激光与光电子学进展 先进成像

基于改进SECOND算法的点云三维目标检测

张莹，蒋亮亮*，张东波，段万林，孙月
湘潭大学自动化与电子信息学院，湖南 湘潭 411105

摘要 快速识别和精准定位周围目标是自动驾驶车辆安全、自主行驶的前提和基础。针对基于体素的点云三维目标检测方法识别与定位不准的问题，提出一种基于改进SECOND算法的点云三维目标检测算法。首先，在二维卷积骨干网络中引入自适应的空间特征融合模块融合不同尺度的空间特征，提高模型的特征表达能力。其次，充分利用边界框参数之间的关联性，采用three-dimensional distance-intersection over union（3D DIoU）损失作为边界框的定位回归损失函数，使得回归任务更加高效。最后，同时考虑候选框的分类置信度和定位精度，通过一个新的候选框质量评价标准，获得更平滑的回归结果。在KITTI测试集的实验结果表明，所提算法的3D检测精度优于许多以往的算法，与基准算法SECOND相比，在简单难度下的car类和cyclist类分别提高2.86百分点和3.84百分点，中等难度下分别提高2.99百分点和3.89百分点，困难难度下分别提高7.06百分点和4.27个百分点。

关键词 自动驾驶；三维目标检测；特征融合；损失函数
中图分类号 TP391.4 **文献标志码** A **DOI**：10.3788/LOP231016

Point Cloud 3D Object Detection Based on Improved SECOND Algorithm

Zhang Ying, Jiang Liangliang*, Zhang Dongbo, Duan Wanlin, Sun Yue
College of Automation and Electronic Information, Xiangtan University, Xiangtan 411105, *Hunan, China*

Abstract Rapid identification and precise positioning of surrounding targets are prerequisites and represent the foundation for safe autonomous vehicle driving. A point cloud 3D object detection algorithm based on an improved SECOND algorithm is proposed to address the challenges of inaccurate recognition and positioning in voxel-based point cloud 3D object detection methods. First, an adaptive spatial feature fusion module is introduced into a 2D convolutional backbone network to fuse spatial features of different scales, so as to improve the model's feature expression capability. Second, by fully utilizing the correlation between bounding box parameters, the three-dimensional distance-intersection over union (3D DIoU) is adopted as the bounding box localization regression loss function, thus improving regression task efficiency. Finally, considering both the classification confidence and positioning accuracy of candidate boxes, a new candidate box quality evaluation standard is utilized to obtain smoother regression results. Experimental results on the KITTI test set demonstrate that the 3D detection accuracy of the proposed algorithm is superior to many previous algorithms. Compared with the SECOND benchmark algorithm, the car and cyclist classes improves by 2.86 and 3.84 percentage points, respectively, under simple difficulty; 2.99 and 3.89 percentage points, respectively, under medium difficulty; and 7.06 and 4.27 percentage points, respectively, under difficult difficulty.

Key words autonomous driving; three-dimensional object detection; feature fusion; loss function

1 引 言

三维目标检测[1]是自动驾驶汽车的关键感知任务。准确检测远距离目标和遮挡目标是一项艰巨的挑战。根据传感器获取数据的类型，三维目标检测算法可分为基于图像的算法、基于点云的算法以及图像和点云融合的算法[2]。

基于图像的算法有模板匹配、几何约束、深度估计等。MF3D[3]通过子网络生成深度图，并将目标感兴趣区域与深度图融合来回归目标3D信息。AutoShape[4]引入关键点来建模物体的形状信息，利用形状信息提升单目3D检测性能。改进的Stereo-RCNN[5]采用确定

收稿日期：2023-04-03；**修回日期**：2023-04-26；**录用日期**：2023-08-02；**网络首发日期**：2023-08-15
基金项目：国家自然科学基金（62003288）、广东省基础与应用基础研究基金联合基金重点项目（2020B1515120050）
通信作者：*1017204759@qq.com

性网络作为主干网络，增强了对远距离目标的检测。改进的MonoDLE[6]通过学习实例深度与多尺度感知模块缓解了由于缺少深度信息带来的定位误差问题。2D图像能提供丰富的颜色和纹理信息，但缺少深度信息，因此很难准确估计目标的大小和位置，且对于远距离目标和遮挡目标的检测更为困难。

基于点云的算法通常分为基于点的算法和基于体素的算法。基于点的算法如PointRCNN[7]、3DSSD[8]等利用PointNet++[9]对输入点云进行特征提取，再处理下游任务，能达到较高的检测精度，但是效率较低，不适用于实时高效的大规模自动驾驶场景。基于体素的算法如VoxelNet[10]将输入点云体素化后，采用3D CNN对下游任务进行处理，计算规模和内存占用比较大。SECOND[11]将VoxelNet中的三维卷积层替换为稀疏卷积层，减少了无用的空体素计算，提高了检测效率。PointPillars[12]只在平面上进行体素化，进一步提高了计算效率。基于Swin Transformer结构的改进PointPillars[13]采用特征采样模块，提高了检测精度，改善了目标朝向判断不准的问题。基于体素的算法在体素化过程中损失了部分信息，对远处小目标及遮挡目标的检测效果通常低于基于点的算法，但具有更高的检测效率。

基于图像和点云融合的算法如F-PointNet[14]等先通过2D目标检测算法生成候选区域，再对候选区域内的点云进行3D目标检测。PI-RCNN[15]分别对图像和点云进行特征提取，再将图像特征和点云特征融合，实现最终的3D检测。此类算法传感器之间的视图错位问题很难解决，因此目前融合算法的检测精度不及基于激光雷达点云的算法。

SECOND算法能在保证实时性的同时取得较高的检测精度，本文以SECOND算法为基准，提出一种改进的SECOND算法，进一步提高算法精度。针对SECOND算法对于远处小目标及遮挡目标检测不准确的问题，采用自适应空间特征融合模块（ASFF）融合2D CNN的各层特征，加强对小目标的检测。改用three-dimensional distance-intersection over union（3D DIoU）损失替代Smooth L1[16]损失函数，改善对受遮挡目标的检测，同时在非极大值抑制（NMS）过程中，用分类置信度与交并比（IoU）相乘的结果作为候选框的评价标准，兼顾候选框的分类质量与定位质量，提高算法的整体定位精度。

2　SECOND算法简介

SECOND算法网络框架结构如图1所示，将点云转换成体素进行表征，使用稀疏卷积网络将3D稀疏数据转化为2D鸟瞰图，接着使用2D的backbone进行特征提取，包括点云体素化、体素特征提取、2D骨干网络、多任务检测头等4个部分，在KITTI验证集上的效果如图2所示，将3D预测边界框和真值目标框可视化，并将真值目标框投影到对应的图像中。其中，绿色框和黄色框分别表示car类和cyclist类的真值目标框，红色框表示预测边界框，可见SECOND算法对于远处的小目标和遮挡严重的目标检测效果并不理想。

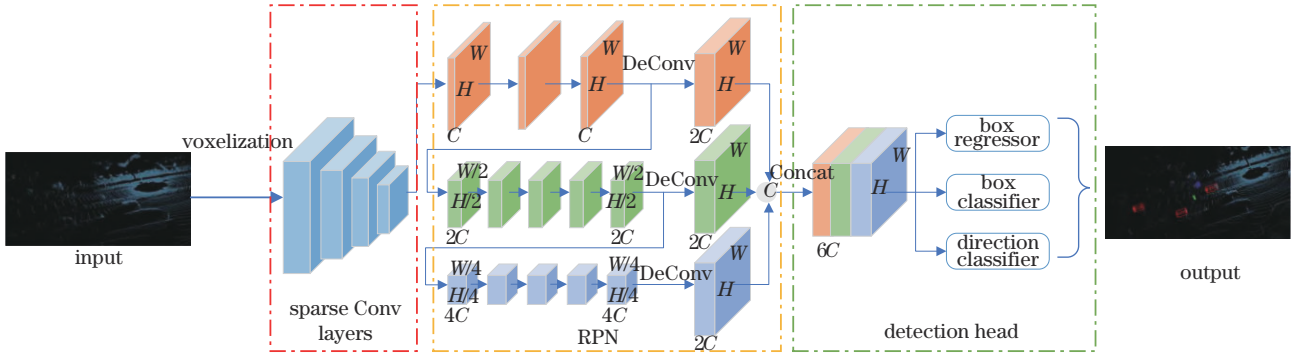

图1　SECOND算法整体网络框架结构

Fig. 1　The overall network framework structure of the SECOND algorithm

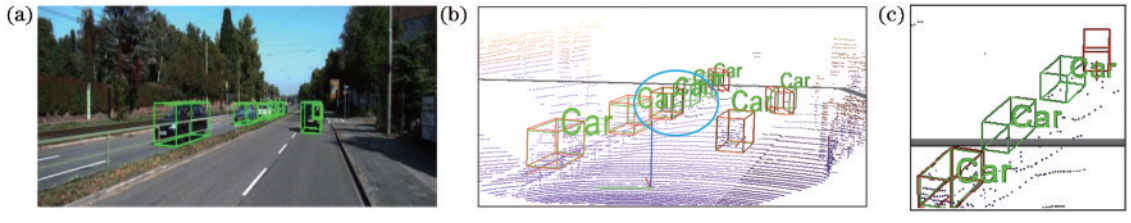

图2　SECOND算法在KITTI验证集上的检测结果。(a) 相机图像；(b) 点云视图；(c) 细节放大图

Fig. 2　Detection results of SECOND algorithm on KITTI validation set. (a) Camera image; (b) point cloud; (c) enlarged detail

3　改进的SECOND算法

SECOND算法采用点云体素化的表征方式，通过稀疏卷积层提取体素特征并将3D体素转化为2D伪图像，然后使用二维卷积网络提取特征，将目标定位与分类作为独立的任务，而NMS过程用分类置信度作为候选框质量的评价标准，使得算法仅考虑到候选框的分类质量，而没有考虑到定位质量。

3.1　2D CNN骨干网络改进

SECOND算法将经过稀疏卷积层得到的bird's eye view(BEV)特征图作为2D骨干网络的输入。2D骨干网络中，通过多次下采样得到3个不同尺度的特征，它们的形状分别是 $W\times H\times C$、$(W/2)\times(H/2)\times 2C$ 和 $(W/4)\times(H/4)\times 4C$。通过二维反卷积操作统一到相同的维度($W\times H\times 2C$)，然后直接进行拼接操作。低层特征分辨率高，包含更多位置、细节信息，语义性低，噪声多；而高层特征具有更强的语义信息，但是分辨率很低，对细节的感知能力较差。当某一物体在某一层次的特征图中被指定为正时，其他层次特征图中的相应区域被视为背景。如果一个场景同时存在大小物体，那么不同层次的特征之间的冲突往往会占据多尺度特征融合的主要部分，从而降低特征融合的有效性。所提算法通过ASFF模块在每个空间位置上自适应地融合不同尺度的特征，改进不同尺度特征之间存在不一致性的问题，即某些特征在该位置携带矛盾的信息，可能被过滤掉，而另一些更具判别性的特征占据主导地位。

ASFF模块如图3所示：首先，将反卷积操作输出的3个相同维度($W\times H\times 2C$)的特征图输入1×1×1的2D卷积中，得到3个空间权重向量，大小都是 $W\times H\times 1$；然后，沿通道方向拼接得到 $W\times H\times 3$ 的权重融合图；接着，在通道方向用Softmax进行归一化得到3个大小都是 $W\times H\times 1$ 的权重图，如式(1)所示，Softmax操作使得权重参数都在[0,1]内，同时建立3个特征权重之间的依赖关系，从而实现自适应特征融合；最后，将权重图作为对应特征的权重与特征进行逐元素乘操作，将新特征进行拼接融合后得到 $W\times H\times 6C$ 特征。

$$\mathrm{Softmax}(\mu,\nu,\gamma)=\left[\frac{\exp(\mu)}{\exp(\mu)+\exp(\nu)+\exp(\gamma)},\frac{\exp(\nu)}{\exp(\mu)+\exp(\nu)+\exp(\gamma)},\frac{\exp(\gamma)}{\exp(\mu)+\exp(\nu)+\exp(\gamma)}\right],\tag{1}$$

式中：μ、ν、γ 为同一位置的不同空间权重向量。

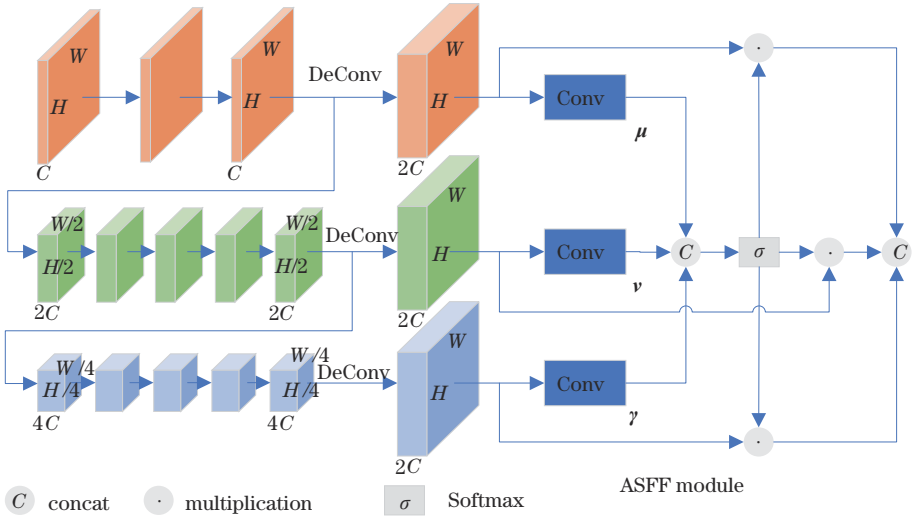

图3　改进后的2D CNN骨干网络

Fig. 3　The improved 2D CNN backbone network

3.2　3D DIOU损失

SECOND算法的3D边界框包含7个参数 (x,y,z,w,l,h,θ)，由于这7个参数是相互耦合的，使用Smooth L1损失单独优化这些参数可能会导致结果局部最优。通常用IoU作为预测边界框定位精度的度量标准，数值越大表明预测框越准确。图4中绿色框为目标边界框，蓝色框为预测边界框。当预测框中心点坐标 (x,y) 保持不变，长度 L 和宽度 W 的损失更小时，总损失也相应减小，然而IoU反而减小了，即回归结果变差了，因此需要将3D回归框作为一个整体进行回归。

在2D目标检测中，文献[17]引入IoU损失来代替Smooth L1损失进行边界框回归，对各种形状和尺度的物体具有鲁棒性，收敛速度快，实现了精确、高效的定位。文献[18]首先讨论了IoU在边界框不重叠情况下的缺点，然后提出了generalized intersection over

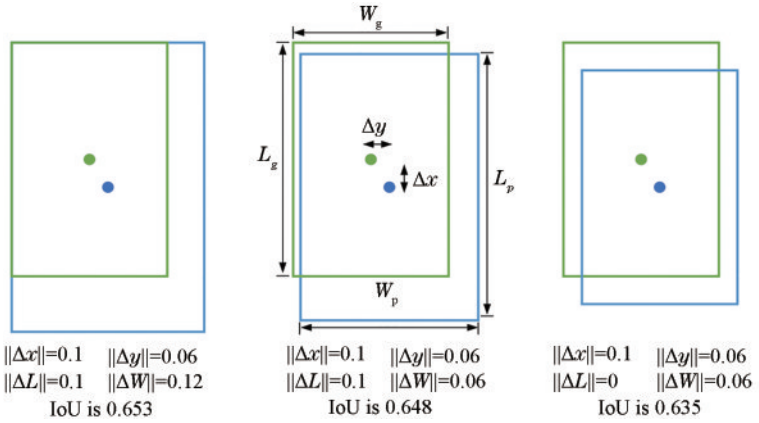

图 4　Smooth L1 损失局部最优示意图

Fig. 4　Illustration of the local optimum of smooth L1 loss

union（GIoU）作为一种新的损失，最后将 GIoU 集成到最先进的 2D 对象检测框架中，验证了其有效性。文献[19]提出了 DIoU 损失，通过最小化预测框和目标框之间的归一化距离，比 IoU 和 GIoU 损失收敛更快，效果更好。以上所有的工作都针对轴对齐的边界框回归任务。在图 5 所示的 3D 目标检测中，边界框是有朝向角的，即预测框与目标框并非轴对齐的，因此不能直接将 2D IoU 损失应用到 3D 目标检测中。文献[20]将 2D IoU 和 GIoU 损失扩展到 3D 目标检测变为 3D IoU 和 3D GIoU，提高了检测精度，改善了测试和训练之间的性能差距。

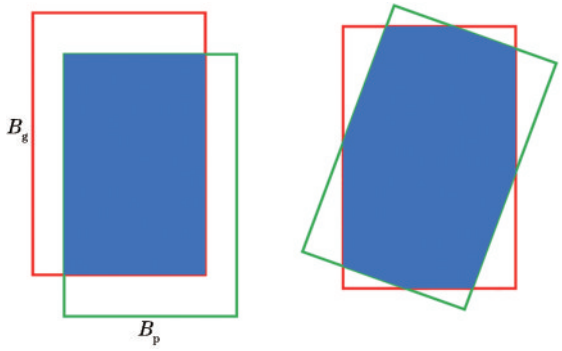

图 5　轴对齐边界框与非轴对齐边界框

Fig. 5　Axis-aligned and rotated bounding boxes

将 DIoU 损失应用到 3D 目标检测中，为了简化计算，首先定义覆盖预测框和目标框的最小封闭盒子的朝向角为 0，3D DIoU 俯视图如图 6 所示。其中：B_g 为真值目标框，其中心点为 O_1，角点为 A、E、F、G；B_P 为预测边界框，其中心点为 O_2，角点为 H、I、K、J。点 H、L、F、M、J 为 B_g 与 B_P 相交区域的顶点。矩形 $ABCD$ 为覆盖预测框和目标框的最小封闭盒子。定义 B_g 为 $(x_g, y_g, z_g, w_g, l_g, h_g, \theta_g)$，$B_P$ 为 $(x_p, y_p, z_p, w_p, l_p, h_p, \theta_p)$。其中，$x$、$y$、$z$ 表示边界框中心点坐标，w、l、h 分别表示边界框的宽、长、高，θ 为边界框的朝向角。3D DIoU 损失的具体计算流程如下：

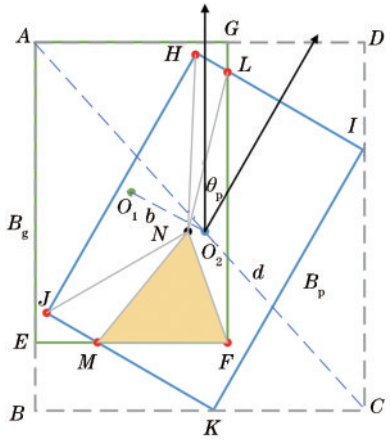

图 6　3D DIoU 俯视图（即 2D DIoU 示意图）

Fig. 6　Bird's eye view of 3D DIoU(2D DIoU schematic)

Input: $B_g(x_g, y_g, z_g, w_g, l_g, h_g, \theta_g)$, $B_P(x_p, y_p, z_p, w_p, l_p, h_p, \theta_p)$;

Output: B_g 与 B_p 的 3D DIoU；

1）计算 B_g 的体积，$V_g = w_g \times l_g \times h_g$；

2）计算 B_p 的体积，$V_p = w_p \times l_p \times h_p$；

3）计算 B_g 角点（点 A、E、F、G）的坐标；

4）计算 B_p 角点（点 H、I、K、J）的坐标；

5）计算相交区域顶点（点 H、L、F、M、J）的坐标，并以所有顶点坐标的均值为中心点（点 N）的坐标；

6）将相交区域中心点（点 N）与顶点（点 H、L、F、M、J）连接起来，相交区域被划分为 5 个三角形（△NFM、△NMJ、△NJH、△NJH、△NLF），相交区域的面积 A_i 即为所有三角形面积之和；

7）计算 B_g 与 B_p 的相交高度 h_i；

8）计算 3D IoU，$R_{\mathrm{IoU3D}}(B_p, B_g) = \dfrac{A_i \times h_i}{V_p + V_g - A_i \times h_i}$；

9）计算 B_g 与 B_p 中心点 $O_1(x_g, y_g, z_g)$、$O_2(x_p, y_p, z_p)$ 之间的欧氏距离，$b = \sqrt{(x_g - x_p)^2 + (y_g - y_p)^2 + (z_g - z_p)^2}$；

10）计算最小封闭盒子的对角线长度（即点 A 与点 C 的距离）d；

11）计算 B_g 与 B_p 的 3D DIoU，$R_{DIoU3D}(B_p, B_g) = R_{IoU3D}(B_p, B_g) - \frac{b^2}{d^2}$；

12）计算 3D DIoU 损失，$L_{DIoU3D} = 1 - R_{DIoU3D}(B_p, B_g)$。

3.3　候选框质量评价改进

使用预测的 IoU 值作为边界框的定位分数，如图 7 所示，在基准网络的多任务检测头中，增加一个分支用来预测目标框与候选框之间的 IoU，并使用 Smooth L1 损失函数对其进行优化，如式（2）所示：

$$L_{IoU} = \text{Smooth L1}(R_{IoUp} - R_{IoUt}), \tag{2}$$

式中：R_{IoUp} 表示候选框与目标框之间的交并比预测值；R_{IoUt} 表示候选框与目标框之间的交并比实际值。定义一个新的质量分数 f 来度量边界框的质量，如式（3）所示：

$$f = c \cdot R_{IoUp}, \tag{3}$$

式中：c 表示分类置信度，由分类分支得到。

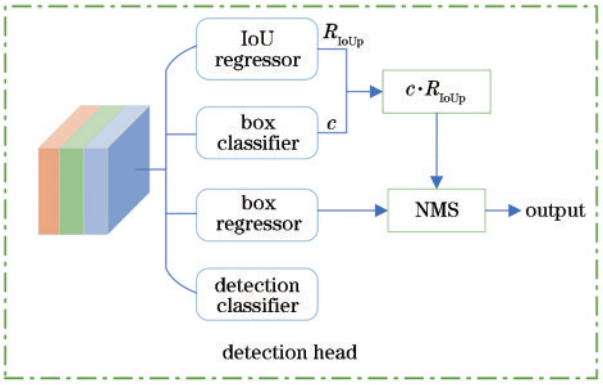

图 7　改进后的多任务检测头结构图

Fig. 7　The improved multi-task detection head

新增的 IoU 回归分支，通过学习预测算法检测到的每个边界框和匹配目标之间的 IoU。将预测得到的定位分数与分类置信度相乘后的结果作为新的质量分数 f，根据 f 进行非极大值抑制，确保定位更准确的边界框在 NMS 过程中被保留下来，从而改进 NMS 过程，提升检测定位精度。

3.4　算法的总损失函数设计

算法的损失函数分为 4 个部分：位置回归损失、方向损失、分类损失和 IoU 损失。

3.4.1　位置回归损失函数

将位置回归损失分为定位回归损失 $L_{reg\text{-}loc}$ 和角度回归损失 $L_{reg\text{-}\theta}$，其中，定位回归损失采用的是所提出的 DIoU 损失函数，定位回归损失 $L_{reg\text{-}loc}$ 的定义如式（4）所示：

$$L_{reg\text{-}loc} = 1 - R_{DIoU3D}(B_p, B_g)。 \tag{4}$$

采用正弦误差损失的角度回归方式，解决朝向角的预测方向和真实方向相反时损失剧增的问题，并能自然地根据角度偏移函数对 IoU 进行建模。角度回归损失 $L_{reg\text{-}\theta}$ 的定义如式（5）所示：

$$L_{reg\text{-}\theta} = \text{Smooth L1}\left[\sin(\theta_p - \theta_t)\right], \tag{5}$$

式中：θ_p 表示朝向角预测值；θ_t 表示朝向角真实值。

3.4.2　方向损失函数

采用 Softmax 损失函数作为方向损失 L_{dir}，解决角度回归损失不能区分朝向相反的边界框的问题。

3.4.3　分类损失函数

采用 Focal 损失函数作为分类损失，解决前景框和背景框的数量极其不均衡的问题，其定义如式（6）所示：

$$L_{cls} = -\alpha_t(1 - p_t)^{\gamma}\log(p_t), \tag{6}$$

式中：p_t 表示预测概率；α_t 和 γ 是 Focal 损失函数的超参数。为了和基准网络进行实验对比，超参数设置为与基准网络相同的值：$\alpha_t = 0.25$、$\gamma = 2$。

3.4.4　总损失函数

算法总损失函数定义如式（7）所示：

$$L_{total} = \alpha(L_{reg\text{-}loc} + L_{reg\text{-}\theta}) + \beta L_{cls} + \lambda L_{dir} + \omega L_{IoU}, \tag{7}$$

式中：α、β、λ 和 ω 表示不同任务的损失函数所占的权重。为了进行算法对比，采用与基准网络 SECOND 相同的权重系数：$\alpha = 2.0$、$\beta = 1.0$、$\lambda = 0.2$、$\omega = 1.0$。

改进后的 SECOND 算法结构如图 8 所示，框内为改进部分，框 1 为 ASFF 模块，框 2 为改进后的候选框质量评分。所提算法能够保留不同尺度特征的有用信息，式（7）定义的综合损失函数使得回归任务更加高

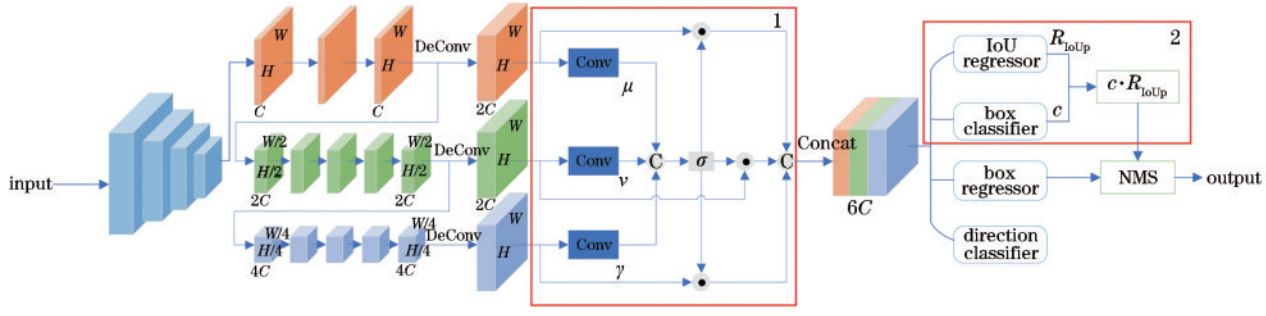

图 8　改进的 SECOND 算法结构

Fig. 8　Improved SECOND algorithm structure

效，NMS 阶段考虑候选框的定位质量，提升了算法的整体定位精度。

4 实验与结果分析

4.1 测试数据集

本实验使用 KITTI 公开数据集[21]，将 KITTI 数据集划分为 7481 个训练样本和 7518 个测试样本。在训练过程中，将训练样本进一步划分为训练集（3712 个样本）和验证集（3769 个样本）。根据目标尺寸、遮挡和截断程度，数据集被划分为简单、中等和困难等 3 个级别。

4.2 实验设置

实验使用 PyTorch 深度学习框架和 OpenPCDet 目标检测框架，Intel(R) Xeon(R) Silver 4210 处理器，RTX 3090GPU、Ubuntu 16.04，CUDA 11.1 版本，CUDNN 8.0.4 版本。训练过程采用 Adam 优化器更新模型权重。初始学习率为 0.003，指数衰减因子为 0.8，每 15 个时期衰减一次，衰减权重为 0.01，β_1 值为 0.9，β_2 值为 0.99。

4.3 评价指标

常用平均精度（AP11 和 AP40）作为衡量指标。本实验使用 AP40 在 KITTI 测试集上进行评估，使用 AP11 在 KITTI 验证集上进行评估。

4.4 不同算法对比实验

由于测试服务器的访问次数限制，其他方法在测试集上的检测精度结果均来自 KITTI 官方排行榜单，而验证集上的检测精度是在本实验室硬件平台上复现所得的。

表 1 是 KITTI 官方采用 AP40 计算的，所提算法在 car 类简单、中等和困难难度级别上的检测精度分别比 SECOND 算法提升 2.86 百分点、2.99 百分点、7.06 百分点，在 cyclist 类的简单、中等和困难难度级别上提升 3.84 百分点、3.89 百分点、4.27 百分点。对于 car 类和 cyclist 类的 BEV 检测，所提算法在 3 个难度级别上优于大部分现有的算法。

表 1　不同算法在 KITTI 测试集上的检测精度结果对比

Table 1　Comparison of detection accuracy with different algorithms on KITTI test set　　unit: %

Algorithm	car-3D			car-BEV			cyclist-3D			cyclist-BEV		
	easy	mod	hard	easy	mod	hard	easy	mod	hard	easy	mod	hard
Autoshape[4]	22.47	14.17	11.36	30.66	20.08	15.95						
PointRCNN[6]	86.96	75.64	70.70	92.13	87.39	82.72	74.96	58.82	52.53	82.56	67.24	60.28
3DSSD[7]	**88.36**	79.57	74.55	92.66	89.02	**85.86**	**82.48**	64.10	56.90	**85.04**	67.62	61.14
VoxelNet[9]	77.47	65.11	57.73	89.35	79.26	77.39	61.22	48.36	44.37	66.70	54.76	50.55
SECOND[10]	84.65	75.96	68.71	91.81	86.37	81.04	75.83	60.82	53.67	79.21	64.26	56.61
PointPillars[11]	82.58	74.31	68.99	90.07	86.56	82.81	77.10	58.65	52.92	79.90	62.73	55.58
F-PointNet[14]	82.19	69.79	60.59	91.17	84.67	74.77	72.27	56.12	49.01	77.26	61.37	53.78
PI-RCNN[15]	84.37	74.82	70.03	91.44	85.81	81.00						
AVOD-FPN[22]	83.07	71.76	65.73	90.99	84.82	79.62	63.76	50.55	44.93	69.39	57.12	51.09
DVFENet[23]	86.20	79.18	74.58	90.93	87.68	84.60	78.73	62.00	55.18	82.29	67.40	60.71
IA-SSD[24]	88.34	**80.13**	75.04	**92.79**	**89.33**	84.35	78.35	61.94	55.70	81.30	66.29	59.58
Proposed algorithm	87.51	78.95	**75.77**	92.09	88.32	85.47	79.67	**64.71**	**57.94**	83.04	**69.61**	**62.65**

表 2 采用 AP11，所提算法检测精度比 SECOND 算法提高近 1 百分点。从表 2 可以看出，所提算法每秒检测帧数（FPS）达到 20.41，远高于 PointRCNN 等算法。PointRCNN 等算法在点云处理阶段需要多次下采样操作与近邻搜索，时间复杂度分别为 $O(n^2)$ 和 $O(n)$；SECOND 等算法只需要将点云划分到对应的体素中，时间复杂度为 $O(n)$，因此基于体素的算法检测效率高于基于点的算法。所提算法在 SECOND 的基础上引入了 ASFF 模块，增加了时间消耗，故检测效率略低于 SECOND 算法。

表 2　不同算法在 KITTI 验证集上的实验结果对比

Table 2　Comparison of experimental results of different algorithms on the KITTI validation set　　unit: %

Algorithm	Modality	FPS	car-3D		
			easy	mod	hard
F-PointNet	Lidar+RGB	9.52	83.76	70.92	63.65
SECOND	Lidar(Voxel-based)	21.74	88.16	78.18	77.04
PointRCNN	Lidar(Point-based)	9.84	88.32	78.26	77.34
PI-RCNN	Lidar(Point-based)	10.16	88.21	78.53	77.72
Proposed algorithm	Lidar(Voxel-based)	20.41	89.02	79.03	78.01

图 9 将点云中的 3D 预测边界框和真值目标框进行了可视化，绿色框表示 car 类的真值目标框，黄色框表示 cyclist 类的真值目标框，红色框表示预测边界框。可以看出，所提算法可以检测到一些没有标记的目标，对远处的小目标、遮挡截断严重的目标能达到较好的检测效果。从图 10 可以看出，所提算法对于遮挡的目标和远处的目标（蓝色椭圆内）识别效果有所提升。

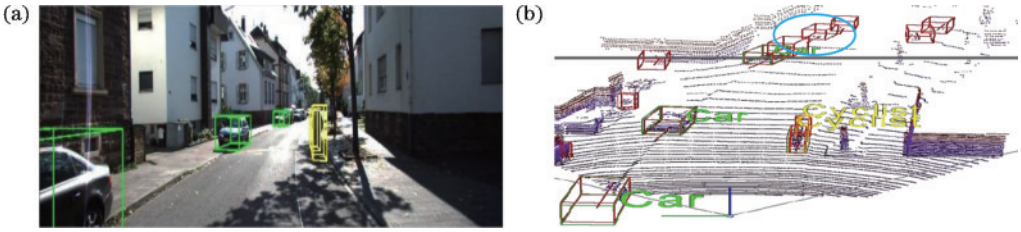

图 9　所提算法在 KITTI 验证集上的定性结果。(a) 相机图像；(b) 点云视图

Fig. 9　Qualitative results of proposed algorithm on KITTI validation set. (a) Camera image; (b) point cloud

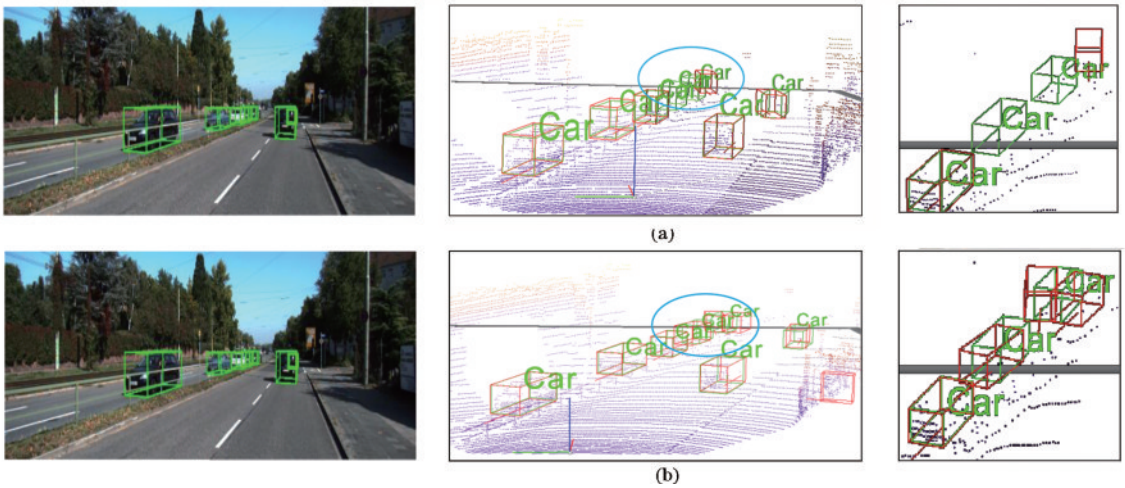

图 10　所提算法与基准网络（SECOND）定性结果对比。(a) SECOND 算法可视化结果；(b) 所提算法可视化结果

Fig. 10　Comparison of qualitative results between the proposed algorithm and the benchmark network (SECOND). (a) Visualization results of SECOND algorithm; (b) visualization results of proposed algorithm

4.5　消融实验

4.5.1　ASFF 模块消融实验

为了验证 ASFF 模块对模型检测精度的影响，在 SECOND 算法和 PointPillars 算法的 2D CNN 骨干网络中添加 ASFF 模块，并在 KITTI 验证集上进行消融实验，结果如表 3 所示。添加 ASFF 模块的改进 SECOND 算法 car 类 3D 检测精度在简单、中等和困难难度下分别提高 0.17 百分点、0.19 百分点和 0.12 百分点，cyclist 类 3D 检测精度分别提升了 2.79 百分点、2.07 百分点和 2.76 百分点。cyclist 类的 BEV 检测精度提升幅度明显高于 car 类的提升幅度，可见 ASFF 模块对小目标的检测提升效果更为明显。添加 ASFF 模块的 PointPillars 算法各类目标的检测精度相比原算法均有所提升，证明了模块的有效性。SECOND 算法和 ASFF 模块可视化结果如图 11 所示，绿色框表示 car 类的真值目标框，黄色框表示 cyclist 类的真值目标框，红色框表示预测边界框。从图 11 可以看出，ASFF 模块能够提升小目标（蓝色椭圆内）的识别效果。

表 3　ASFF 模块在 KITTI 验证集上的消融实验

Table 3　Ablation experiments of ASFF module on the KITTI validation set　　unit: %

Method	car-3D			car-BEV			cyclist-3D			cyclist-BEV		
	easy	mod	hard	easy	mod	hard	easy	mod	hard	easy	mod	hard
SECOND	88.16	78.18	77.04	89.76	87.74	86.52	79.96	67.34	62.44	85.54	70.96	66.68
SECOND+ASFF	88.33	78.37	77.16	90.10	87.82	86.51	82.75	69.41	65.20	87.25	72.24	68.23
PointPillars	86.93	77.18	75.20	89.78	87.41	84.19	80.34	64.59	61.21	83.21	68.31	63.98
PointPillars+ASFF	87.80	77.74	75.23	90.15	87.75	86.04	81.90	65.09	61.53	85.86	70.80	65.69

为了验证 ASFF 模块对多尺度特征融合的作用，对空间权重进行可视化，如图 12 所示，颜色越蓝权重值越接近 0，颜色越红权重值越接近 1，这 3 组权重在每个位置上的和都是 1。从图中可以看出：浅层特征

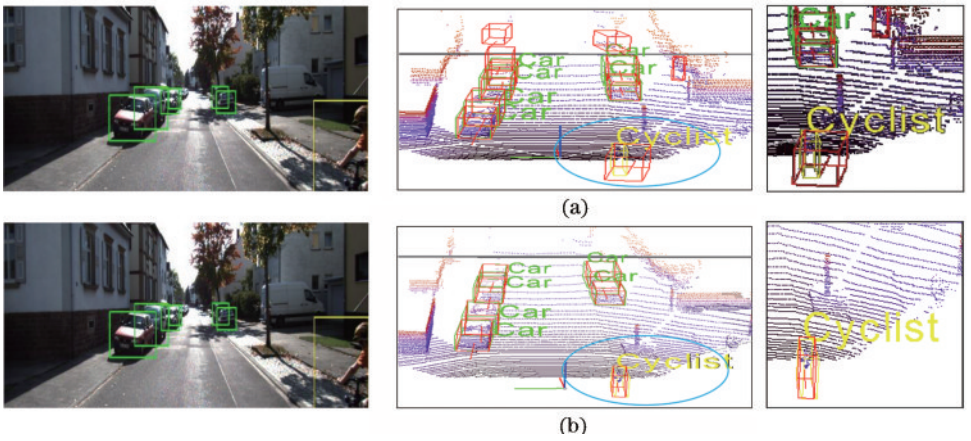

图 11　两种算法定性结果对比。(a) SECOND 算法可视化；(b) SECOND 算法＋ASFF 可视化

Fig. 11　Comparison of qualitative results of two algorithms. (a) Visualization of SECOND algorithm; (b) visualization of SECOND algorithm＋ASFF

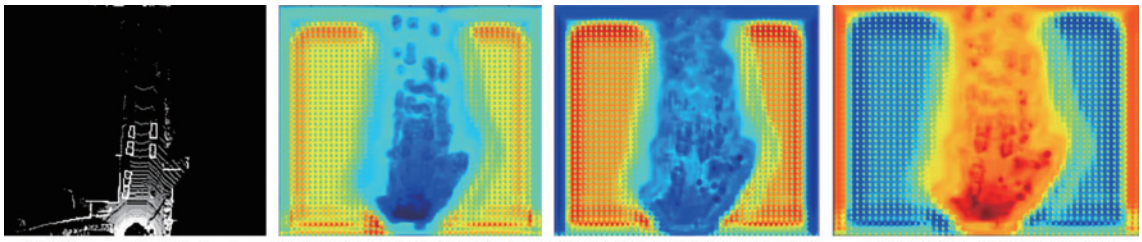

图 12　空间权重可视化

Fig. 12　Visualization of spatial weight

和中间层特征对目标区域关注度低，因此权重小；而深层特征对目标区域关注度高，权重更大。特征融合后，深层特征占据主导地位，从而提升目标检测效果。

4.5.2　不同损失函数的消融实验

为了验证 DIoU 损失对模型检测精度的影响，将 SECOND 算法的定位回归损失函数分别替换为 3D IoU 损失、3D GIoU 损失和 3D DIoU 损失后，在 KITTI 验证集上进行消融实验，结果如表 4 所示。

表 4　不同损失函数在 KITTI 验证集上的消融实验

Table 4　Ablation experiments of different loss functions on the KITTI validation set　　unit: %

Loss	car-3D			car-BEV			cyclist-3D			cyclist-BEV		
	easy	mod	hard	easy	mod	hard	easy	mod	hard	easy	mod	hard
Smooth L1	88.16	78.18	77.04	89.76	87.74	86.52	79.96	67.34	62.44	85.54	70.96	66.68
3D IoU	88.51	78.60	77.51	90.02	87.99	86.70	81.91	69.02	64.22	87.89	71.92	67.79
3D GIoU	88.48	78.69	77.57	89.96	88.05	86.74	82.56	69.34	64.87	88.13	72.21	68.26
3D DIoU	88.49	78.77	77.72	89.98	88.09	86.81	83.22	69.87	65.76	88.66	72.57	68.72

表 4 表明，3D DIoU 损失明显优于其他 3 种损失函数，car 类 3D 检测精度在简单、中等和困难难度下分别提升 0.33 百分点、0.59 百分点和 0.68 百分点，cyclist 类 3D 检测精度分别提升 3.26 百分点、2.53 百分点和 3.32 百分点。car 类和 cyclist 类的 BEV 检测精度都有一定提升，可见 3D DIoU 损失对两类目标的提升效果较为明显。由于 Smooth L1 损失需要对 3D 边界框的各个参数单独进行优化，室外场景距离较远且存在遮挡，很难从稀疏点中获取足够的信息来精确预测边界框的所有维度，故效果不佳。3D IoU 损失将边界框作为一个整体进行回归，能更好地反映边界框重合程度，但在边界框不重叠时会出现梯度消失的问题。3D GIoU 损失缓解了非重叠情况下的梯度消失问题，但当目标框完全包裹预测框时，无法区分其相对位置关系。3D DIoU 损失直接最小化边界框之间的距离，将更多的注意力集中在边界框中心点的回归，能有效解决上述的问题，使得最终回归结果更为准确。

4.5.3　改进后的候选框质量评分消融实验

为了验证改进后的候选框质量分数对模型检测精度的影响，将候选框的质量评分由分类分数替换为分

类分数与定位分数相乘后，在KITTI验证集上进行消融实验，结果如表5所示，可知car类3D检测精度在简单、中等和困难难度下分别提升0.02百分点、0.4百分点和0.46百分点，cyclist类3D检测精度分别提升2.45百分点、1.34百分点和2.12百分点。基准网络仅使用分类分数来评价候选框的质量，没有考虑到定位质量，而改进后的质量评分同时考虑到了分类和定位，因此回归效果更好。

表5 候选框质量评分在KITTI验证集上的消融实验

Table 5 Ablation experiments of bounding box quality score on the KITTI validation set unit: %

Score	car-3D			car-BEV			cyclist-3D			cyclist-BEV		
	easy	mod	hard	easy	mod	hard	easy	mod	hard	easy	mod	hard
c	88.16	78.18	77.04	89.76	87.74	86.52	79.96	67.34	62.44	85.54	70.96	66.68
$c\cdot R_{\mathrm{IoUp}}$	88.18	78.58	77.50	89.84	87.87	86.57	82.41	68.67	64.56	86.85	71.88	67.19

4.5.4 改进的SECOND算法消融实验

上述实验结果表明，ASFF模块、3D DIoU损失与改进后的候选框质量评分能提高目标检测精度，将它们同时应用在SECOND算法中，消融实验结果如表6所示。其中，Baseline是采用基准网络SECOND的2D CNN，检测精度采用AP11，c表示分类分数，R_{IoUp}表示定位分数。

表6表明，在引入ASFF模块、3D DIoU损失与改进后的候选框质量评分后，car类3D检测精度在简单、中等和困难难度下分别提升0.86百分点、0.85百分点和0.97百分点，cyclist类3D检测精度分别提升6.49百分点、4.41百分点和5.17百分点，优于原始SECOND算法。

表6 改进的SECOND算法在KITTI验证集上的消融实验

Table 6 Ablation experiments of improved SECOND algorithm on the KITTI validation set unit: %

Backbone	Loss	Score	car-3D			car-BEV			cyclist-3D			cyclist-BEV		
			easy	mod	hard	easy	mod	hard	easy	mod	hard	easy	mod	hard
Baseline	Smooth L1	c	88.16	78.18	77.04	89.76	87.74	86.52	79.96	67.34	62.44	85.54	70.96	66.68
		$c\cdot R_{\mathrm{IoUp}}$	88.18	78.58	77.50	89.84	87.87	86.57	82.41	68.67	64.56	86.85	71.88	67.19
	3D DIoU	c	88.49	78.77	77.72	89.98	88.09	86.81	83.22	69.87	65.76	88.66	72.57	68.72
		$c\cdot R_{\mathrm{IoUp}}$	88.77	78.89	77.98	90.11	88.03	86.94	85.35	70.77	66.88	89.77	73.46	69.94
ASFF	Smooth L1	c	88.33	78.37	77.16	90.10	87.82	86.51	82.75	69.41	65.20	87.25	72.24	68.23
		$c\cdot R_{\mathrm{IoUp}}$	88.47	78.65	77.60	89.91	87.95	86.66	84.36	70.11	66.08	88.92	72.82	68.89
	3D DIoU	c	88.67	78.81	77.74	90.06	88.17	86.87	84.82	70.26	66.35	89.34	73.15	69.28
		$c\cdot R_{\mathrm{IoUp}}$	89.02	79.03	78.01	90.20	88.27	87.26	86.45	71.75	67.61	91.32	74.12	70.77

5 结 论

在SECOND算法2D CNN骨干网络的基础上，提出一个自适应空间特征的ASFF模块，在每个空间位置上自适应地融合不同尺度的特征，再通过最小化预测框和目标框之间的归一化距离，采用3D DIoU损失替代Smooth L1回归损失函数，使得回归任务更加高效。最后，提出一个新的候选框质量评价标准f，兼顾分类质量和定位质量两个指标，使得后续的NMS操作保留更平滑的结果。在KITTI数据集的实验结果表明，改进算法在检测性能超过SECOND、PointPillars、PointRCNN等经典算法，在car类的困难级别及cyclist类的中等、困难级别上与3DSSD、DVFENet、IASSD相比，检测精度具有明显优势，在实时性超过PointRCNN、F-PointNet等经典的基于点的点云三维目标检测方法。改进的SECOND算法对点云进行体素化操作，不可避免地造成了信息丢失。后续考虑采用多种传感器数据融合以提高算法的鲁棒性，或者采用点与体素相结合的方法进一步提高激光雷达点云数据的检测性能。

参 考 文 献

[1] 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.
Wang Y D, Tian Y L, Li G Q, et al. 3D object detection based on convolutional neural networks: a survey[J]. Pattern Recognition and Artificial Intelligence, 2021, 34(12): 1103-1119.

[2] 刘训华, 孙韶媛, 顾立鹏, 等. 基于改进Frustum PointNet的3D目标检测[J]. 激光与光电子学进展, 2020, 57(20): 201508.
Liu X H, Sun S Y, Gu L P, et al. 3D object detection based on improved frustum PointNet[J]. Laser & Optoelectronics Progress, 2020, 57(20): 201508.

[3] Xu B, Chen Z Z. Multi-level fusion based 3D object detection from monocular images[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 2345-2353.

[4] Liu Z D, Zhou D F, Lu F X, et al. AutoShape: real-time

shape-aware monocular 3D object detection[C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV), October 10-17, 2021, Montreal, QC, Canada. New York: IEEE Press, 2022: 15621-15630.

[5] 于洁潇, 张美琪, 苏育挺. 基于双目视觉的三维车辆检测算法[J]. 激光与光电子学进展, 2021, 58(2): 0215004.
Yu J X, Zhang M Q, Su Y T. Three-dimensional vehicle detection algorithm based on binocular vision[J]. Laser & Optoelectronics Progress, 2021, 58(2): 0215004.

[6] 王凤随, 熊磊, 钱亚萍. 联合实例深度的多尺度单目3D目标检测算法[J]. 激光与光电子学进展, 2023, 60(16): 1612002.
Wang F S, Xiong L, Qian Y P. Multiscale monocular three-dimensional object detection algorithm combined with instance depth[J]. Laser & Optoelectronics Progress, 2023, 60(16): 1612002.

[7] Shi S S, Wang X G, Li H S. PointRCNN: 3D object proposal generation and detection from point cloud[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2020: 770-779.

[8] Yang Z T, Sun Y N, Liu S, et al. 3DSSD: point-based 3D single stage object detector[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 11037-11045.

[9] Qi C R, Yi L, Su H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space [EB/OL]. (2017-06-07)[2023-03-02]. https://arxiv.org/abs/1706.02413.

[10] Zhou Y, Tuzel O. VoxelNet: end-to-end learning for point cloud based 3D object detection[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 4490-4499.

[11] Yan Y, Mao Y X, Li B. SECOND: sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.

[12] Lang A H, Vora S, Caesar H, et al. PointPillars: fast encoders for object detection from point clouds[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2020: 12689-12697.

[13] 陈德江, 余文俊, 高永彬. 基于改进PointPillars的激光雷达三维目标检测[J]. 激光与光电子学进展, 2023, 60(10): 1028012.
Chen D J, Yu W J, Gao Y B. Lidar 3D target detection based on improved PointPillars[J]. Laser & Optoelectronics Progress, 2023, 60(10): 1028012.

[14] Qi C R, Liu W, Wu C X, et al. Frustum PointNets for 3D object detection from RGB-D data[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 918-927.

[15] Xie L A, Xiang C, Yu Z X, et al. PI-RCNN: an efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12460-12467.

[16] Girshick R. Fast R-CNN[C]//2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2016: 1440-1448.

[17] Yu J H, Jiang Y N, Wang Z Y, et al. UnitBox: an advanced object detection network[C]//Proceedings of the 24th ACM international conference on Multimedia, October 15-19, 2016, Amsterdam, The Netherlands. New York: ACM Press, 2016: 516-520.

[18] Rezatofighi H, Tsoi N, Gwak J, et al. Generalized intersection over union: a metric and a loss for bounding box regression[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2020: 658-666.

[19] Zheng Z H, Wang P, Liu W, et al. Distance-IoU loss: faster and better learning for bounding box regression[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12993-13000.

[20] Zhou D F, Fang J, Song X B, et al. IoU loss for 2D/3D object detection[C]//2019 International Conference on 3D Vision (3DV), September 16-19, 2019, Quebec City, QC, Canada. New York: IEEE Press, 2019: 85-94.

[21] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite [C]//Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 16-21, 2012, Providence, Piscataway. New York: ACM Press, 2012: 3354-3361.

[22] Ku J, Mozifian M, Lee J, et al. Joint 3D proposal generation and object detection from view aggregation [C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 1-5, 2018, Madrid, Spain. New York: IEEE Press, 2019.

[23] He Y Q, Xia G H, Luo Y K, et al. DVFENet: dual-branch voxel feature extraction network for 3D object detection[J]. Neurocomputing, 2021, 459: 201-211.

[24] Zhang Y F, Hu Q Y, Xu G Q, et al. Not all points are equal: learning highly efficient point-based detectors for 3D LiDAR point clouds[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 18-24, 2022, New Orleans, LA, USA. New York: IEEE Press, 2022: 18931-18940.