·激光雷达感知与自动驾驶多模态3D目标检测技术专题·

基于改进PointPillars的激光雷达三维目标检测算法*

郭烈[1,2]　黄亮[1]　赵一兵[1]

（1.大连理工大学机械工程学院，大连 116024；2.大连理工大学宁波研究院，宁波 315016）

【摘要】为解决复杂交通场景中目标因被遮挡、截断等造成局部特征丢失而产生的漏检问题，提出了一种基于改进PointPillars的激光雷达三维目标检测算法。首先，通过改进PointPillars中的点云特征编码网络，更好地保留点云数据中重要的结构信息及其相互关系，然后，通过堆叠的稀疏卷积模块和子流形卷积模块提取并聚合多尺度点云特征，在融合多尺度特征的鸟瞰特征图上对目标的三维检测框进行分类和回归，提升网络运行效率，最后，在KITTI数据集上进行了试验验证。结果表明：与传统PointPillars算法相比，所提出的三维目标检测算法对车辆困难检测样本的3D检测平均精度（AP）提高了6.02百分点，对鸟瞰视角下行人和骑行者简单样本的AP分别提高了3.35百分点和3.18百分点，有效提升了三维目标检测精度。

主题词：激光雷达　三维目标检测　体素特征　稀疏卷积

中图分类号：U495　　**文献标志码：**A　　**DOI:** 10.19620/j.cnki.1000-3703.20240211

LiDAR-Based 3D Object Detection Algorithm Based on Improved PointPillars

Guo Lie[1,2], Huang Liang[1], Zhao Yibing[1]

（1. School of Mechanical Engineering, Dalian University of Technology, Dalian 116024; 2. Ningbo Institute of Dalian University of Technology, Ningbo 315016）

【Abstract】To avoid the missing detection phenomenon caused by loss of local features due to the occlusion and truncation under complex traffic scenes, a LiDAR-based 3D object detection algorithm based on an improved PointPillars framework is proposed. First, the point cloud feature encoding network in PointPillars is enhanced to better preserve important structural information and the relationships within the point cloud data. Then, multi-scale point cloud features are extracted and aggregated through stacked sparse convolution modules and submanifold convolution modules. On the bird's-eye-view feature map that fuses these multi-scale features, 3D bounding boxes of objects are classified and regressed, improving the network's operational efficiency. Finally, experimental tests are conducted on the KITTI dataset. The results show that compared to the traditional PointPillars algorithm, the proposed 3D object detection method improves the Average Precision (AP) of 3D detection for challenging vehicle samples by 6.02 percentage points, and increases the AP for simple pedestrian and cyclist samples in the bird's-eye view by 3.35 and 3.18 percentage points respectively, effectively enhancing 3D object detection accuracy.

Key words: LiDAR, 3D object detection, Pillar features, Sparse convolution

【引用格式】郭烈, 黄亮, 赵一兵. 基于改进PointPillars的激光雷达三维目标检测算法[J]. 汽车技术, 2026(2): 32-38.

GUO L, HUANG L, ZHAO Y B. LiDAR-Based 3D Object Detection Algorithm Based on Improved PointPillars[J]. Automobile Technology, 2026(2): 32-38.

1　前言

准确高效的环境感知是实现可靠自动驾驶的核心前提[1]。三维目标检测作为环境感知的关键任务，能够在交通场景中识别所有感兴趣目标，并确定其位置、尺寸和方向等三维信息。目前，激光雷达已成为该领域的主流传感器[2]，根据对激光雷达点云特征表示的差异，三维目标检测方法可分为基于点云和基于体素两类。

基于点云的方法直接从原始点云中提取特征并生成目标检测框。Shi等[3]基于PointNet++架构进行特征提取和点云转换学习局部空间特征，获得了精确的候选框和预测置信度。Yang等[4]提出稀疏到稠密网络应用

*基金项目：国家重点研发计划项目（2022YFB2602305）；国家自然科学基金项目（52372409，52175078）。

点云池化网络生成候选框的特征，实现类别预测和回归参数计算，使用交并比分支网络提高了定位精度。Liao等[5]设计点云地图特征提取与残差特征融合网络，改善了远距离物体的检测性能。基于点云的方法虽能较完整地保留点云结构信息，但面临点云分布稀疏、搜索邻域点云特征耗时长等问题。

基于体素的方法将点云离散化到三维体素特征，应用三维卷积神经网络提取该特征并转换为鸟瞰特征图，生成目标检测框。如Zhou等[6]提出了包含特征编码、卷积中间层和目标检测的体素网络（VoxelNet），但其计算成本高，难以满足自动驾驶的实时性需求。为此，Yan等[7]提出稀疏嵌入卷积检测网络，降低了计算复杂度；Lang等[8]提出点云柱子编码网络（PointPillars），将点云划分为垂直支柱网格，使用PointNet[9]网络对点云进行编码，得到稀疏的二维鸟瞰特征图，再由头部网络完成检测；Zhang等[10]利用三维候选框和偏移生成模块确定目标坐标和形状自适应偏移量，提取了多尺度关键点特征并进行细化，以提高预测置信度。基于体素的方法在点云离散化时会造成大量结构信息和点云相互关系的丢失，影响检测性能。

针对点云投影过程中目标高度和结构信息丢失的问题，虽可通过划分体素网格并保留点云高度信息解决，但体素化仍会产生结构信息的量化损失。为此，本文提出基于改进PointPillars的三维目标检测算法，设计新型点云特征编码网络，搜索点云邻域特征并进行位置编码，聚合体素单元内的局部结构特征，在鸟瞰特征图上进行特征提取并最终生成目标检测框，实现高效准确的三维目标检测。

2 基于改进PointPillars的三维目标检测

2.1 算法整体框架

本文提出的三维目标检测算法主要由体素编码网络、特征提取网络和检测头部网络3个部分组成，整体框架如图1所示。

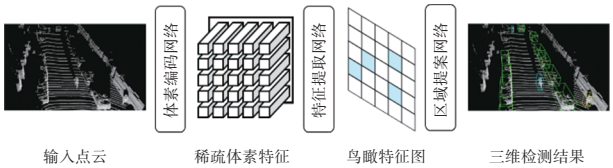


图1 三维目标检测算法框架

算法以激光雷达传感器扫描得到的点云数据作为输入，首先使用基于体素的点云特征编码网络将无序且不规则的点云转换为稀疏的体素特征，然后基于稀疏卷积的特征提取网络通过堆叠的稀疏卷积模块和子流形卷积模块提取并聚合多尺度点云特征，最后，检测头部网络在鸟瞰特征图上对目标的三维检测框进行分类和回归。

2.2 体素特征编码

受PointNet网络直接从点云数据中提取特征的启发，VoxelNet提出了由多层感知机和对称函数组成的体素特征编码网络，使用多层感知机从各点云坐标中单独学习点云特征，并使用对称函数聚合各体素单元内的全部点云特征，该方法虽然能够较好地实现点云特征的保留，但无法准确描述点云间相互关系所表达的结构信息。

为此，本文提出一种改进的体素特征编码网络，如图2所示。改进后的体素特征编码网络主要由点云结构映射模块、邻域特征搜索模块和位置编码网络组成。其中，结构映射模块采用分组采样和集合抽象策略聚合点云和体素的局部结构特征，邻域特征搜索模块根据目标点云所在体素单元的坐标高效搜索点云局部邻域内的有效特征，位置编码网络根据点云相对体素单元中心的坐标对点云位置信息进行编码。该体素特征编码网络避免了复杂的局部几何特征提取器，有效利用经过高度优化的前馈神经网络的效率优势，并在点云特征提取方面具有出色的性能。

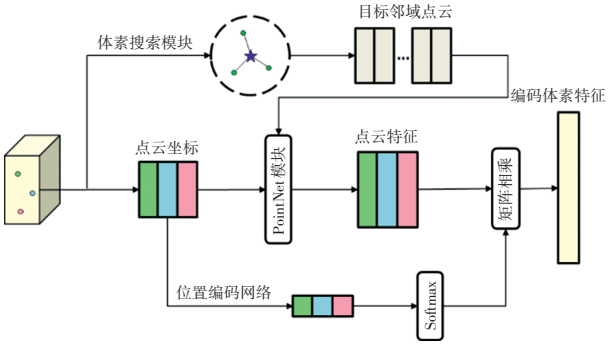


图2 改进的点云编码网络结构

2.2.1 点云结构映射模块

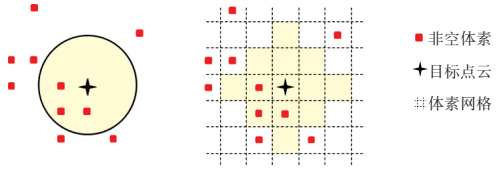
针对给定的点云输入数据，体素特征编码网络首先将点云空间划分为规则的三维体素网格。具体来说，假设空间沿x轴、y轴和z轴的点云范围为(w,h,d)，体素单元尺寸为(v_w,v_h,v_d)，则划分后的三维体素网格分辨率为$(w/v_w,h/v_h,d/v_d)$。完成体素单元划分后，编码网络按照点云坐标将无序点云划分到各体素单元中，由于点云在空间中的分布是稀疏且不均匀的，因此非空体素只占整个体素网格中体素单元的少部分，且各非空体素中包含的点云数量不同。为将体素中包含的不同数量的点云转换为统一维度的特征向量，网络采用随机采样的方法从各非空体素单元中选择固定数量的点云，同时使用

零向量对点云数量少于固定采样数量的体素单元进行补全，这种分组采样的方法不仅可以减少网络计算量，还能够有效缓解点云数据分布不均匀的问题，加快网络训练时的收敛速度。

2.2.2　邻域特征搜索模块

完成点云数据的体素化后，改进的体素特征编码网络使用邻域特征搜索模块为各体素单元内的点云高效搜索邻域点云，从而解决点云体素化过程中因距离较近点云被划分到不同体素单元所产生的离散化信息损失问题，同时使点云特征编码获得更加灵活的感受范围。具体来说，邻域特征搜索模块根据目标点云所处体素单元坐标对距离阈值内的相邻体素单元进行搜索以得到邻域点云集合，随后，基于K近邻算法从这些邻域点云中选择K个邻域点云作为目标点云的邻域特征，最后，基于PointNet模块聚合目标点云的邻域特征以得到点云的编码特征。

邻域特征搜索模块基于体素搜索算法获取指定距离阈值内的点云特征，该算法最早来自体素区域卷积神经网络（Voxel Region-based Convolutional Neural Network，Voxel R-CNN）[11]，如图3所示。



（a）球状搜索　　（b）体素搜索

图3　球状搜索和体素搜索算法对比

体素搜索算法利用体素坐标的规则性在常数时间复杂度下找到目标点云周围的相邻点云。不同于球状搜索算法逐点计算各点云与目标点云之间的欧式距离以得到一组邻域点云，体素搜索算法通过向目标点云所处体素单元的坐标(i,j,k)添加一组三元偏移量$(\Delta i,\Delta j,\Delta k)$以查询相邻体素单元内是否包含点云，并将其中非空体素单元内的点云作为候选的邻域点云集合。体素搜索算法使用曼哈顿距离计算目标体素与相邻体素之间的距离：

$$D_{\mathrm{m}}(\alpha,\beta)=|i_\alpha-i_\beta|+|j_\alpha-j_\beta|+|k_\alpha-k_\beta| \tag{1}$$

式中：$D_{\mathrm{m}}(\alpha,\beta)$为目标体素$\alpha=(i_\alpha,j_\alpha,k_\alpha)$与相邻体素$\beta=(i_\beta,j_\beta,k_\beta)$间的距离。

与直接从全部点云中搜索相邻点云的球状搜索算法相比，体素搜索算法能够充分利用体素单元在三维空间中规则排布的特点，从而在保证算法运行效率的同时实现跨体素的点云信息交互。

2.2.3　体素特征编码网络

完成点云局部特征提取后，体素特征编码网络使用多层感知机将点云特征映射到更高维的空间，并结合各点云相对体素中心的坐标的位置编码信息聚合体素单元内的全部采样点云特征。传统的体素特征编码网络使用最大池化对称函数聚合点云特征，但这种方法忽视了点云特征之间的相互关系而造成点云信息丢失。为了解决此问题，改进的体素特征编码网络对点云位置编码信息进行Softmax归一化以获得各点云特征的权重系数，最后通过加权求和的方式聚合点云特征。改进的体素特征编码网络仍然仅使用了高度优化的前馈神经网络，避免了使用卷积神经网络[12]、图神经网络[13]及自注意力机制[14]等方法探索点云局部集合信息，这保证了编码网络在车载环境等嵌入式平台上的运行效率。

2.3　目标检测网络

完成体素特征编码后，目标检测网络在鸟瞰特征图上提取体素特征并生成三维检测结果。目标检测网络的结构如图4所示，网络主要由金字塔特征提取编码网络和目标检测解码网络组成。目标检测网络的编码部分由堆叠的稀疏卷积模块和子流形卷积模块组成：稀疏卷积模块通过卷积核大小为3×3、步长为2的稀疏卷积层对体素特征进行下采样；子流形卷积模块由卷积核大小为3×3、步长为1的稀疏子流形卷积层对体素特征进行深度语义信息提取。目标检测网络的解码部分采用转置稀疏卷积对不同阶段的特征图上采样到相同分辨率后进行融合，并最终将稀疏体素特征投影到鸟瞰特征图上生成三维候选区域。

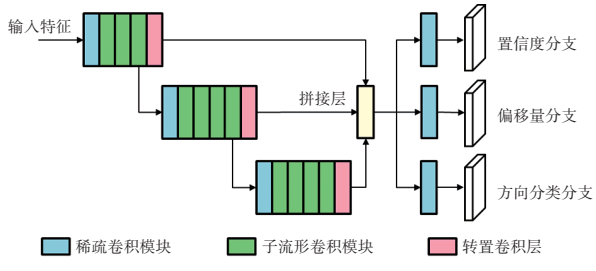


图4　目标检测网络示意

2.3.1　目标检测编码网络

目标检测网络包含3个检测分支，分别对候选框的类别标签、回归偏移量和方向进行预测。目标检测网络在鸟瞰特征图上均匀放置一系列固定尺寸和方向的三维锚框，然后根据检测框真值与预设锚框之间的交并比（Intersection of Union，IoU）对正样本锚框进行分配和编码。

目标在图像中的大小会因为距离的不同而发生变化，因此，目标检测网络需要使用不同大小的锚框来适应不同大小的目标，并对这些锚框进行分类和回

归，以判断它们是否包含感兴趣目标。在三维目标检测任务中，目标的大小不再受距离因素的影响，因此，每个检测类别具有相对固定的三维尺寸，目标检测网络可以根据检测目标的实际尺寸进行锚框的设置。提出的目标检测网络根据KITTI数据集中各类别的标注真值的平均高度和尺寸选择固定的锚框参数，并在共享的鸟瞰特征图上密集设置一系列锚框以实现对感兴趣区域的生成。同时，为兼顾不同朝向的三维目标，每个锚框的方向设置为0°和90°，以更好地对目标候选框进行回归。各检测类别的锚框参数设置如表1所示。

表1　锚框参数设置

检测类别	锚框尺寸	中心高度/m
车辆	(3.9 m,1.6 m,1.56 m)	−1.0
行人	(0.8 m,0.6 m,1.73 m)	−0.6
骑行者	(1.76 m,0.6 m,1.73 m)	−0.6

完成锚框参数确定后，需要对特征图上密集的锚框进行正样本匹配。当锚框与检测框真值的IoU大于匹配阈值时，锚框被分配为正样本；当IoU小于非匹配阈值时，锚框被分配为负样本；其他样本在网络训练过程中被忽略。各检测类别的匹配阈值和非匹配阈值参数如表2所示。

表2　正样本分配参数设置

检测类别	匹配阈值	非匹配阈值
车辆	0.6	0.45
行人	0.5	0.35
骑行者	0.5	0.35

2.3.2　目标检测解码网络

每个成功匹配的正样本锚框会被分配到一个表示目标类别的独热向量、一个代表检测框三维参数的7维向量和一个表示检测框方向分类的独热向量。网络通过编码函数将检测框真值转换为相对锚框的偏移值，从而减小网络预测结果的波动范围以加快网络收敛。网络编码函数的计算过程如下：

$$\begin{cases} x_t = \dfrac{x_{gt} - x_a}{d_a} \\ y_t = \dfrac{y_{gt} - y_a}{d_a} \\ z_t = \dfrac{z_{gt} - z_a}{d_a} \end{cases} \quad \begin{cases} w_t = \log\left(\dfrac{w_{gt}}{w_a}\right) \\ l_t = \log\left(\dfrac{l_{gt}}{l_a}\right) \\ h_t = \log\left(\dfrac{h_{gt}}{h_a}\right) \end{cases} \quad \theta_t = \sin(\theta_{gt} - \theta_a) \tag{2}$$

式中：(x_t,y_t,z_t)为检测框中心坐标相对于锚框的偏移量，(x_{gt},y_{gt},z_{gt})为检测框真值的中心坐标，(x_a,y_a,z_a)为锚框的中心坐标，w_t、l_t、h_t分别为检测框长度、宽度和高度相对于锚框的偏移量，w_{gt}、l_{gt}、h_{gt}分别为检测框真值的长度、宽度和高度，w_a、l_a、h_a分别为锚框的长度、宽度和高度，θ_t、θ_{gt}分别为检测框绕z轴航向角的偏移量和航向角真值，θ_a为锚框绕z轴的航向角，$d_a = (l_a^2 + w_a^2)^{1/2}$为锚框底平面的对角线长度。

2.4　损失函数设计

目标检测网络使用多任务损失函数进行优化，损失函数由3个部分组成：

$$L_{loss} = \alpha L_{cls} + \beta L_{reg} + \gamma L_{dir} \tag{3}$$

式中：L_{loss}为总损失函数，L_{cls}为分类损失函数，L_{reg}为回归损失函数，L_{dir}为方向损失函数，α=1.0、β=2.0、γ=0.2为权重系数。

2.4.1　分类损失函数

网络训练时产生的大量锚框中只有极少部分是正样本，导致正负样本不平衡问题严重，这使得网络偏向于将锚框预测为背景。为了解决此问题，RetinaNet[15]使用了焦点损失函数，函数通过可调参数来减小简单样本所占的权重，使模型更加关注难分类样本，进而提升分类准确性。本文同样使用焦点损失函数进行分类网络的优化，其计算过程为：

$$L_{cls} = -\alpha_t (1 - p_t)^{\gamma_t} \log(p_t) \tag{4}$$

式中：p_t为网络对样本属于正确类别的预测概率，α_t=0.25、γ_t=2.0分别为焦点损失函数的权重系数和调节因子。

2.4.2　回归损失函数

为了使训练更快收敛，使用平均绝对值误差损失函数计算网络预测值与检测框真值之间的回归损失，回归目标包含检测框中心位置的偏移量、检测框的长度、宽度、高度和航向角，回归损失函数为：

$$\begin{aligned} L_{reg} = {} & L_1(x_t) + L_1(y_t) + L_1(z_t) + L_1(w_t) + \\ & L_1(l_t) + L_1(h_t) + L_1(\theta_t) \end{aligned} \tag{5}$$

式中：$L_1(n) = \begin{cases} 0.5n^2, & |n| \le 1 \\ |n| - 0.5, & \text{其他} \end{cases}$为平均绝对误差。

2.4.3　方向损失函数

回归网络通常直接使用网络的预测值与真实值之间的误差作为优化目标，但当对检测框方向的预测值与真实值偏移接近180°时会产生较大误差，这不利于网络的优化。为了解决此问题，网络以朝向角的正弦值作为回归目标，同时添加一个方向分类网络来区分相反朝向的检测框，其中朝向角大于0°的检测框被划分为正样本，小于0°的检测框被划分为负样本，网络使用Softmax交叉熵损失函数对方向分类网络进行优化：

$$L_{\mathrm{dir}} = y\ln(a_{\mathrm{pre}}) + (1-y)\ln(1-a_{\mathrm{pre}}) \quad (6)$$

式中：a_{pre}为网络输出经过Softmax归一化后的预测值，y为检测框方向类别。

3 试验与结果分析

本文提出的算法基于PyTorch深度学习框架及OpenPCDet三维目标检测代码库开发，模型的训练和评估均在一台搭载英伟达Titan RTX显卡的本地服务器上进行，服务器具体参数和运行环境如表3所示。

表3 服务器参数设置及运行环境

系统	Linux Ubuntu18.04
中央处理器型号	Intel i9-9920X
内存/GB	64
图形处理器型号	NVIDIA TITAN RTX
Pytorch版本	1.8.0
CUDA版本	11.1
cuDNN版本	8.0

使用基于梯度下降的Adam优化算法进行端到端训练，固定权重衰减率设置为0.01。网络的学习率采用单周期策略，最大学习率设置为0.003。算法基于KITTI数据集中公开的训练子集进行训练和评估，共包含7 481个样本。训练子集划分为2个部分，其中3 712个样本用于训练网络，另外3 769个样本用于评估网络性能。网络共训练200个周期，训练批次大小设置为4。在使用网络进行推理时，非极大值抑制的IoU阈值设置为0.01，评分阈值设置为0.1。

将所提出的三维目标检测网络与其他先进的三维目标检测网络在KITTI数据集上进行对比。评估结果中的检测目标包含车辆、行人和骑行者3个类别，每个类别按照目标在相机平面中的高度、遮挡程度和截断属性分为简单、中等和困难3个检测难度等级，如表4所示。

表4 KITTI数据集样本难度划分准则

样本难度	最小检测框高度/像素	最大遮挡级别	最大截断程度/%
简单	40	完全可见	15
中等	25	部分遮挡	30
困难	25	难以看见	50

参与比较的目标检测算法包括多视图三维目标检测网络(Multi-View 3D Networks，MV3D)[16]、面向三维目标识别的像素级预测神经网络(ORiented 3D object detection from PIXel-wise neural network predictions，PIXOR)[17]、VoxelNet[6]和PointPillars网络[8]。其中MV3D、PIXOR和VoxelNet的评估结果来自相应文献，PointPillars网络的评估结果同样由OpenPCDet代码库提供的模型在本地服务器上测得。本文的改进主要集中于提高算法的检测精度，考虑到算法成熟度和研究的关联性，通过继续使用PointPillars作为基准，可以直接展示出改进点对于原有技术框架的具体增益。使用平均精度(Average Precision，AP)作为评价算法检测性能的指标[2]，各算法的评估结果如表5所示。

表5 KITTI评测集上本文方法与其他方法性能比较结果 %

类别	方法	鸟瞰视角AP			3D检测AP		
		容易	中等	困难	容易	中等	困难
车辆	MV3D[16]	86.18	77.32	76.33	71.19	56.60	55.30
	PIXOR[17]	86.79	80.75	76.60			
	VoxelNet[6]	89.60	84.81	78.57	81.97	65.46	62.85
	PointPillars[8]	90.07	87.44	84.78	85.22	76.52	69.73
	本文方法	89.21	87.01	85.21	86.25	77.08	75.75
行人	PointPillars[8]	61.63	56.27	52.60	57.75	52.29	47.91
	本文方法	64.98	58.88	54.65	60.30	55.00	49.98
骑行者	PointPillars[8]	82.26	66.11	62.56	80.05	62.61	59.53
	本文方法	84.44	67.94	64.53	82.62	64.21	60.89

由表5可知，本文提出的方法在车辆类别的困难难度和行人、骑行者类别的各难度上均取得了较明显的提升，这表明本文提出的方法能够更加准确地捕获目标的特征，尤其对因遮挡或截断而丢失部分特征的困难样本具有较明显的提升效果，这证明了本文提出方法的有效性。相对于传统PointPillars网络，本文方法在检测困难等级的车辆类别时，鸟瞰视角(Bird Eye View，BEV)下的平均精度和3D检测的平均精度分别提升了0.43百分点和6.02百分点，在简单和中等难度下，本文方法的平均精度与对应的最好方法基本持平；在行人类别上，本文方法在各难度等级上均取得了最好的效果，其中在鸟瞰视角简单、中等和困难难度下平均精度分别提升了3.35百分点、2.61百分点和2.05百分点，3D检测的3个难度下平均精度分别提升了2.55百分点、2.71百分点和2.07百分点；在骑行者类别上，本文方法在简单难度下取得了最好的效果，在鸟瞰视角下和3D检测中分别提升了3.18百分点和2.57百分点。

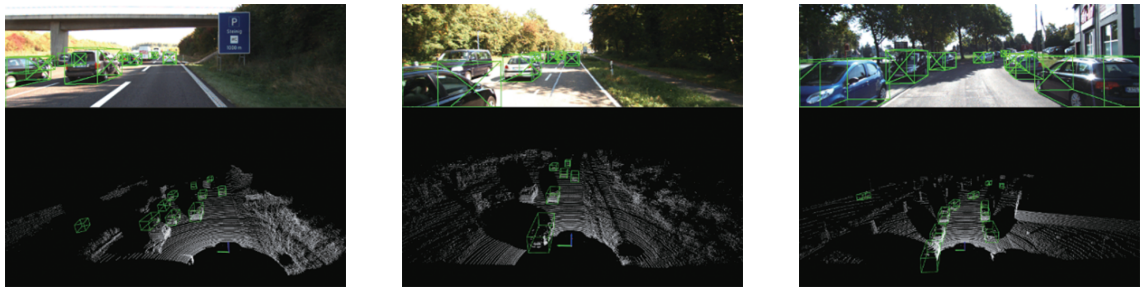
在推理速度方面，测试结果如表6所示，本文模型对单帧点云数据的推理时间为33.79 ms，相比基线网络PointPillars增加了6.68 ms。由于在基线网络中引入了特征提取与融合等改进模块，不可避免地增加了模型的

复杂度，从而导致其推理时间有所延长。

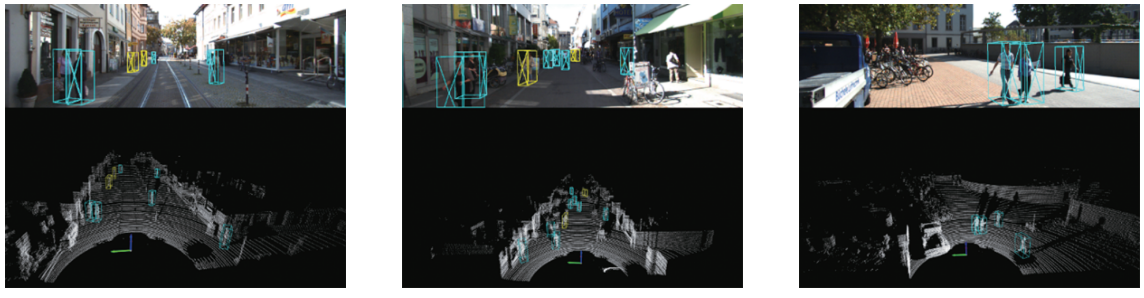
表6　模型推理时间　　ms

模型选择	推理时间
PointPillars	27.11
本文方法	33.79

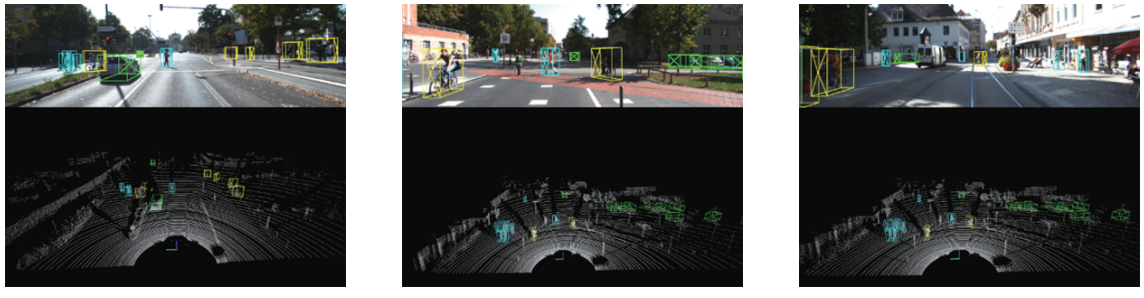
在KITTI数据集上对本文所提出的三维目标检测网络的检测结果进行了可视化分析，如图5所示。检测结果覆盖了多种场景，场景中的车辆、行人和骑行者分别使用绿色、蓝色和黄色的三维检测框表示，这些三维检测框同时被投影到相机图像上。



(a)车辆



(b)行人



(c)骑行者

图5　KITTI数据集的可视化检测结果

由图5可以看出：本文网络能够在各种场景下实现对车辆的精准检测，包括高速公路上密集拥挤的车辆、道路两侧停放的车辆以及对向车道驶来的车辆；网络对两侧人行道行走的行人、车辆前方正在横穿马路的行人以及站在车辆旁的行人均能实现准确的检测，同时对距离自车较远的行人也能实现较好的检测效果；网络对同向行驶、对向行驶、横穿马路以及推自行车的骑行者均能实现较为准确的检测。

4　结束语

本文对当前基于体素的点云特征编码网络的优势和局限进行了分析，在PointPillars网络的基础上提出了一种考虑点云结构信息和相互关系的体素特征编码网络，网络仅使用前馈神经网络和对称函数分层地聚合体素化后的点云特征，并且较好地保留了体素特征编码网络的高效性，并提出了一种基于稀疏卷积的三维目标检测网络，对编码后的点云特征进行特征提取和候选框生成，先使用体素特征编码网络将输入点云数据编码为稀疏体素特征，随后使用基于金字塔特征的特征提取网络将其转换为鸟瞰特征图，并在特征图上生成检测目标的候选检测框。KITTI数据集上的试验结果表明，本文提出的算法相对于现有算法优化效果显著，相对于传统PointPillars网络，在检测困难等级的车辆类别时，鸟瞰视角下平均精度和3D检测的平均精度分别提升了0.43百分点和6.02百分点，在行人和骑行者类别上，在各难度等级下均取得了最好的效果，能够提升自动驾驶场景

中三维目标检测算法的检测精度。

本文只在KITTI数据集上进行了测试与对比分析，为体现算法的普适性和有效性，未来可在其他认可度高的数据集上进行测试与验证，探索在不牺牲检测精度的前提下进一步优化模型的推理速度，以更好地适应实际应用需求。

参 考 文 献

[1] 王海, 徐岩松, 蔡英凤, 等. 基于多传感器融合的智能汽车多目标检测技术综述[J]. 汽车安全与节能学报, 2021, 12(4): 440-455.
WANG H, XU Y S, CAI Y F, et al. Overview of Intelligent Vehicle Multi-Target Detection Technology Based on Multi-Sensor Fusion[J]. Journal of Automotive Safety and Energy, 2021, 12(4): 440-455.

[2] 程鑫, 王宏飞, 周经美, 等. 基于体素柱形的激光雷达点云车辆目标检测算法[J]. 中国公路学报, 2023, 36(3): 247-260.
CHENG X, WANG H F, ZHOU J M, et al. Vehicle Detection Algorithm Based on Voxel Pillars from LiDAR Point Clouds[J]. China Journal of Highway and Transport, 2023, 36(3): 247-260.

[3] SHI S S, WANG X G, LI H S. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA: IEEE, 2019: 770-779.

[4] YANG Z T, SUN Y N, LIU S, et al. STD: Sparse-to-Dense 3D Object Detector for Point Cloud[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Korea: IEEE, 2019: 1951-1960.

[5] LIAO W H, WANG C C, LIN W C. GNN-Based Point Cloud Maps Feature Extraction and Residual Feature Fusion for 3D Object Detection[C]// Proceedings of the IEEE International Conference on Robotics and Automation. London, United Kingdom: IEEE, 2023: 7010-7016.

[6] ZHOU Y, TUZEL O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018: 4490-4499.

[7] YAN Y, MAO Y X, LI B. SECOND: Sparsely Embedded Convolutional Detection[J]. Sensors, 2018, 18(10).

[8] LANG A H, VORA S, CAESAR H, et al. PointPillars: Fast Encoders for Object Detection from Point Clouds[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA: IEEE, 2019: 12697-12705.

[9] QI C R, SU H, MO K C, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017: 652-660.

[10] ZHANG H, LUO G Y, WANG X, et al. SASAN: Shape-Adaptive Set Abstraction Network for Point-Voxel 3D Object Detection[J]. IEEE Transactions on Neural Networks and Learning Systems, 2025, 36(2): 2465-2479.

[11] DENG J J, SHI S S, LI P W, et al. Voxel R-CNN: Towards High Performance Voxel-Based 3D Object Detection[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(2): 1201-1209.

[12] LI Y Y, BU R, SUN M C, et al. PointCNN: Convolution on X-Transformed Points[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, Canada: Curran Associates Inc., 2018.

[13] LI G H, MÜLLER M, QIAN G C, et al. DeepGCNs: Making GCNs Go as Deep as CNNs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(6): 6923-6939.

[14] GUO M H, CAI J X, LIU Z N, et al. PCT: Point Cloud Transformer[J]. Computational Visual Media, 2021, 7(2): 187-199.

[15] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal Loss for Dense Object Detection[C]// Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy: IEEE, 2017: 2980-2988.

[16] CHEN X Z, MA H M, WAN J, et al. Multi-View 3D Object Detection Network for Autonomous Driving[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017: 1907-1915.

[17] YANG B, LUO W J, URTASUN R. PIXOR: Real-Time 3D Object Detection from Point Clouds[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018: 7652-7660.

（责任编辑 斛 畔）

修改稿收到日期为2025年10月9日。