文章编号：1007-2780(2022)10-1355-09

体素化点云场景下的三维目标检测

李瑞龙[1,2]，吴　川[1,2*]，朱　明[1,2]
(1. 中国科学院 长春光学精密机械与物理研究所，吉林 长春 130033；
2. 中国科学院大学，北京 100049)

摘要：基于激光雷达点云数据的三维目标检测算法受制于数据量大，无法实现速度与准确率的平衡。本文提出一种改进的三维目标检测算法 Pillar RCNN。首先将目标点云空间划分为体素格，使用一种基于稀疏卷积的三维主干网络将体素格逐步转化为立柱体素，三维信息量化为致密的二维信息。然后使用二维主干网络提取特征，同时将三维骨干网络中不同尺度的体素特征与二维主干网络通过多尺度体素特征聚合模块进行特征级联，通过损失函数进一步细化检测框。算法在 KITTI 公开数据集上进行测试，在 RTX 2080Ti 硬件平台上识别速度为 2.48 ms。汽车、行人、自行车 3 种类别的检测效果同 PointPillars 基准算法相比较，其中自行车中等难度检测效果提升 13.34%，困难难度的车检测效果提升 8.85%，其他类别的检测准确率指标也有所提升，实现了速度与准确率的平衡。

关　键　词：计算机视觉；三维目标检测；体素；稀疏卷积；特征聚合
中图分类号：TP391.4　　**文献标识码**：A　　**doi**：10.37188/CJLCD.2022-0082

3D object detection in voxelized point cloud scene

LI Rui-long[1,2], WU Chuan[1,2*], ZHU Ming[1,2]

(1. *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun* 130033, *China*;
2. *University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract: The 3D object detection algorithm is constrained by the large amount of point cloud data, and can not achieve the balance of real-time speed and accuracy. This paper presents an improved 3D object detection algorithm—Pillar RCNN. The algorithm firstly divides the target point cloud space into voxels, presents a 3D object detection backbone network based on sparse convolution which gradually converts voxels into column voxels, quantifies 3D information into dense 2D information, and then processes the dense 2D information through the 2D backbone network. At the same time, the voxel features of different scales in the 3D backbone network and the 2D backbone network detection results are cascaded through the multiscale voxel feature aggregation module, and the result is further refined by the loss function. The algorithm is tested on KITTI public datasets and has a recognition speed of 2.48 ms on RTX 2080Ti hardware platform. Compared with the PointPillars benchmark algorithm, the performance indicators of the three categories of car, pedestrian and bicycle are improved. The detection results of the mode of bicycle and the hard of car are improved by 13.34% and 8.85%, and the detectation accuracies of other

收稿日期：2022-03-10；**修订日期**：2022-03-25.
*通信联系人，E-mail：wuchuan0458@sina.com

categories are improved also, achieving a balance between speed and accuracy.

Key words: computer vision; 3D detection; voxel; sparse convolution; feature aggregation

1 引 言

随着自动驾驶技术的发展,三维目标检测成为三维场景感知和理解的核心,是车辆与外界信息交互的重要渠道,从而实现车辆与周边环境的信息交互,在自动驾驶中扮演不可替代的作用[1]。其中,基于激光雷达点云的三维目标检测算法从点云数据精确的几何形状和特征信息预估三维边界框和类别,实现检测目标功能。激光雷达具有精度高、探测距离远、不受天气光照影响、数据稳定性强的优点,基于点云的视觉感知技术也成为自动驾驶的研究热点,在三维目标检测中占主导地位。

点云是目标表面特性的海量点的集合,包含目标场景的立体坐标信息、反射强度、法向量、局部稠密度和局部曲率等特征。由于点云在空间中离散分布,因此具有稀疏性且分布不均的特点[2],不能直接运用处理图像的方式处理点云数据,也无法利用传统算法获取点云丰富的结构信息和空间语义信息。随着深度学习在处理点云方面取得突破性进展,现已发展出很多基于深度学习的三维目标检测方向,主要分为基于点云的方法和基于体素的方法。

PointNet[3]直接使用点云作为输入,利用卷积神经网络有效解决无序点云特征建模问题。使用空间变换矩阵预测网络学习原始三维点云的空间分布特性,将具有同一特征的三维点云对齐后送入多层感知机提取特征。但是在点云数据稀疏时,目标主干点的点云缺失导致无法准确提取全局特征,鲁棒性效果较差。PointNet++[4]根据CNN多层感受野思想,设置不同的邻域搜索半径(Ball Query),使用PointNet提取不同尺度范围的点云特征,再使用多层感知机生成全局特征,最终实现点级特征分类。PointNet++在点云语义分割任务上效果较好,但其邻域搜索复杂度较高,无法实时处理。PointNet及PointNet++实现了对点云的处理与特征提取,在后续算法中得到了广泛的应用。

PointRCNN[5]算法使用原始点云作为输入,在第一阶段,使用PointNet++逐点提取特征向量,将点云分类为前景和背景,再使用多层感知机根据每个前景点的特征向量生成3D检测框。在第二阶段,将语义特征和局部空间特征结合对检测框进行优化。STD[6](Sparse-to-Dense 3D Object Detector for Point Cloud)在第一阶段使用PointsPool将内部点特征从稀疏表达转换为致密表示,在每个点上使用球形锚框生成准确的三维检测框。在第二阶段通过IoU分支回归边界框。结果显示,STD在精度和速度方面均优于PointRCNN。

三维点云空间划分成的立方体称为体素,体素能有效表示点云空间。Voxelnet[7]将点云划分为等间距的规则体素,使用VFE层(Voxel feature encoder)将每个体素内的点的特征量化统一,再采用3D卷积神经网络提取点云特征信息,最终使用RPN网络生成检测框,实现端到端训练。

PointPillars[8]网络通过在鸟瞰图(Bird's Eye View,BEV)上划分网格,实现立柱形式的体素的划分。通过立柱的划分,将三维点云空间降维生成鸟瞰图,借鉴成熟的二维图像卷积网络进行特征提取和边界框回归,实现端到端的训练,最终实现62 Hz的运行速度。Voxel R-CNN[9]算法由3D骨干网、2D鸟瞰图区域提议网络和SSD检测头组成,利用体素聚合模块直接从体素特征中提取三维特征,算法检测速度较慢。

在三维目标检测算法中,点表示的方法可以保留点云精确位置,算法精度高,但将点云直接作为输入的计算量较大。基于体素划分的方法具有速度优势,但体素的划分会导致三维特征信息丢失,影响检测精度。

两种点云表示方法各有优缺点。本文将点云和体素两种数据表示方法结合使用,通过对体素化的点云数据进行处理保证运行速度,同时结合点表示的方法聚合点云特征保证准确率,实现了速度与准确率的平衡。

2 基于体素的三维目标检测

本文提出的Pillar RCNN(Pillar Region with

CNN feature）由 3 个核心模块组成：（1）基于稀疏卷积的 3D 主干网络；（2）2D 主干网络和 RPN（Region Proposal Network）模块；（3）体素 RoI 多尺度特征聚合模块。网络结构如图 1 所示。

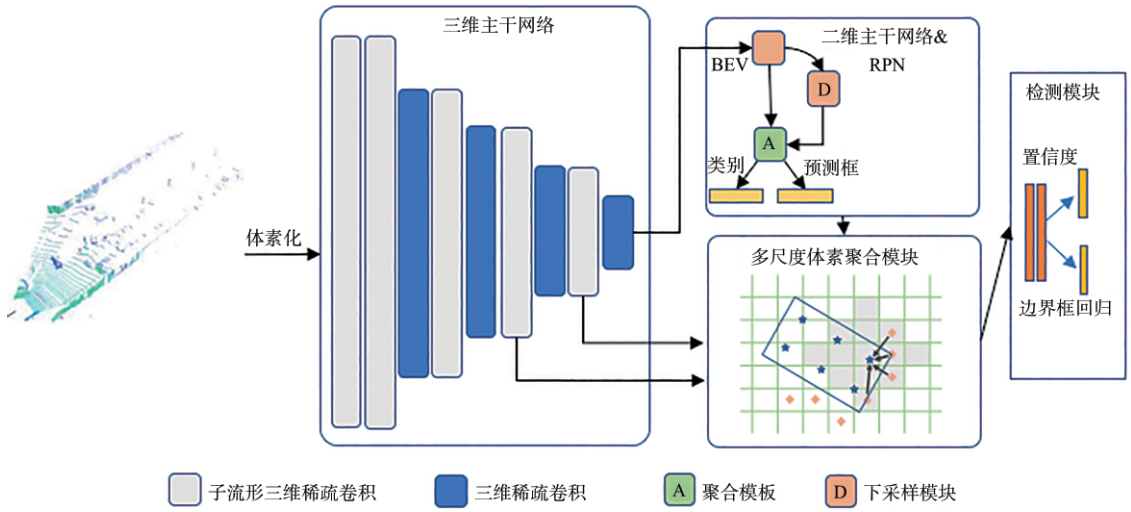


图 1　Pillar RCNN 框架图

Fig. 1　Framework of Pillar RCNN

根据点云场景比较稀疏的特点，本文重新设计了三维子流形稀疏卷积骨干网络，快速将体素下采样为立柱形式，生成致密的二维鸟瞰图信息。再使用 PointNet++提取邻域体素的点云特征，利用聚合模块对二维鸟瞰图的特征和邻域体素特征进行级联，进行三维边界框进一步的回归细化。

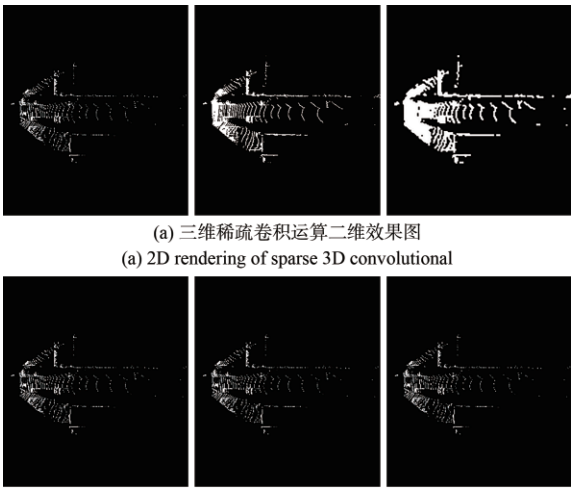
2.1　点云数据预处理及体素化

三维空间内点云具有无序性。首先将场景空间内划分为立体体素。对于点云数据，以车辆前进方向为 X 轴，左右方向为 Y 轴，垂直于 X-Y 平面的方向为 Z 轴[10]，设检测目标场景在 3 个轴上范围区间为 L，单位为 m，取 $L_X \times L_Y \times L_Z = [0, 70.4] \times [-40, 40] \times [-3, 1]$，体素大小为 $r_X \times r_Y \times r_Z = 0.05\,\text{m} \times 0.05\,\text{m} \times 0.1\,\text{m}$，得到目标场景为 1 408×1 600×40 个初始体素。非空体素内随机采样 5 个点来代表体素，若点云数量不足 5 个则重复采样，初始特征取三维坐标（x，y，z）及反射率 r_r。体素的尺寸大小直接影响算法运行速度，其中非空体素仅占检测空间的 5%～10%。因为大量不包含点云的体素的存在，体素的稀疏性较强，传统的三维卷积网络遍历体素空间会极大降低模型推理速度，占用大量内存，这既浪费了计算资源，也浪费了内存资源。为克服密集卷积的缺点，本文使用三维稀疏卷积提取体素内点云特征。

2.2　基于稀疏卷积三维主干网络

为加速普通三维卷积运行速度在 S3D-CNN[11]（Sparse 3D Convolutional Neural Network）网络中使用三维稀疏卷积层进行加速运算。三维稀疏卷积层遵循如果没有相关的输入点，则不计算输出点的原则。本文采用三维稀疏卷积对体素数据进行卷积运算，卷积范围内无点云的体素不进行计算，在加速运算的同时减小内存占用。然而，由于存在大量活动点，如有数据体素相邻区域的无数据的体素，以该体素为卷积中心，经过卷积运算后仍会有输出，导致输出稀疏度高于输入稀疏度，后续卷积层的速度降低。如图 2（a）所示，左图经过多层三维稀疏卷积后，在中间和右图的输出数据量逐渐增加。Graham 提出子流形三维稀疏卷积[12]，在输入位置处于有数据状态时，对应输出位置进行卷积计算。经过子流形稀疏卷积后，稀疏性不变，如图 2（b）所示，对左图进行子流形三维稀疏卷积后，中间和右图输出数据大小不变。

本文设计了一个主要由两种稀疏三维卷积网络组成的三维骨干网络。两种类型的稀疏卷积块由稀疏三维卷积块和子流形稀疏三维卷积块组成。在三维卷积模块中，稀疏卷积中步长为



(a) 三维稀疏卷积运算二维效果图
(a) 2D rendering of sparse 3D convolutional

(b) 子流形三维稀疏卷积二维效果图
(b) 2D rendering of submanifold 3D sparse convolutional

图 2 三维稀疏卷积(a)和子流形稀疏卷积(b)运算示意图

Fig. 2 Sample images of sparse 3D convolutional (a) and submanifold 3D sparse convolutional (b)

2,负责将三维体素进行下采样,降低三维分辨率。这种稀疏卷积操作会降低数据的稀疏性,但可以提取更加丰富的结构特征,同时经过子流形稀疏卷积后,数据的稀疏性保持不变。子流形稀疏卷积中步长为 1。两种稀疏卷积模块后均使用 BN 和 ReLU 层,实现数据的归一化,加快学习速度。最终初始体素维度逐步下采样至 150×200×1,快速实现立柱形式的体素。

2.3 鸟瞰图目标检测主干网络

三维骨干网将体素化的点云信息逐渐转换为特征体块,然后将输出张量沿 Z 轴叠加生成鸟瞰图特征映射。二维主干网由 3 部分组成:两个标准 4 层 3×3 特征提取子网络和一个多尺度特征融合子网络,在每个卷积层后进行 BN 和 ReLU 运算。第一个特征提取模块的 X、Y 轴上分辨率与 3D 骨干网输出分辨率相同,第二个特征提取模块分辨率为上一模块的 1/2,最后多尺度特征融合子网络将两模块的特征进行融合,构建高分辨率的特征图。最后将二维主干网的输出与 RPN 中 1×1 卷积层进行卷积,生成三维目标预测框和预测类别。

2.4 多尺度体素特征聚合模块

为了更好地从三维体素空间内聚合空间上下文信息,提高算法准确率,使用多尺度体素 RoI 特征聚合模块,如图 3 所示。将稀疏的三维体块

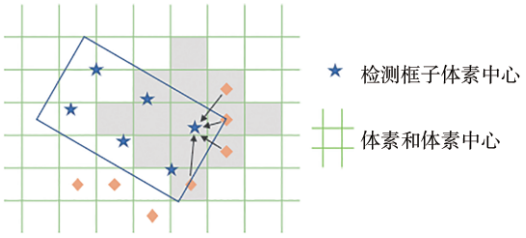


图 3 多尺度体素 RoI 聚合模块二维示意图

Fig. 3 2D diagram of multi-scale voxel RoI aggregation module

表示为一组非空体素中心点 $\left\{d_i=\left(x_i,y_i,z_i\right)_{i=1}^{N}\right\}$ 及对应的特征向量 $\{v_i\}_{i=1}^{N}$,体素中心坐标通过位置索引、体素边界计算得来。与无序的点云相比,体素在量化空间中有规律地排列,方便查询相邻体素,例如查询 26 临近体素可以通过索引值偏移 $\left\{\left(o_x,o_y,o_z\right),\ o_x,o_y,\ o_z\epsilon[-1,0,1]\right\}$ 得到。在体素查询中采用曼哈顿距离 $D_{\alpha,\beta}$ 计算相邻体素 $\alpha=\left(i_\alpha,j_\alpha,k_\alpha\right),\beta=\left(i_\beta,j_\beta,k_\beta\right)$ 的距离:

$$D_{\alpha,\beta}=\left|i_\alpha-i_\beta\right|+\left|j_\alpha-j_\beta\right|+\left|k_\alpha-k_\beta\right|. \quad (1)$$

区域推荐网络 RPN 生成的三维检测框划分为 6×6×6 子体素网格,每个子体素网格使用中心点为索引,体素 RoI 特征聚合模块将子体素邻近体素的特征整合到网格点中进行特征提取,没有直接使用最大池化层(Max Pooling)简单聚合相邻体素的特征,而是使用 PointNet 模块将邻域体素聚合为:

$$\eta_i=\max\left\{\Psi\left(\left[v_i-g_i;\varnothing_i\right]\right)\right\}, \quad (2)$$

v_i-g_i 为一个相邻体素的位置关系,$\varnothing_i$ 为 v_i 点代表的邻域体素经过 PointNet 模块得到的体素特征,$\Psi(\cdot)$ 表示多层感知机。最后再使用最大池化层(Max Pooling)或平均池化层(Average pooling)处理多层感知机输出的特征。最大池化层能够减小由卷积层参数误差造成的均值偏移,更好地保留纹理信息,效果较好。邻域内每个体素重复上述过程。利用体素 RoI 聚合模块,在每个从 3D 骨干网的最后两个阶段的体素中提取体素特征。每个阶段设置两个曼哈顿距离阈值对多尺度的体素进行分组。将从不同阶段和尺度汇集的特征进行拼接,得到体素 RoI 特征进行边界框的精细回归。

本文将两种池化层对特征提取模块的影响

进行对比实验（表1），发现最大池化层效果较好，故后文均采用最大池化层。检测模块将多尺度体素RoI特征作为输入。首先使用2个多层感知机将RoI特征转换为特征向量，再分别送入边界框回归分支和置信度预测分支。边界框回归分支预测三维检测框与目标真值框的偏差，置信度预测分支预测三维检测框分数，得到最终检测输出。

表1　最大池化层和平均池化层在3D目标检测上汽车类目标AP40精度对比

Tab. 1　Accuracy comparison of max pooling layer and average pooling layer in 3D object detection of car AP40　（%）

	Easy	Mod	Hard
Max	92.34	84.63	82.48
Average	92.17	83.01	82.27

2.5　参数设置

参数设置中汽车的检测框设置为[3.9×1.6×1.56]，行人检测框设置[0.8×0.6×1.73]，自行车检测框设置大小为[1.76×0.6×1.73]。初始学习率为0.01，区域推荐网络中使用非极大值抑制（NMS），IoU阈值为0.7，保留100个检测区域作为检测头的输入。在检测阶段，再次应用NMS，IoU阈值为0.1，去除冗余预测。本文对多尺度体素RoI模块设置在三维骨干网络的最后的模块3和模块4上不同作用邻域阈值进行实验。经过实验对比，在三维骨干最后两模块中曼哈顿距离阈值均设置为[1,2]、[2,4]。不同邻域阈值对多尺度RoI特征聚合模块在3D目标检测上汽车类AP40效果对比如表2所示。

表2　不同邻域阈值对多尺度RoI特征聚合模块在3D目标检测上汽车类AP40效果对比

Tab. 2　Comparison of car AP40 effects of different neighborhood thresholds on multi-scale ROI feature aggregation module in 3D object detection　（%）

[block3][block4]	Easy	Mod	Hard
[1−2][2,4]	92.42	85.43	83.04
[2,4][2,4]	92.34	84.63	82.48
[2,4][4,8]	92.37	85.15	82.98

3　损失函数设计

根据KITTI数据集目标框标注信息，可以将目标量化为 $\boldsymbol{G}_b=(x_b,y_b,z_b,l_b,w_b,h_b,\theta_b)$ 的七维向量，(x_b,y_b,z_b) 为目标框中心坐标，(l_b,w_b,h_b) 为目标框的长宽高，θ_b 代表目标框在鸟瞰图上绕 Z 轴的旋转角度。在训练阶段，检测框也量化为 $\boldsymbol{G}_{\text{pred}}=(x_{\text{pred}},y_{\text{pred}},z_{\text{pred}},l_{\text{pred}},w_{\text{pred}},h_{\text{pred}},\theta_{\text{pred}})$ 七维向量，参数意义与目标真值框意义相同，针对类别为汽车的目标，真实框与检测框IoU阈值大于0.6的视为前景目标框，小于0.45视为背景目标框，介于0.45～0.6之间视为无效框，对网络学习无贡献。针对行人和骑自行车的人，阈值选择为0.35和0.5。目标真值框与前景目标预测框可以编码为：

$$x_l=\frac{x_b-x_{\text{pred}}}{d_{\text{pred}}},\ y_l=\frac{y_b-y_{\text{pred}}}{d_{\text{pred}}},\ z_l=\frac{z_b-z_{\text{pred}}}{h_{\text{pred}}}$$

$$l_l=\log\left(\frac{l_b}{l_{\text{pred}}}\right),\ w_l=\log\left(\frac{w_b}{w_{\text{pred}}}\right),\ h_l=\log\left(\frac{h_b}{h_{\text{pred}}}\right)$$

$$\theta_l=\sin(\theta_b-\theta_{\text{pred}}),\tag{3}$$

式中 $d_{\text{pred}}=\sqrt{(l_{\text{pred}})^2+(w_{\text{pred}})^2}$ 为前景目标预测框的鸟瞰图平面对角线，用来量化 x_l 和 y_l 误差。

3.1　RPN损失函数

根据边界框的分类设计RPN损失函数：

$$\mathcal{L}_{\text{RPN}}=\frac{1}{N_{\text{post}}}\left[\sum_i\mathcal{L}_{\text{cls}}(\text{cls}_i^{\text{pred}},\text{cls}_i^{\text{b}})+\sum_i\mathcal{L}_{\text{reg}}(G_i^{\text{pred}},G_i^{\text{b}})\right],\tag{4}$$

式中 N_{post} 为前景目标框的数量，$\text{cls}_i^{\text{pred}}$ 和 G_i^{pred} 为分类和目标框回归分支的输出，cls_i^{b} 和 G_i^{b} 为真值框分类标签和目标框。由于数据集中的正负样本比例失衡，虽然网络采用数据增强方法扩展了正样本，但与检测框相比比例仍不平衡。类别分类采用Focal loss[13]分类交叉熵损失解决类别不平衡分布问题，具体如式(5)所示：

$$\mathcal{L}_{\text{cls}}=\begin{cases}-\alpha(1-p_{\text{pred}})^{\gamma}\log(p_{\text{pred}}),\ y=1\\-(1-\alpha)\,p_{\text{pred}}{}^{\gamma}\log(1-p_{\text{pred}}),\ y=0\end{cases},\tag{5}$$

式中：α 为平衡因子，γ 为调整因子，p_{pred} 为分类预测值。实验中取值 $\alpha=0.25$，$\gamma=2$。回归目标边界框时采用Huber Loss，增强对预测边框回归的

鲁棒性。当预测偏差小于δ时，采用平方误差；当预测偏差大于δ时，采用线性误差。公式如式(6)所示：

$$L_\delta=\begin{cases}\frac{1}{2}\left(L_{\mathrm{reg}}\right)^2\delta, & \left|L_{\mathrm{reg}}\right|\leqslant\delta\\ \delta\left(\left|L_{\mathrm{reg}}\right|-\frac{1}{2}\delta\right), & \text{otherwise}\end{cases}. \tag{6}$$

3.2 预测损失

检测分支得到前景目标框与真值框进行IoU操作后，得到检测框与真值框的交并比。检测分支的损失函数如式(7)所示：

$$l_i^*(\mathrm{IoU}_i)=\begin{cases}0 & \mathrm{IoU}_i<\theta_{\mathrm{L}}\\ \frac{\mathrm{IoU}_i-\theta_{\mathrm{L}}}{\theta_{\mathrm{H}}-\theta_{\mathrm{L}}} & \theta_{\mathrm{L}}<\mathrm{IoU}_i<\theta_{\mathrm{H}}\\ 1 & \mathrm{IoU}_i>\theta_{\mathrm{H}}\end{cases}, \tag{7}$$

式中IoU_i为与真值框对应第i个前景框的IoU，θ_{H}和θ_{L}分别为前景IoU阈值和背景IoU阈值。本文采用二元交叉熵损失进行置信度预测，回归损失为Huber Loss。最终损失函数为：

$$\mathcal{L}_{\mathrm{head}}=\frac{1}{N_s}\left[\sum_i\mathcal{L}_{\mathrm{cls}}\left(p^i,l_i^*(\mathrm{IoU}_i)\right)+\mathcal{L}\left(\mathrm{IoU}_i\geqslant\theta_{\mathrm{reg}}\right)\sum_i\mathcal{L}_{\mathrm{reg}}\left(\delta_i^a,t_i^*\right)\right], \tag{8}$$

式中N_s为训练阶段采样的锚框个数，$\mathcal{L}\left(\mathrm{IoU}_i\geqslant\theta_{\mathrm{reg}}\right)$表示只计算$\mathrm{IoU}_i\geqslant\theta_{\mathrm{reg}}$的区域提案才会导致回归损失，$\theta_{\mathrm{reg}}$取0.55。

4 实验过程

本文使用KITTI数据集评估Pillar RCNN在汽车、行人和自行车上3D目标检测和BEV目标检测任务的性能。KITTI[14]数据集的点云数据通过VelodyneHDL-64E机械式旋转激光雷达采集。考虑到目标的大小、遮挡和截断，每个类的对象被划分为3个不同的难度级别：容易、中等和困难。本文将7 481个样本分为包含3 712个样本的子训练集和包含3 769个样本的子验证集，遵循KITTI的官方评估协议，对3D物体检测和BEV物体检测进行评估，汽车的IoU阈值为0.7，行人和骑自行车的人的IoU阈值为0.5，平均精度(AP)的计算作为性能衡量。

4.1 实验环境

本文实验环境操作系统为Ubuntu18.04，硬件显卡型号为NVIDIA 2080Ti，Intel(R) Core™ i9-9900K CPU@3.60 GHz。实验程序基于Pytorch1.3框架编写，Python环境为3.6。

4.2 数据增强

本文通过数据增强策略生成更多的目标样本训练模型。首先根据SECOND[15]中的方法，从训练数据集中收集目标样例，包括标签、三维真值框和框内的点云。对于子训练集中的场景，随机选取其他场景15、10、10个汽车、行人和骑自行车的人的数据，融合到训练数据中。同时，所有的目标真值框和相应的点云分别围绕Z轴旋转，并沿着X、Y、Z轴平移，旋转角度沿$\varphi[-\pi/20,\pi/20]$均匀分布，平移量服从高斯分布$\mu[0,0.25]$。如果背景点与目标三维真值框冲突，移除背景点。最后整体场景随机全局旋转，旋转量服从$\varphi[-\pi/4,\pi/4]$均匀分布，沿X、Y、Z轴按照高斯分布$\mu[0,0.2]$随机全局平移，按照$\varphi[0.95,1.05]$均匀分布随机缩放。

5 实验结果分析

Pillar RCNN在KITTI训练数据集训练120个epoch，在验证集进行自动评估，采用平均精度(AP11)进行比较，实验结果如图4所示，检测结果投影到对应图像和点云中。

实验结果通过与基于雷达数据的PointPillars、

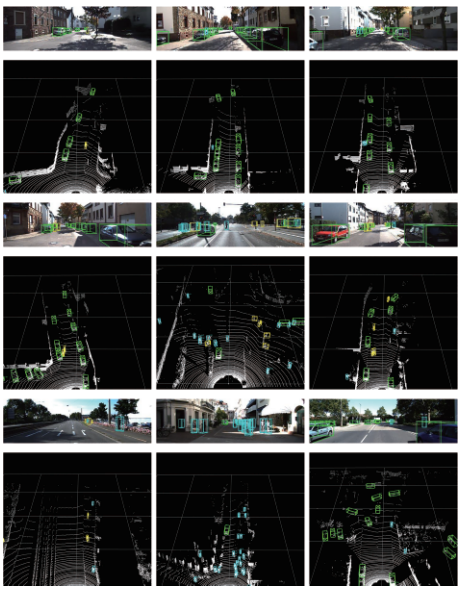


图4 三维检测效果在图像和点云中的投影图

Fig. 4 Projection of 3D detection effects in images and point clouds

PIXOR[16]、VoxelNet、SECOND、Voxel R-CNN和基于点云和图像融合的MV3D[17]等经典网络进行对比（上述网络性能来自文献），最终在三维检测和鸟瞰图上的实验结果如表3、表4所示。

表3　KITTI测试数据集3D目标AP11精度对比

Tab. 3　Performance comparison on the KITTI test set with AP calculated by 11 recall positions of 3D object　（%）

	Modality	FPS	Car			Pedestrian			Cyclist		
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
MV3D	Lidar & Img.	3	71.19	56.6	55.3	N/A	N/A	N/A	N/A	N/A	N/A
SECOND	Lidar	30.4	87.43	76.48	69.1	N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet	Lidar	4.44	81.97	65.46	62.85	57.86	53.42	48.87	67.17	47.65	45.11
Voxel R−CNN	Lidar	25.2	**89.41**	**84.52**	**78.93**	N/A	N/A	N/A	N/A	N/A	N/A
PointPillars	Lidar	**62.5**	85.22	76.52	69.73	63.87	58.08	52.8	77.16	59.22	55.91
Pillar RCNN	Lidar	40.3	89.26	83.61	78.58	**65.03**	**58.80**	**54.47**	**85.35**	**72.56**	**68.68**

注：加粗字体为每行最优值

表4　KITTI测试数据集BEV目标AP11精度对比

Tab. 4　Performance comparison on the KITTI test set with AP calculated by 11 recall positions of BEV object　（%）

	Modality	FPS	Car			Pedestrian			Cyclist		
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
MV3D	Lidar & Img.	3	86.18	77.32	76.33	N/A	N/A	N/A	N/A	N/A	N/A
PIXOR	Lidar & Img.	10.6	86.79	80.75	76.6	N/A	N/A	N/A	N/A	N/A	N/A
SECOND	Lidar	30.4	89.96	87.07	79.66	N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet	Lidar	4.44	89.6	84.81	78.57	69.95	61.05	56.98	74.41	52.18	50.49
PointPillars	Lidar	**62.5**	90.07	87.44	84.78	**71.18**	**65.92**	**61.36**	82.2	62.81	59.77
Pillar RCNN	Lidar	40.3	**90.15**	**88.13**	**87.65**	66.13	61.16	58.03	**92.21**	**73.85**	**70.65**

注：加粗字体为每行最优值

Pillar RCNN在3D目标检测上效果较好，在检测精度和速度上取得了较好的平衡。3D目标检测经过体素RoI特征模块匹配，相较于PointPillars检测性能在3个类别上均实现了明显提升，其中自行车和汽车两种类别的3D检测类别提升效果明显，自行车类的中等难度提升13.34%，汽车类别中困难难度提升了8.85%。在汽车类别上检测精度在Voxel R-CNN上效果较好，中等难度比Pillar RCNN高0.91%。表3为3D目标检测的AP11精度对比。Pillar RCNN在各项检测任务上相较于其他网络均有不错的提升，算法实时性与性能取得平衡。但PointPillars在检测速度上具有明显优势。PointPillars在划分体素阶段直接把检测空间划分为立柱形式的体素，没有使用三维卷积和体素RoI模块，且PointPillars经过NVIDIA TensorRT模块加速，算法速度优势明显。TensorRT加速模块不支持稀疏卷积，但在Pytorch原始框架测试Pillar RCNN仍取得了40.3 Hz的检测速度，验证了Pillar RCNN算法的性能。

Pillar RCNN在目标检测上对于大目标检测效果较好，对行人类别检测效果不具有优势。因为设置体素尺寸较大，经过三维骨干网络8倍下采样，行人目标区域在致密的二维空间中占不超过两个像素点，目标较小，因此在目标检测中对于行人等小目标检测效果较一般。

6　结　　论

本文研究了基于立柱体素三维目标检测算法在点云场景中的应用。针对点云稀疏性，重新设计了稀疏三维卷积主干网络提取特征，能够快速将三维体素量化为二维致密区域特征，同时避免了直接划分立柱体素丢失三维空间信息的缺点。经过多尺度体素RoI特征聚合模块实现特

征融合后,进一步细化边界框检测三维目标,提高了检测精度。Pillar RCNN使用稀疏卷积降低传统三维卷积的计算冗余,提高了算法检测性能。在KITTI数据集上的实验表明,Pillar RCNN实现了速度与检测精度之间的平衡,在3D目标检测任务上进一步提升了三维检测算法性能。

参 考 文 献 :

[1] 赵毅强,艾西丁·艾克白尔,陈瑞,等. 基于体素化图卷积网络的三维点云目标检测方法[J]. 红外与激光工程,2021,50(10):20200500.
ZHAO Y Q, AKBAR A, CHEN R, *et al*. 3D point cloud object detection method in view of voxel based on graph convolution network [J]. *Infrared and Laser Engineering*, 2021, 50(10): 20200500. (in Chinese)

[2] 严娟,方志军,高永彬. 结合混合域注意力与空洞卷积的3维目标检测[J]. 中国图象图形学报,2020,25(6):1221-1234.
YAN J, FANG Z J, GAO Y B. 3D object detection based on domain attention and dilated convolution [J]. *Journal of Image and Graphics*, 2020, 25(6): 1221-1234. (in Chinese)

[3] CHARLES R Q, SU H, KAICHUN M, *et al*. PointNet: deep learning on point sets for 3D classification and segmentation [C]//*Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017: 77-85.

[4] QI C R, YI L, SU H, *et al*. PointNet++: deep hierarchical feature learning on point sets in a metric space [C]//*Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach: Curran Associates Inc, 2017.

[5] SHI S S, WANG X G, LI H S. PointRCNN: 3D object proposal generation and detection from point cloud [C]//*Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach: IEEE, 2019.

[6] YANG Z T, SUN Y N, LIU S, *et al*. STD: sparse-to-dense 3D object detector for point cloud [C]//*Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul: IEEE, 2019.

[7] ZHOU Y, TUZEL O. VoxelNet: end-to-end learning for point cloud based 3D object detection [C]//*Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 4490-4499.

[8] LANG A H, VORA S, CAESAR H, *et al*. PointPillars: fast encoders for object detection from point clouds [C]//*Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach: IEEE, 2019: 12689-12697.

[9] DENG J J, SHI S S, LI P W, *et al*. Voxel R-CNN: towards high performance voxel-based 3D object detection [C]//Proceedings of the 35th AAAI Conference on Artificial Intelligence. Online: AAAI, 2021.

[10] WANG B, ZHU M, LU Y, *et al*. Real-time 3D object detection from point cloud through foreground segmentation [J]. *IEEE Access*, 2021, 9: 84886-84898.

[11] GRAHAM B. Sparse 3D convolutional neural networks [C]//*Proceedings of the British Machine Vision Conference* 2015. Swansea: BMVA Press, 2015.

[12] GRAHAM B, ENGELCKE M, VAN DER MAATEN L. 3D semantic segmentation with submanifold sparse convolutional networks [C]//*Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 9224-9232.

[13] LIN T Y, GOYAL P, GIRSHICK R, *et al*. Focal loss for dense object detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 42(2): 318-327.

[14] 刘长吉,郝志成,杨锦程,等. 基于实例分割的目标三维位置估计方法[J]。液晶与显示,2021,36(11):1535-1544.
LIU C J, HAO Z C, YANG J C, *et al*. Object 3D position estimation based on instance segmentation [J]. *Chinese Journal of Liquid CRystals and Displays*, 2021, 36(11): 1535-1544. (in Chinese)

[15] YAN Y, MAO Y X, LI B. SECOND: sparsely embedded convolutional detection [J]. *Sensors*, 2018, 18(10):

3337.

[16] YANG B, LUO W J, URTASUN R. PIXOR: real-time 3D object detection from point clouds [C]//*Proceedings of* 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 7652-7660.

[17] CHEN X Z, MA H M, WAN J, *et al*. Multi-view 3D object detection network for autonomous driving [C]//*Proceedings of the* 30*th IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017: 6526-6534.

作者简介：

李瑞龙（1996—），男，山东枣庄人，硕士研究生，2019年于兰州交通大学获得学士学位，主要从事计算机视觉方面的研究。E-mail：liruilong19@mails.ucas.edu.cn

吴　川（1976—），男，吉林长春人，博士，研究员，2005年于中国科学院长春光学精密机械与物理研究所获得博士学位，主要从事图像融合、目标跟踪、嵌入式系统开发等方面的研究。E-mail：wuchuan0458@sina.com