基于点云的自动驾驶下三维目标检测

杨咏嘉，钟良琪，闫胜业+

（南京信息工程大学 自动化学院，江苏 南京 210044）

摘　要：针对当前三维目标检测算法对行人、骑行人等小目标检测效果不佳的缺点，提出一种改进 PV-RCNN 的三维目标检测算法。改进关键点下采样方式，通过滤除背景及离群点提高关键点在目标上的命中率；设计多尺度区域建议网络，尺度匹配的特征图提高边界框的生成质量；使用加入方向感知的 DIoU 损失函数优化边界框的回归。实验结果表明，与基准网络相比，算法在 KITTI 测试集的车辆、行人和骑行人的 mAP 分别提高了 0.77%、6.33%和 2.05%，有效提高了网络性能。

关键词：深度学习；三维目标检测；特征金字塔；原始点云；交并比损失函数；特征融合；点云下采样

中图法分类号：TP391.4　　**文献标识号**：A　　**文章编号**：1000-7024（2024）04-1093-07

doi：10.16208/j.issn1000-7024.2024.04.019

3D object detection in automatic driving based on point cloud

YANG Yong-jia，ZHONG Liang-qi，YAN Sheng-ye+

（School of Automation，Nanjing University of Information Science and Technology，Nanjing 210044，China）

Abstract：Aiming at the disadvantage of current 3D target detection algorithm for small targets such as pedestrian and cyclists，a 3D object detection algorithm based on improved PV-RCNN was proposed. The method of key points sampling was improved，and the hit rate of key points on the target was improved by filtering the background and outliers. A multi-scale region proposal network was designed. The feature map of scale matching improved the quality of proposal box generation. The DIoU loss function with direction perception was used to optimize the regression of the bounding box. Experimental results show that compared with the benchmark network，the mAP of car，pedestrian and cyclist in the KITTI val set of the algorithm is improved by 0.77%，6.33% and 2.05% respectively，effectively improving the network performance.

Key words：deep learning；3D object detection；feature pyramid network；raw point cloud；IoU loss；feature fusion；point cloud downsampling

0　引　言

三维目标检测作为计算机视觉任务的基础课题之一，在自动驾驶和机器人视觉领域有着广泛的应用。尤其在自动驾驶领域，如何精确检测出车辆周围的三维目标，是一项富有意义的任务。目前主流的三维目标检测算法采用的输入数据有单目图像、RGB-D 图像和点云数据[1]。与 RGB 图像相比，点云数据几何信息丰富、对应用环境具有较强鲁棒性[2]。因此为了应对复杂多变的现实交通场景，点云数据通常是检测算法的首选。

近年来，以点云作为输入，使用深度学习技术的三维目标检测算法因其具有良好的学习与泛化能力受到广泛关注。作为开创性的工作，VoxelNet[3]将点云通过网格进行体素化，大幅降低了网络的计算量。PointNet[4]则设计了一种全对称网络，解决了将原始点云直接输入时点云的无序性带来的检测结果不稳定问题。PV-RCNN[5]则将两者的优势进行结合，以体素化点云作为主要输入，并且将部分原始点云的特征一同送入边界框微调过程，提高了体素化方法的特征表达能力。

然而，PV-RCNN 的体素特征图分辨率过低，以及均匀的点云下采样，导致算法在行人等小目标上检测效果不佳。因此，我们提出多尺度区域建议网络以及非均匀的精

收稿日期：2022-10-26；**修订日期**：2024-04-07

基金项目：国家自然科学基金项目（61300163）

作者简介：杨咏嘉（1996-），男，浙江杭州人，硕士研究生，CCF 学生会员，研究方向为三维目标检测与深度学习；钟良琪（1997-），男，安徽马鞍山人，硕士研究生，研究方向为目标检测与深度学习；+通讯作者：闫胜业（1978-），男，河南新乡人，博士，教授，研究方向为计算机视觉、模式识别和机器学习。E-mail：956750373@qq.com

确关键点下采样的三维目标检测算法（RPV-RCNN），为不同尺度的目标生成更适配的特征图，同时有针对性地对点云进行下采样，使得算法在小目标检测性能上有较大提升。

1 相关工作

基于点云的三维目标检测算法可以划分为 3 类：基于点的方法、基于体素的方法和基于点和体素融合的方法。基于点的方法通常采用 PointNet 或 PointNet++[6] 作为骨干网络对原始点云进行多层次的局部特征提取。原始点云具有丰富的位置信息，因此这类算法提取出的特征包含了更多的几何信息。然而由于要多次对大量的点进行特征提取，导致算法需要消耗大量的计算资源。典型的方法有 PointRCNN[7]、STD[8]、3DSSD[9] 等。基于体素的方法通过将原始点云划分为规则大小的网格，即体素。对体素内的点进行特征提取来降低点云的分辨率，实现计算量的降低。提取后的体素通过 3D 稀疏卷积进行进一步的特征提取，由于点云的稀疏性，3D 稀疏卷积仅需要提取少量的非空体素的特征，从而大幅度地提高了检测效率。然而体素化损失的几何信息是不可逆的，导致此类方法的检测精度受到一定限制。此类经典方法有 VoxelNet、SECOND[10]、CIA-SSD[11] 等。基于点和体素融合的方法融合了两者的优势，通过对原始点云下采样得到关键点进行特征提取并将其与对应位置的体素特征进行融合，一定程度上弥补了体素特征的几何信息缺失问题，从而带来了更好的检测效果。此类经典方法有 PV-RCNN、Point-Voxel CNN[12]、SA-SSD[13] 等。

综上这些方法采用了不同的方式对点云进行特征提取，其中基于点和体素融合的方法在检测速度与精度上达到较好的平衡，是基于点云的三维目标检测任务中理想的网络结构。

2 网络模型

2.1 网络设计

基于 PV-RCNN 在特征提取及边界框微调阶段的良好性能，本文延用了该算法的 3D 稀疏卷积层、关键点特征融合层、关键点预测层和边界框微调层。并将其原有的关键点采样、区域建议网络以及损失函数改进为新设计的精确关键点采样、多尺度区域建议网络和基于 DIoU 的损失函数，以此提出了 RPV-RCNN 算法。

RPV-RCNN 算法网络结构如图 1 所示，首先将一帧原始点云体素化后输入至 3D 稀疏卷积层，在输出时将 z 维特征与通道特征进行叠加得到输出尺寸为（C，H，W）的鸟瞰图（BEV）特征。同时，将原始点云通过精确关键点采样，得到由 k 个关键点。将鸟瞰图特征经过 2D 卷积分别得到尺寸为（2C，H/2，W/2），（4C，H/4，W/4）的多尺度特征图，送入多尺度区域建议网络后得到初步的边界框。与此同时，另一个分支将采样得到的 k 个关键点映射回 3D 稀疏卷积不同尺度的特征图以及鸟瞰图特征中以提取不同尺度的关键点特征，将多尺度特征拼接并送入关键点预测层后得到的关键点特征与边界框送入边界框微调层进一步精确地回归以得到最终的边界框。

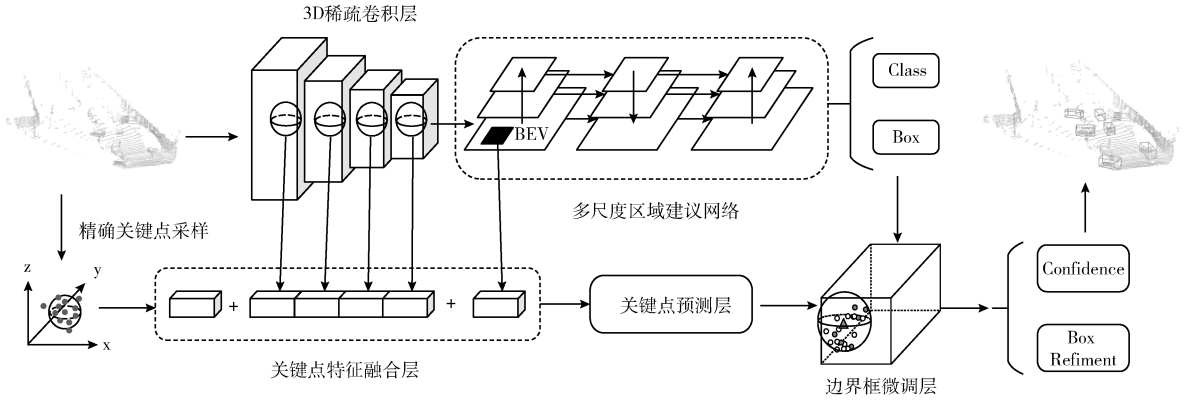


图 1 RPV-RCNN 网络结构

2.2 精确关键点采样

本文设计了精确关键点采样算法，相较于 PV-RCNN 直接使用最远点采样，我们在采样前删除部分无关的背景点，从而提高关键点在目标上的命中率。首先，为了去除地面上的点云，在原始点云中随机选取 3 个点并拟合出平面 p，统计平面 p 及与平面一定误差范围内包含的点的数量并多次迭代，直到查找到包含最多点的平面，则认为该平面为地面并将其从原始点云中去除。随后，为了过滤离群点，采用统计滤波对点云进行过滤。统计滤波算法首先计算点云中的每个点 x_i 到 k 个邻近点的距离 d_{ij} 并将其存放在集合 D 中，计算过程如式（1）所示

$$d_{ij} = x_i - x_j$$
$$D = \{d_{i1}, d_{i2}, \cdots, d_{ik}\} \tag{1}$$

随后计算集合 D 的均值 d_i 作为点 x_i 的特征，计算过程

如式（2）所示

$$d_i = \frac{\sum_{j=1}^{k} d_{ij}}{k} \tag{2}$$

通过统计点云中所有点的距离均值可以得到点云的均值 μ 和方差 σ^2 特征，从而获得该点云的概率分布以及筛选阈值 t。计算过程如式（3）所示

$$\mu = \frac{\sum_{i=1}^{N} d_i}{N}$$

$$\sigma^2 = \frac{\sum_{i=1}^{N} (d_i - \mu)^2}{N}$$

$$t = \mu + \sigma^2 \tag{3}$$

式中：N 表示点云包含的点的数量。如果点 x_i 的距离均值 d_i 大于 t，则被判断为离群点并删除。本算法 k 设置为 50，最终的过滤效果对比如图 2 所示，其中左边为原始点云，右边为过滤后点云。最后通过最远点采样对过滤后的点云进行采样得到关键点。

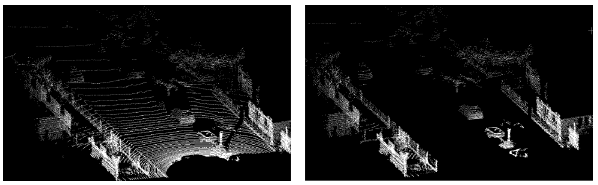


图 2　过滤前后原始点云对比

2.3　多尺度区域建议网络

特征金字塔网络（feature pyramid network，FPN）[14] 作为目标检测网络的重要组成部分之一已被广泛使用。FPN 具有两个显著优势：多尺度特征融合和分治策略。其中前者受到高度重视，然而通过解耦两者的实验[15]结果表明，分治策略相较于多尺度特征融合的作用更为突出。因此，我们使用双聚合金字塔网络结构[16]并加入权重融合构建了多尺度区域建议网络。双聚合金字塔网络构建了双向信息路径，其中自顶向下的信息路径将高层的强语义特征向下传递，而自底向上的信息路径则可以将底层的强定位特征向上传递，从而丰富每一个尺度的特征图的语义与定位特征信息。同时，引入短链接思想[17]，添加了从原始输入到同层级输出的短链接。双聚合金字塔网络结构如图 3 所示，其中 P_1^{in}、P_2^{in} 和 P_3^{in} 为对 BEV 特征图进行卷积得到的不同尺度特征图，P_2^{td} 为特征金字塔自顶向下（top-down）特征传递中的中间特征层，P_1^{out}、P_2^{out} 和 P_3^{out} 分别表示特征金字塔的最终输出层。

同时为了反映不同特征层在特征融合中的重要性，我们通过向特征添加额外权重来实现这一点，融合过程计算公式如式（4）所示

$$f_{fused} = \sum_i \frac{w_i}{\sum_j w_j + \epsilon} \cdot f_i \tag{4}$$

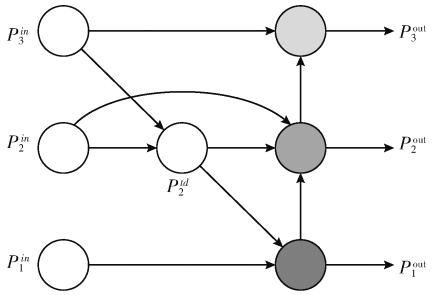


图 3　双聚合金字塔网络结构

其中，w_i 是一个可学习的权重，以反映不同输入的重要性，f_i 表示输入该层的其它特征层，i 表示输入特征层个数，f_{fused} 表示该层经过融合过后的特征。为防止训练不稳定，我们采用权重归一化来限制每个权重的值范围。$\epsilon = 0.0001$，用于避免数值不稳定。

通过集成双向跨尺度连接和归一化权重融合，我们的双聚合金字塔可以在第 2 级用式（5）描述

$$P_2^{td} = Conv\left(\frac{w_1 \cdot P_2^{in} + w_2 \cdot Resize(P_3^{in})}{w_1 + w_2 + \epsilon}\right)$$

$$P_2^{feature} = \frac{w_1' \cdot P_2^{in} + w_2' \cdot P_2^{td} + w_3' \cdot Resize(P_1^{out})}{w_1' + w_2' + w_3' + \epsilon}$$

$$P_2^{out} = Conv(P_2^{feature}) \tag{5}$$

其中，w_1 和 w_2 分别表示 P_2^{in} 和 P_3^{in} 在输入 P_2^{td} 进行融合时网络根据式（4）分配给两者的权重，w_1'、w_2' 和 w_3' 分别表示 P_2^{in}、P_2^{td} 和 P_1^{out} 在特征融合时网络根据式（4）分配给三者的权重，$P_2^{feature}$ 表示该层经过融合后的特征层，最终通过卷积得到 P_2^{out} 作为该层输出的同时将其输入更大感受野的上一层。其它输出层的特征以类似的方式构造。最终我们在 P_1^{out}、P_2^{out}、P_3^{out} 分别放置适配行人、骑行人以及汽车尺寸的锚框，并将结果发送到检测头得到初步的边界框。

2.4　损失函数

RPV-RCNN 的全局损失函数由多尺度区域建议网络损失、边界框微调层损失以及关键点分类损失组成，计算公式如式（6）所示

$$L = L_{rpn} + L_{rcnn} + L_{k-point} \tag{6}$$

（1）多尺度区域建议网络损失

多尺度区域建议网络损失 L_{rpn} 由分类损失和回归损失两部分组成。分类损失采用交叉熵损失函数计算，计算公式如式（7）所示

$$L_{cls} = -\sum_i y_g \ln(y_p) + (1 - y_g)\ln(1 - y_p) \tag{7}$$

其中，y_g 表示真实分类标签，y_p 表示神经网络预测的分类结果，i 表示 anchor 中被判定为正样本的数量。

回归损失相较于 PV-RCNN 网络的 smooth-L1 损失改为加入方向感知的 DIoU[18] 损失。DIoU 损失可以最小化预测框与真值的中心不对齐问题，而方向感知可以进一步减

小两者的朝向差别。损失函数具体计算公式如式（8）所示

$$L_{reg}=\sum_i 1-\mathrm{IoU}(B_p,B_g)+\frac{c^2}{d^2}+\beta(1-|\cos(\Delta r)|) \quad (8)$$

其中，B_p 表示神经网络预测的边界框，B_g 表示预测的边界框对应的真值标签，c 与 d 分别表示两者的最小闭包区域 ABCD 的对角线距离和两个框中心点 O_1 与 O_2 的欧氏距离，几何如图 4 所示。Δr 表示 B_p 与 B_g 在 BEV 下的角度差值，通过 $1-|\cos(\Delta r)|$ 函数可以快速帮助预测框角度进行回归。β 作为调节角度差值在回归损失中占比的权重，在本网络中取 $\beta=2$。

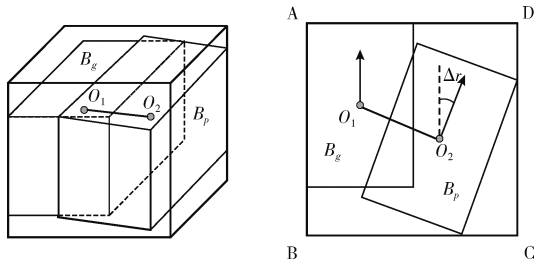


图 4 3D 边界框 IoU

（2）边界框微调层损失

边界框微调层损失 L_{rcnn} 由置信度损失和回归损失两部分组成。对于置信度采用交叉熵损失，计算公式如式（9）所示

$$y_k=\min(1,\max(0,2IoU_k-0.5))$$
$$L_{confidence}=-\sum_i y_k\ln(\widetilde{y_k})+(1-y_k)\ln(1-\widetilde{y_k}) \quad (9)$$

其中，y_k 表示网络的训练目标，由边界框与真值标签的 IoU 得到，$\widetilde{y_k}$ 为网络预测的边界框置信度。回归损失采用和 L_{rpn} 一致的加入方向感知的 DIoU 损失。

（3）关键点分类损失

由于前景点与背景点数量相差巨大，因此采用 focal loss 作为关键点分类损失以平衡正负样本不均衡现象，计算公式如式（10）所示

$$L_{k-point}=-\alpha_t(1-p_t)^\gamma\ln(p_t)$$
$$p_t=\begin{cases}p & \text{if } y=1\\ 1-p & \text{otherwise}\end{cases}$$
$$\alpha_t=\begin{cases}\alpha & \text{if } y=1\\ 1-\alpha & \text{otherwise}\end{cases} \quad (10)$$

其中，p 为网络输出的关键点置信度，y 为关键点的真值标签，通过统计关键点是否在边界框的真值标签内得到。α、γ 分别为调节正负样本以及难易样本数量失衡的权重，在本网络中取 $\alpha=0.25$，$\gamma=2$。

3 实验与分析

3.1 实验数据集

本文的所有实验与测试评估都是在 KITTI 数据集上进行的，该数据集在三维目标检测领域被广泛使用。KITTI 数据集包含 7481 个训练样本和 7528 个测试样本，在评估时对数据集分别对 car 类、cyclist 类和 pedestrian 类进行评估。由于点云数据中的目标存在被遮挡或者截断的情况，KITTI 数据集据此将检测目标划分为容易、中等和困难 3 种难度级别。

实验评估指标采用均值精度（average precision，AP）以及平均均值精度（mean average precision，mAP）。对于同一类别的不同难度采用 AP 进行评估，计算方式为查准率（precision）与查全率（recall）构成的曲线下的面积。查准率与查全率定义如式（11）所示

$$\mathrm{precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$
$$\mathrm{recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \quad (11)$$

其中，TP(ture positive) 指原本为正类且被划分为正类的样本；FP(false positive) 指原本为负类且被划分为正类的样本；FN(false negetive) 指原本为正类且被划分为负类的样本。对于不同类别目标命中定义设置不同的 IoU 阈值，car 类设置为 0.7，cyclist 类和 pedestrian 类设置为 0.5。算法在每个类别的 mAP 通过统计该类所有难度的 AP 并计算其均值得到。

3.2 实验环境

本文使用的硬件平台为 Intel(R) Core(R) 8700k CPU，NVIDIA RTX 2080ti 显卡，操作系统为 Ubuntn 16.04LTS。训练过程采用 Adam 优化器进行参数更新，初始学习率设置为 0.001，衰减因子为 0.8，每 15 个周期更新一次。用于平滑累积梯度及平方的衰减速率 β_1 和 β_2 分别为 0.9 和 0.999。训练的 epoch 设置为 80，batch_size 设置为 4。

3.3 模型对比与验证实验

在实验中将训练数据集划分为 3712 个训练样本和 3796 个验证样本。本文选择 SECOND、PointPillars[19]、PointRCNN、3DSSD 和 PV-RCNN 作为对比算法，不同算法在 KITTI 数据集 3 个类别的不同难度上的 3D 检测结果见表 1～表 3。每个难度采用 AP 进行评估，并对比了每个类别的 mAP 差异，所有数据以百分比为单位。

表 1 car 类别下不同算法平均精度均值对比

Method	car			
	easy	moderate	hard	mAP/%
SECOND	83.34	72.55	65.82	81.48
PointPillars	82.58	74.31	68.99	79.76
PointRCNN	89.01	78.77	78.10	81.96
3DSSD	88.36	79.57	74.55	80.83
PV-RCNN	92.57	84.83	82.69	86.79
ours	**93.31**	**85.84**	**83.53**	**87.56**

表 2　pedestrian 类别下不同算法平均精度均值对比

Method	car			
	easy	moderate	hard	mAP/%
SECOND	56.00	50.02	43.64	49.89
PointPillars	66.73	61.06	56.50	61.43
PointRCNN	62.69	55.36	51.60	56.55
3DSSD	63.91	**65.77**	51.30	60.33
PV-RCNN	62.32	54.42	49.81	55.52
ours	**67.78**	62.32	**55.44**	**61.85**

表 3　cyclist 类别下不同算法平均精度均值对比

Method	car			
	easy	moderate	hard	mAP/%
SECOND	80.97	63.43	56.67	67.02
PointPillars	83.65	63.40	59.71	68.92
PointRCNN	84.48	65.37	59.83	69.89
3DSSD	88.71	71.37	63.89	74.65
PV-RCNN	90.39	75.71	65.99	77.36
ours	**92.57**	**76.31**	**69.36**	**79.41**

通过对比可以看出，与 PV-RCNN 相比，RPV-RCNN 在 pedestrian 类和 cyclist 类这两个小目标类上的检测性能有显著提升，mAP 分别提高了 6.33%和 2.05%，分析认为是由于精确关键点采样给小目标物体采样到更多关键点以及为小目标物体提供了与其尺寸相匹配的特征图共同造成的结果。在 pedestrian 类的 moderate 难度中检测精度相较 3DSSD 落后 3.45%，分析认为是由于 3DSSD 以原始点云作为输入，经过多层感知机提取后保留了更丰富的几何特征导致。但是本算法在该类的 mAP 优于所有对比实验方法，可以验证算法设计的有效性。在 car 类上的性能提升有限，mAP 相较于 PV-RCNN 提升了 0.77%，分析认为车辆目标较大，在原算法中对其特征提取已经较为充分。总体来说 RPV-RCNN 在 KITTI 数据集的 3 个类别上的检测精度达到了较为优秀的水平。

本文对算法的预测结果与真值标签进行了可视化对比，对比结果如图 5～图 7 所示。图中的图（a）和图（b）表示的是真值标签，图（c）表示的是算法的预测结果，v、p、c 分别表示该框的类别信息为汽车、行人和骑行人。通过图 5 可以分析得出，RPV-RCNN 对车辆目标检测能力良好，所有真值标签标注目标均被成功检测出。通过图 6 可以分析出算法在骑行人方面检测能力良好，检测出了所有目标。通过图 7 可以分析出算法在行人方面检测能力较强，在标注目标全部检出的同时也检测出了部分未标注目标。但是也有部分目标存在误检，分析认为是由于部分稀疏的点云特征较为相似导致。

3.4　消融实验

为了验证精确关键点采样、多尺度区域建议网络以及基于 DIoU 的损失函数对模型检测精度的贡献率，决定对模型进行消融实验。将上述模块分别从算法模型中移除，对网络进行重新训练并观察检测精度的变化。使用 mAP 作为精度衡量指标，使用 KITTI 数据集的测试集作为验证数据集。实验结果见表 4～表 6。

根据实验结果可知，精确关键点采样和多尺度区域建议网络对模型检测性能提升最明显，其中在 pedestrian 类别中提升最大，提升了 4.79%，在 cyclist 和 car 类别中分别

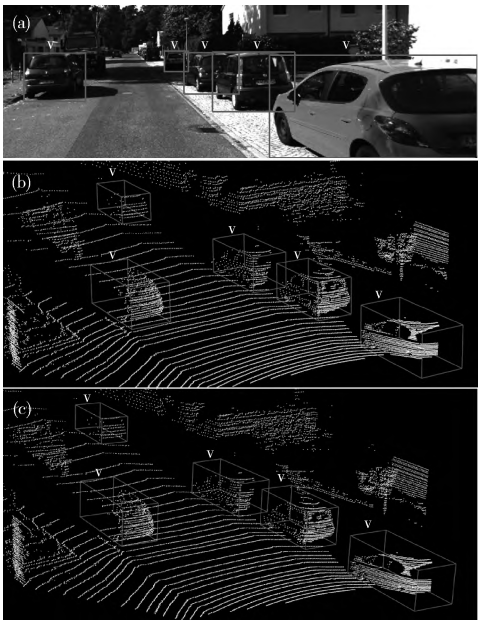


图 5　KITTI 数据集可视化样例 1

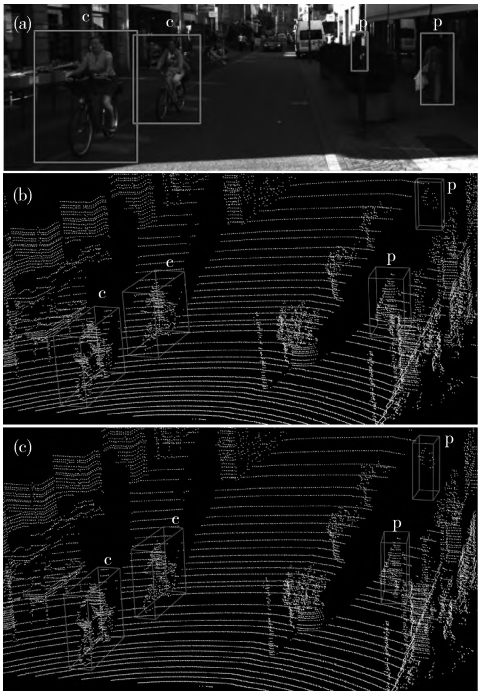


图 6　KITTI 数据集可视化样例 2

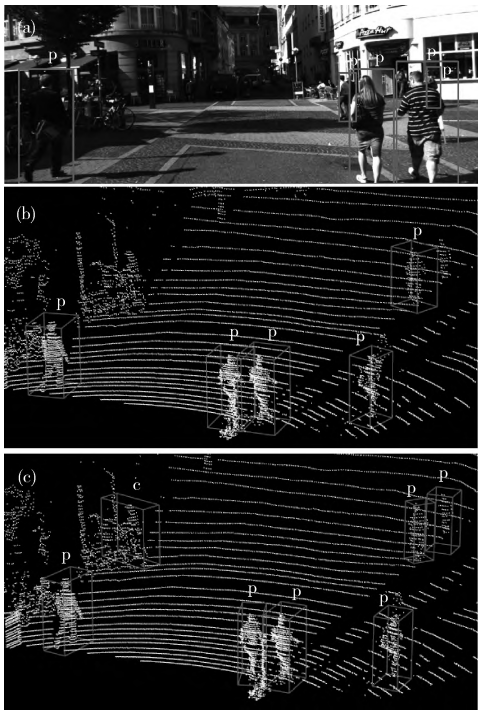


图 7 KITTI 数据集可视化样例 3

表 4 不同改进方法对 car 类别检测精度的影响

精确关键点采样	多尺度区域建议网络	基于 DIoU 的损失函数	mAP/%
✔			87.04
	✔		86.99
		✔	86.85
✔	✔		87.19
✔		✔	87.11
	✔	✔	87.08
✔	✔	✔	**87.36**

表 5 不同改进方法对 pedestrian 类别检测精度的影响

精确关键点采样	多尺度区域建议网络	基于 DIoU 的损失函数	mAP/%
✔			58.40
	✔		58.37
		✔	57.12
✔	✔		60.31
✔		✔	59.28
	✔	✔	59.67
✔	✔	✔	**61.85**

提升了 2.64%和 0.40%，分析认为行人目标在点云中比较稀疏，经过原有的最远点采样后留下的关键点数量不多，通过精确关键点采样可以为行人目标保留更多关键点，同时多尺度特征金字塔为行人目标提供了与其尺度适配的特征图，在行人目标的边界框生成以及回归阶段提供了更加丰富的特征信息，从而改善模型的检测性能。使用基于 IoU 的损失函数对模型精度提升有限，在 car、pedestrian 和 cyclist 类别中分别提升了 0.11%、0.92%和 0.62%。分析认为是由于三维目标检测中边界框重叠程度不高，导致基于 DIoU 的损失函数不能最大程度发挥性能。

表 6 不同改进方法对 cyclist 类别检测精度的影响

精确关键点采样	多尺度区域建议网络	基于 DIoU 的损失函数	mAP/%
✔			78.34
	✔		78.41
		✔	77.98
✔	✔		79.00
✔		✔	78.56
	✔	✔	78.73
✔	✔	✔	**79.41**

为了进一步探究不同采样方法对关键点选取的影响，将精确关键点采样与 PV-RCNN 采用的最远点采样进行对比实验。实验采用 KITTI 数据集，在同一场景内分别采用两种采样方法对原始点云进行采样，分别截取采样前后的车辆、行人和骑行人目标并对其包含的点云数量进行统计，实验结果见表 7。

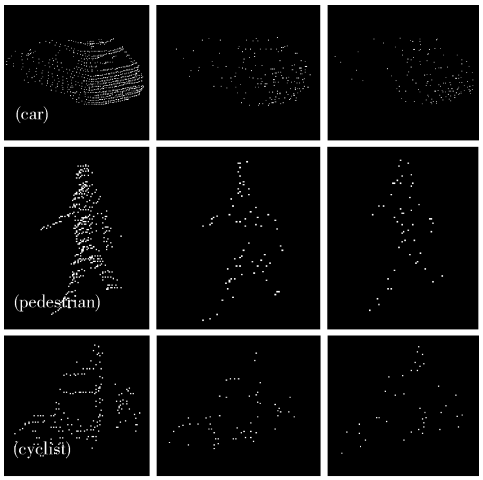
表 7 不同采样方法对比实验

点云类别	原始点云数量	精确关键点采样点云数量	最远点采样点云数量
car	966	179	145
pedestrian	374	67	51
cyclist	187	55	43

根据实验结果可以看出，在同一场景下，精确关键点采样能够采集到更多的前景点，为后续边界框回归提供更多关键点特征信息。相较于最远点采样，精确关键点采样在车辆样本的点云采样数量提升了 23.45%，在行人样本的点云采样数量提升了 31.37%，在骑车人样本的点云采样数量提升了 27.91%。不同采样方法的可视化结果如图 8 所示，其中第一列（a）原始点云对应的 3 幅图分别为从同一原始点云中截取出的车辆、行人和骑行人点云，第二列（b）精确关键点采样对应的 3 幅图分别为原始点云经过精确关键点采样后在相同的位置截取出的同一目标的点云，第三列（c）最远点采样对应的 3 幅图分别为原始点云经过最远点采样后在相同位置截取出的同一目标的点云。通过对比可以看出，在每个类别中，精确关键点采样都可以为检测目标保留更多点。

4 结束语

本文提出了一种基于点云的三维目标检测算法。设计



(a) 原始点云　(b) 精确关键点采样　(c) 最远点采样

图 8　不同采样方法对比实验

了加入点云滤波的精确关键点下采样，改善了原始网络不能将检测目标上的点在下采样中较好地保留的问题；设计了多尺度区域建议网络为不同尺度目标生成更合适的特征图；设计了加入方向感知的 DIoU 损失，帮助边界框更精确地回归。实验结果表明，所提出的模型总体性能优于原始的 PV-RCNN，在 KITTI 数据集的车辆、行人和骑行人的检测精度分别提升了 0.77%、6.33%和 2.05%。但是本算法在实时性上仍然存在不足，主要是由于 3D 稀疏卷积参数量较多导致，后续将研究对模型采用自适应剪枝进行轻量化，使算法能够更好地适用于实际使用场景。

参考文献：

[1] WANG Yadong，TIAN Yonglin，LI Guoqiang，et al. 3D object detection based on convolutional neural networks：A survey [J]. Pattern Recognition and Artificial Intelligence，2021，34 (12)：1103-1119 (in Chinese). [王亚东，田永林，李国强，等. 基于卷积神经网络的三维目标检测研究综述[J]. 模式识别与人工智能，2021，34 (12)：1103-1119.]

[2] LI Yujie，LI Xuanpeng，ZHANG Weigong. Survey of vision-based 3D object detection methods [J]. Computer Engineering and Applications，2020，56 (1)：11-24 (in Chinese). [李宇杰，李煊鹏，张为公. 基于视觉的三维目标检测算法研究综述[J]. 计算机工程与应用，2020，56 (1)：11-24.]

[3] 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. Piscataway：IEEE，2018：4490-4499.

[4] Qi C R，Su H，Mo K，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. Piscataway：IEEE，2017：652-660.

[5] Shi S，Guo C，Jiang L，et al. Pv-rcnn：Point-voxel feature set abstraction for 3D object detection [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2020：10529-10538.

[6] Qi C R，Yi L，Su H，et al. Pointnet++：Deep hierarchical feature learning on point sets in a metric space [C] //Advances in Neural Information Processing Systems. Cambridge：MIT Press，2017：5105-5114.

[7] Shi S，Wang X，Li H. Pointrcnn：3D object proposal generation and detection from point cloud [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2019：770-779.

[8] Yang Z，Sun Y，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. Piscataway：IEEE，2019：1951-1960.

[9] Yang Z，Sun Y，Liu S，et al. 3DSSD：Point-based 3D single stage object detector [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2020：11040-11048.

[10] Yan Y，Mao Y，Li B. Second：Sparsely embedded convolutional detection [J]. Sensors，2018，18 (10)：3337-3354.

[11] Zheng W，Tang W，Chen S，et al. Cia-ssd：Confident iou-aware single-stage object detector from point cloud [C] //Proceedings of the AAAI Conference on Artificial Intelligence，2021：3555-3562.

[12] Liu Z，Tang H，Lin Y，et al. Point-voxel CNN for efficient 3D deep learning [C] //Proceedings of the 33rd International Conference on Neural Information Processing Systems. Cambridge：MIT Press，2019：965-975.

[13] He C，Zeng H，Huang J，et al. Structure aware single-stage 3D object detection from point cloud [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2020：11873-11882.

[14] Lin T Y，Dollár P，Girshick R，et al. Feature pyramid networks for object detection [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2017：2117-2125.

[15] Chen Q，Wang Y，Yang T，et al. You only look one-level feature [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2021：13039-13048.

[16] Tan M，Pang R，Le Q V. Efficientdet：Scalable and efficient object detection [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2020：10781-10790.

[17] He K，Zhang X，Ren S，et al. Deep residual learning for image recognition [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2016：770-778.

[18] Zheng Z，Wang P，Liu W，et al. Distance-IoU loss：Faster and better learning for bounding box regression [C] //Proceedings of the AAAI Conference on Artificial Intelligence，2020：12993-13000.

[19] 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. Piscataway：IEEE，2019：12697-12705.