Gulati D, *et al.* Providentia—A large-scale sensor system for the assistance of autonomous vehicles and its evaluation. arXiv preprint arXiv:1906.06789, 2019.
- Maturana D, Scherer S. VoxNet: A 3D convolutional neural network for real-time object recognition. Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems. Hamburg, Germany. 2015. 922-928. [doi: 10.1109/IROS.2015.7353481]
- Garcia-Garcia A, Gomez-Donoso F, Garcia-Rodriguez J, *et al.* PointNet: A 3D convolutional neural network for real-time object class recognition. Proceedings of 2016 International Joint Conference on Neural Networks. Vancouver, BC, Canada. 2016. 1578-1584. [doi: 10.1109/IJCNN.2016.7727386]
- Shi SS, Guo CX, Jiang L, *et al.* PV-RCNN: Point-voxel feature set abstraction for 3D object detection. Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA. 2020. 10526-10535. [doi: 10.1109/CVPR42600.2020.01054]
- He KM, Gkioxari G, Dollár P, *et al.* Mask R-CNN. IEEE



- Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386–397. [doi: [10.1109/TPAMI.2018.2844175](https://doi.org/10.1109/TPAMI.2018.2844175)]
- 7 Redmon J, Divvala S, Girshick R, *et al.* You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 779–788.
  - 8 Palangi H, Deng L, Shen YL, *et al.* Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(4): 694–707. [doi: [10.1109/TASLP.2016.2520371](https://doi.org/10.1109/TASLP.2016.2520371)]
  - 9 Dworak D, Ciepiela F, Derbisz J, *et al.* Performance of LiDAR object detection deep learning architectures based on artificially generated point cloud data from CARLA simulator. Proceedings of 2019 24th International Conference on Methods and Models in Automation and Robotics. Miedzyzdroje, Poland. 2019. 600–605.
  - 10 Li Y, Ma LF, Zhong ZL, *et al.* Deep learning for LiDAR point clouds in autonomous driving: A review. IEEE Transactions on Neural Networks and Learning Systems, 2020, doi: [10.1109/TNNLS.2020.3015992](https://doi.org/10.1109/TNNLS.2020.3015992).
  - 11 Wang Y, Sun YB, Liu ZW, *et al.* Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics, 2018, 38(5): 146.
  - 12 Yang B, Luo WJ, Urtasun R. PIXOR: Real-time 3D object detection from point clouds. Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA. 2018. 652–7660. [doi: [10.1109/CVPR.2018.00798](https://doi.org/10.1109/CVPR.2018.00798)]
  - 13 Liu ZJ, Tang HT, Lin YJ, *et al.* Point-voxel CNN for efficient 3D deep learning. Proceedings of the 33rd Conference on Neural Information Processing Systems. Vancouver, Canada. 2019. 965–975.
  - 14 Tremblay J, Prakash A, Acuna D, *et al.* Training deep networks with synthetic data: Bridging the reality gap by domain randomization. Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Salt Lake City, UT, USA. 2018. 1082–1088.
  - 15 Alhaja HA, Mustikovela SK, Mescheder L, *et al.* Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision, 2018, 126(9): 961–972. [doi: [10.1007/s11263-018-1070-x](https://doi.org/10.1007/s11263-018-1070-x)]
  - 16 Fang J, Zhou DF, Yan FL, *et al.* Augmented LiDAR simulator for autonomous driving. IEEE Robotics and Automation Letters, 2020, 5(2): 1931–1938. [doi: [10.1109/LRA.2020.2969927](https://doi.org/10.1109/LRA.2020.2969927)]
  - 17 Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv: 1511.06434, 2015.
  - 18 Ren SQ, He KM, Girshick R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031)]
  - 19 Lin TY, Goyal P, Girshick R, *et al.* Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318–327. [doi: [10.1109/TPAMI.2018.2858826](https://doi.org/10.1109/TPAMI.2018.2858826)]
  - 20 Rong GD, Shin BH, Tabatabaee H, *et al.* LGSVL simulator: A high fidelity simulator for autonomous driving. Proceedings of 2020 IEEE 23rd International Conference on Intelligent Transportation Systems. Rhodes, TN, USA. 2020. 1–6.