激光与光电子学进展 先进成像

基于极端稀疏激光点云和RGB图像的3D目标检测

秦超[1,2]，王亚飞[1]，张宇超[2]，殷承良[1*]

[1]上海交通大学机械与动力工程学院，上海 200240；

[2]上海智能网联汽车技术中心有限公司，上海 201499

摘要 复杂交通场景下的3D目标检测是重要且具有挑战性的任务。针对主流检测算法使用的高线数激光雷达昂贵和基于毫米波雷达和相机的检测算法效果不佳的问题，提出了一种利用低线数激光雷达和相机实现3D目标检测的算法，可以大幅降低自动驾驶的硬件成本。首先，将64线激光雷达点云降采样至原始点云数量的10%，生成极端稀疏点云，并将其和RGB图片一同输入到深度补全网络中得到深度图；然后，在新提出的计算点云强度的算法基础上，由深度图生成点云俯视图；最后，将点云俯视图输入检测网络，得到目标立体边界框的几何信息、航向角和类别等信息。在KITTI数据集上对算法进行实验验证，实验结果表明所提算法在检测精度上可以超过部分基于高线数激光雷达的检测算法。

关键词 遥感；激光点云；卷积神经网络；关键点检测；深度学习

中图分类号 TP751 **文献标志码** A **DOI**：10.3788/LOP202259.1828004

3D Object Detection Based on Extremely Sparse Laser Point Cloud and RGB Images

Qin Chao[1,2], Wang Yafei[1], Zhang Yuchao[2], Yin Chengliang[1*]

[1]*School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai* 200240, *China*;

[2]*Shanghai Intelligent and Connected Vehicle R&D Center Co., Ltd., Shanghai* 201499, *China*

Abstract The task of detecting 3D objects in complex traffic scenes is crucial and challenging. To address the high-cost problem of high-definition LiDAR and the poor effect of detection algorithms based on the millimeter wave radar and cameras used in mainstream detection algorithms, this study proposes a 3D target detection algorithm using low-definition LiDAR and a camera, which can significantly reduce the hardware cost of autonomous driving. To obtain a depth map, the 64-line LiDAR point cloud is first downsampled to 10% of the original point clouds, resulting in an extremely sparse point cloud, and fed to the depth-completion network with RGB images. Then, a point cloud bird-eye view is generated from the depth map based on the proposed algorithm for calculating the point cloud intensity. Finally, the point cloud bird-eye view is fed into the detection network to obtain the geometric information, heading angle, and category of the target stereo bounding box. The different algorithms are experimentally validated using KITTI dataset. The experimental results demonstrate that the proposed algorithm can outperform some conventional high-definition LiDAR-based detection algorithms in terms of detection accuracy.

Key words remote sensing; laser point cloud; convolutional neural network; key point detection; deep learning

1 引 言

在复杂交通场景中，车辆与行人的目标检测对自动驾驶技术具有重要意义[1]。目标检测作为车辆和路侧感知系统的重要组成部分[2]，其输出的检测结果将被用于后续的多目标跟踪、碰撞预警和路径规划等多个技术模块。目前主流的3D目标检测算法可根据所使用的传感器输入分为三类：利用单目相机的RGB图片作为网络输入的检测算法；利用高线数激光雷达采集的稠密点云作为网络输入的检测算法；同时使用RGB图片和稠密点云作为网络输入的多模态检测算法。以上三类算法在近年来随着计算机视觉技术的快速发展均取得了重大进步。基于相机的2D图像检测取得巨大进展。由于相机只能提供二维测量信息，并

收稿日期：2021-08-03；**修回日期**：2021-08-10；**录用日期**：2021-08-23

基金项目：国家自然科学基金(52072243)、四川省科技计划(2020YFSY0058)

通信作者：*clyin@sjtu.edu.cn

且测距误差较大，而激光雷达可以提供包含三维坐标信息的点云数据，能够提供更为丰富的几何信息并且测距误差也更为精确，因此基于激光雷达的目标检测开始受到众多学者的关注。

基于图像的3D目标检测是在已经成熟的2D目标检测技术基础上逐渐发展而来的，RGB图片的优势在于可以提供丰富的语义信息和纹理信息，而其固有缺点在于损失了一个维度的几何信息。当前基于图像的神经网络算法可以分为三类。第一类是有候选框生成阶段的双阶段目标检测算法，例如Fast R-CNN[3]和Faster R-CNN[4]。算法先对图像提取候选框，然后基于候选区域进行二次修正得到检测结果，检测精度较高，但检测速度较慢。第二类是单阶段目标检测算法。YOLOv1[5]舍去了算法中的候选框提取分支，直接将特征提取、候选框分类和回归在同一个无分支的深度卷积网络中实现，简化了网络结构，Faster R-CNN的检测速度从7 frame/s提升到了45 frame/s，能够满足实时检测任务的需求。YOLO检测算法在后续研究中不断得到改进，YOLOv2算法[6]主要利用批归一化、高分辨率分类器分类、直接目标框位置检测、多尺度训练等操作来提高模型的检测精度。YOLOv3[7]在YOLOv2的基础上，使用全新设计的Darknet-53残差网络并结合特征金字塔网络(FPN)[8]进行多尺度融合预测。YOLOv4[9]将骨干网络融合CSPNet算法[10]，保证检测精度的同时降低了网络计算量，同时将特征金字塔网络加入空间金字塔池化层，改善了浅层特征丢失问题。第三类是无锚框(anchor free)的关键点检测算法。CornerNet[11]提出了基于角点的检测方法，网络从特征图中预测物体的左上角和右下角这一对角点，从而得出物体的边界框，引入角点池化(corner pooling)使角点定位更加精确。CornerNet-Lite[12]在CornerNet的基础上引入了注意力机制，以减少像素处理个数，从而提高了检测速度。CenterNet[13]利用目标边界框的中心点来表示所检测的目标，通过关键点估计来找到中心点，并从特征图中回归目标的尺寸、位置甚至姿态等其他属性。在相同速度的条件下，CenterNet的精度比YOLOv3高了4个百分点。

相比于相机，激光雷达点云可以提供物体精确的三维坐标信息，尤其是高线数的激光雷达可以较好地刻画物体的轮廓信息，同时由于点云的稀疏、无序和位置敏感等特点使其难以直接套用图像检测算法，因此利用激光雷达进行3D目标检测一直是学术界的热点领域。点云3D目标检测网络按照点云的组织方式主要可以分为三类：第一类是将原始点云编码为体素形式，然后输入神经网络进行检测；第二类是直接基于原始点云进行特征学习，然后进行检测；第三类是将点云编码成图像形式，然后输入神经网络进行目标检测。

在基于体素的模型中，VoxelNet[14]将三维空间按照指定间隔划分网格，按照是否位于同一网格的要求对原始点云进行分组，然后使用体素特征编码层对网格中的点云抽取特征，最后组合多尺度特征以提升网络学习的能力。SECOND[15]提出了一种改进的稀疏卷积网络，显著提高了训练和预测的速度，引入新的航向角损失函数，提高了航向角的检测精度。PointRCNN[16]直接用原始点云进行检测，检测过程分为两个阶段：先使用PointNet++[17]作为一个前景分割模型，分割出的前景作为3D的提案(proposal)；第二阶段将得到的提案和上一阶段的特征一起作为模型的输入，得到精准的边界框回归。3DSSD[18]提出了新的下采样融合策略，替换掉PointNet++中的全局特征上采样模块，大幅减少了计算损耗，在提高检测精度的同时提高了速率。为了利用图像领域较为成熟的卷积神经网络技术，一些学者采用将点云编码成图像的方式，将点云投影成2D图像的投影方式有以下几种：鸟瞰图(BEV)投影[19]、前视图投影[20]和圆柱投影[21]。Complex-yolo[19]先将三维点云栅格化，在鸟瞰图空间的网格中分别计算网格内点集的最大高度、最大强度、点云密度，然后将三种信息归一化后分别填充到RGB三个通道中形成RGB-Map，最后采用YOLOv2的Darknet19进行特征提取并进行目标检测。目前在点云编码成图像的方式中主要采用鸟瞰图投影，采用其他投影方式的算法比较少。

作为测量物体不同物理属性的两类传感器，相机和激光雷达分别提供了关于物体的两种异质信息，两者各有优缺点并且具有一定的互补性。相机缺少的关于物体的精确几何信息可以由激光雷达提供，而激光雷达缺少的关于物体的纹理和语义信息可以由相机提供。因此如何将两者结合以提高目标检测效果的多模态检测算法逐渐成为新的研究方向。Frustum PointNets[22]是学术界较早利用激光雷达和相机融合进行3D目标检测的神经网络，其先使用卷积网络由图像生成2D的边界框(bounding box)，再通过平截头体(frustum)的方法由边界框映射出一个3D的候选区域，然后使用卷积网络对上一阶段的候选区域进行实例分割和最终3D边界框回归。AVOD[23]首先使用特征提取网络分别从图片和点云鸟瞰图中提取特征，然后一个区域提议网络利用两个模态的特征生成3D候选区，基于候选区对图像特征和点云鸟瞰图特征进行融合，最后将融合结果输入到后续检测网络中进行目标检测。虽然基于激光雷达和相机融合的检测算法取得了较好的效果，但在公开数据集上仅使用激光雷达的检测算法仍然优于基于融合的算法，并未体现出融合的优越性。为填补这一空白，PointPainting[24]将激光雷达点云投影到图像语义分割网络所输出的语义分割图像上，然后将每个点所对应的语义类别信息附加在每个点的坐标和反射强度的后面，将附加了语义信息的点云送入后续的目标检测网络中。在KITTI数据集上，PointPainting分别将融合后的点云送入Point-

RCNN、VoxelNet、PointPillars三个经典的基于激光雷达的目标检测网络，实验结果表明三个网络的目标检测结果均优于之前输入非融合点云的结果。

目前基于高线数激光雷达的3D目标检测算法已经取得了较高的性能，但高线数激光雷达高昂的成本制约了其在自动驾驶中的应用。另一方面，低线数激光雷达和相机虽然成本较低，但其检测效果与高线数激光雷达相比仍然有较大差距。而目前学术界鲜有基于低线数激光雷达和相机融合的算法以解决上述问题。

针对高线数激光雷达昂贵和基于相机的检测算法效果不佳的问题，本文提出了基于稀疏点云和RGB图片的卷积神经网络模型用于3D目标检测。所提3D目标检测网络由三个部分组成：深度补全网络；深度图生成点云俯视图（Bird Eye View）模块；基于关键点特征金字塔的目标检测网络。对KITTI数据集上的每帧激光雷达点云进行随机降采样，将每帧点云数量稀疏化到原始点云数量的10%，使点云数量低于低线数激光雷达；将稀疏点云和RGB图像输入所提目标检测网络，得到输出目标的3D检测信息，如立体边界框的大小、中心点坐标和航向角等。在KITTI数据集上评估所提目标检测网络，实验结果表明所提算法在检测精度上大幅超过基于单目相机的3D目标检测算法，并且超过了经典的基于激光点云的3D目标检测算法。综上所述，所提算法的主要贡献在于：1）提出了一种结合深度补全和关键点特征金字塔的3D目标检测算法，该算法可以利用低成本的低线数激光雷达和相机取得与基于高线数激光雷达的3D目标检测算法相当的检测效果，降低自动驾驶感知系统的硬件成本；2）提出了一种新的方法，利用深度补全网络输出的置信度图生成点云强度值，从而实现了由深度图生成点云俯视图的目的；3）研究和分析影响算法性能的深层次因素，包括点云的数量、点云的组织方式、稀疏点云的作用等。

2　3D目标检测算法框架

2.1　算法框架

所提3D目标检测网络框架如图1所示。该网络由深度补全网络、深度图生成点云俯视图网络、基于关键点特征金字塔的目标检测网络三个部分组成。网络的输入为经过降采样后的稀疏化点云和RGB图像，网络的输出为3D目标检测信息，包括中心点三维坐标、立体边界框的长宽高、航向角和所属类别等信息。

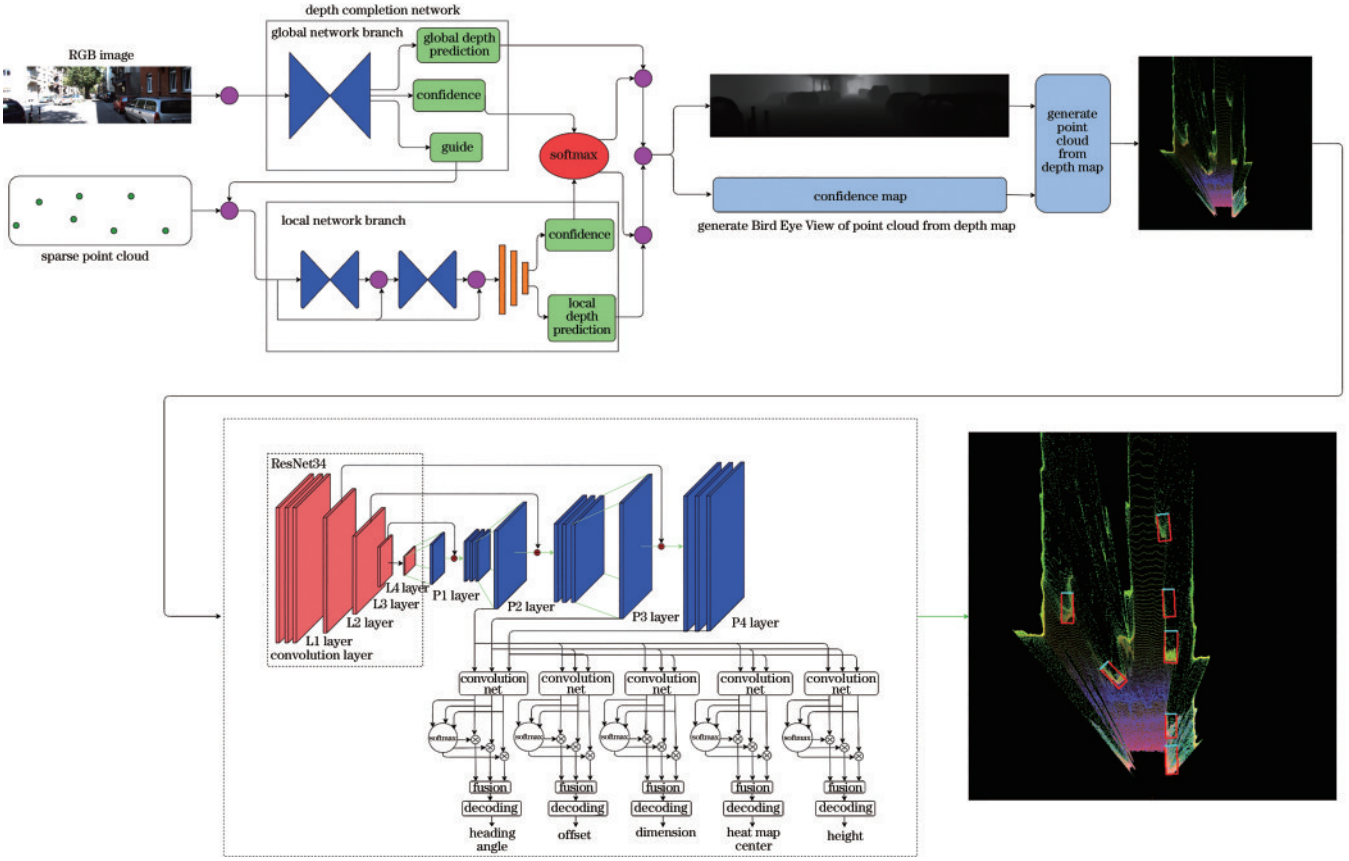

图1　所提3D目标检测算法框架

Fig. 1　Structure of proposed 3D object detection algorithm

2.2　深度补全网络

2.2.1　网络结构

基于 van Gansbeke 等[25]提出的网络结构，设计了一个深度补全网络，网络框架如图 2 所示。网络的输入为 RGB 图片和点云图片，其中点云图片是将 3D 点云投影到 2D 平面上得到的，网络通过融合全局网络生成的全局信息和局部网络生成的局部信息得到生成深度图。

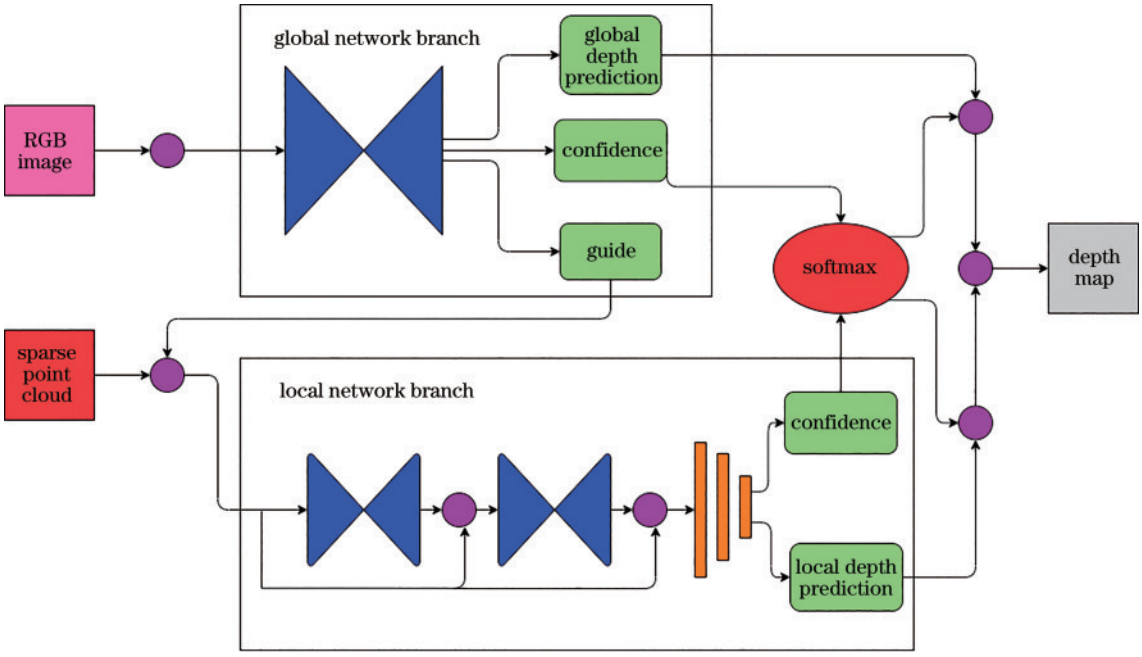

图 2　深度补全网络

Fig. 2　Depth completion network

图 2 显示的整体网络结构分为两个部分：上半部分是基于 ERFNet[26]的编码-解码（encoder-decoder）全局网络分支；下半部分是采用堆叠沙漏网络的局部网络分支，该分支由两个沙漏网络组成，每个沙漏网络有 6 个卷积层，以学习原始深度预测的残差。

ERFNet 在最初文献中被用于解决自动驾驶中的语义分割问题，其实时性和精度均比较突出，在全局网络中可以利用其提供的语义分割信息。全局网络分支输出三个特征图：带有全局信息的指导图、一张深度图、一张置信度图。而局部网络输出两个映射：一张深度图和一张置信度图。将全局深度图和置信度图相乘得到全局预测，将局部深度图和置信度图相乘得到局部预测，然后将两张预测图相加，得到最终的深度预测图。

2.2.2　输入点云降采样

深度补全网络的输入点云为经降采样的极端稀疏点云，降采样过程为：设原始稠密点云中点的个数为 N，随机地从 N 个点中选出 $0.1\times N$ 个点，即降采样后的稀疏点云数量只有原始点云的 10%。为可视化稀疏点云，分别将原始稠密点云和稀疏点云投影到图像上，如图 3 所示，其中上图显示的是原始稠密点云投影到图像上的效果，下图显示的是稀疏点云投影到图像上的效果。

图 3　稠密点云和稀疏点云分别投影到图像上的结果

Fig. 3　Results of dense point cloud and sparse point cloud projected on the image respectively

2.2.3　损失函数

为了衡量最终输出深度图、全局深度图和局部深度图的损失，采用 focal-MSE 损失函数，计算公式为

$$\lambda(\hat{y},y)=\frac{1}{n}\sum_{i=1}^{n}\left(1+0.05\cdot E_{\text{epoch}}\cdot\left|y_i-\hat{y}_i\right|\right)\cdot\left(y_i-\hat{y}_i\right)^2,\tag{1}$$

$$\Lambda=\omega_1\cdot\lambda\left(\hat{y}_{\text{global}},y\right)+\omega_2\cdot\lambda\left(\hat{y}_{\text{local}},y\right)+\omega_3\cdot\lambda\left(\hat{y}_{\text{out}},y\right),\tag{2}$$

式中：y 和 $\hat{y}$ 分别表示预测深度图和真值深度图中每个像素点的深度值；$\hat{y}_{\text{global}}$、$\hat{y}_{\text{local}}$ 和 $\hat{y}_{\text{out}}$ 分别表示全局深度

图、局部深度图和最终输出深度图的损失；ω为每类深度图损失分配的权重。式(1)用于计算单张预测深度图和标签真值之间的损失，式(2)是计算最终深度图、全局深度图和局部深度图的总损失。在模型开始训练时对三个权重给定初始值：$w_1=1, w_2=1, w_3=1$。始终固定$w_3=1$并保持w_1和w_2相等，在实验中不断调整w_1以加快损失下降速度，最终得到权重的一组优化值：$w_1=0.1, w_2=0.1, w_3=1$。

2.3　由深度图生成点云俯视图

深度补全网络输出的是深度图而目标检测网络需要的是激光点云信息，因此需要由深度图生成激光点云。

首先根据射影几何的原理将深度图上每个点的2D坐标和深度信息转换为点云的三维坐标，生成后的点云图像如图4所示。相机内参矩阵为

$$\boldsymbol{K}=\begin{bmatrix} f_x & S & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

式中：f_x和f_y分别是相机在x和y方向上的焦距；S为坐标轴倾斜参数。设点云坐标为(x,y,z)，像素坐标为(u,v)，深度图上每个点的深度为z，则三维点云坐标为

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}=z\begin{bmatrix} 1/f_x-S/\left(f_xf_y\right) & \left(Sc_y-c_xf_y\right)/\left(f_xf_y\right) \\ 0 & 1/f_y & -c_yf_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \\ 1/z \end{bmatrix}。 \tag{4}$$

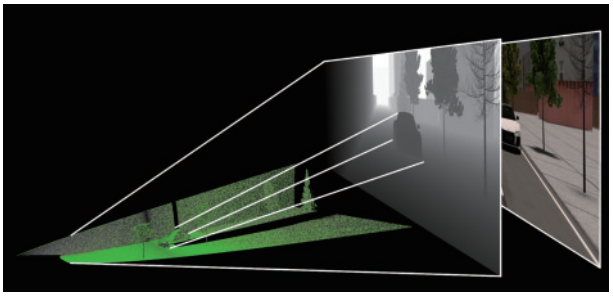

图4　深度图生成点云图像

Fig. 4　Point cloud image generated from depth map

真实世界中激光雷达产生的点云数据除了三维坐标信息以外还包括每个点的强度信息。而激光雷达反射强度受多重因素影响，包括扫描角度、距离、物体表面结构、粗糙度及空气湿度等，而在深度图生成点云过程中需要计算点云中每个点的强度值。对激光雷达的点云数据进行分析可知，点云的强度值随着点云与激光雷达的距离增加而逐渐降低，即越远的点强度值越低。而在实验过程中，观察到深度补全网络输出的置信度图中每个点的置信度值也遵循随着距离增加而逐渐降低的规律，因此认为深度补全网络输出的每个点的置信度信息与点云的强度存在相关性，可以由深度补全网络输出点的置信度值得到点云强度。深度补全网络输出的置信度值范围为(0,1)，而一般激光雷达的强度值为(0,100)。所提算法将深度补全网络输出的每个点的置信度乘以100后的值设定为该点的强度信息，最终可以得到点云中每个点的三维坐标信息和强度信息；然后将点云投射到2D网格平面上，并以0.1 m的精度离散化。在每个单元格中，高度值取单元格中最高点的点云的高度，强度值取最高点的反射强度值。经过上述处理，可以得到编码为高度、强度和密度的点云BEV图，其中密度表示的是每个网格中的点云数量。转换得到的BEV图将作为后续目标检测网络的输入。

2.4　基于关键点特征金字塔的3D目标检测网络

2.4.1　网络结构

受到RTM3D[27]启发，设计了基于关键点特征金字塔的3D目标检测网络。网络的输入为点云俯视图，输出为目标3D边界框的中心点坐标、长宽高、航向角和所属类别等信息。基于关键点特征金字塔的3D目标检测网络结构如图5所示，主要由骨架网络、特征金字塔网络和检测头网络组成。

为了既保证检测精度又提高运行速度，骨架网络采用了ResNet34[28]。同时借鉴了Lin等提出的特征金字塔网络(FPN)，FPN算法可以对浅层与深层的特征图进行融合，利用邻近特征图的语义信息，通过融合上下两层的特征，得到语义信息更加丰富的特征图供后续的检测。通过融合这些不同层的特征图来进行预测，解决了多尺度下小目标准确检测的问题。因此，所提算法在骨架网络后面也借鉴了特征金字塔网络的算法思想。如图5所示，使用双线性插值将ResNet网络的L4层特征图上采样，得到P1层特征图；再对L3层特征图和P1层特征图进行拼接操作，得到新的特征图；使用1×1的卷积核对拼接后的特征图进行卷积操作，并使用双线性插值将输出的特征图上采样，从而得到特征图P2；重复上述步骤可以得到P3特征图，然后对P3特征图和L2特征图进行拼接操作，并使用1×1的卷积核对其进行卷积计算，经过卷积层后得到特征图P4。

本模型总计5个检测头(head)网络，每个检测头网络由卷积网络、softmax函数和解码模块组成。其中卷积网络由三个部分组成：一个卷积核大小为3×3的卷积层，卷积层的输入通道数等于特征金字塔网络中特征层的通道数，卷积层的输出通道数为64；接着是ReLU函数激活层；最后是卷积核大小为1×1的卷积层，该卷积层的输入通道数等于64，输出通道数则取决于每个检测头网络的功能，例如用来预测目标3D立体边界框的头部网络的输出通道数为3，其分别用来预测长度、宽度和高度。特征金字塔网络中表示不同尺度信息的P2、P3和P4特征图首先被输入到检测头网络中的卷积网络层，每张特征图经过卷积网络层后生成新的特征图。需要将三个表示不同尺度信息的新特征图融合成一张特征图，因此这里使用归一化指数

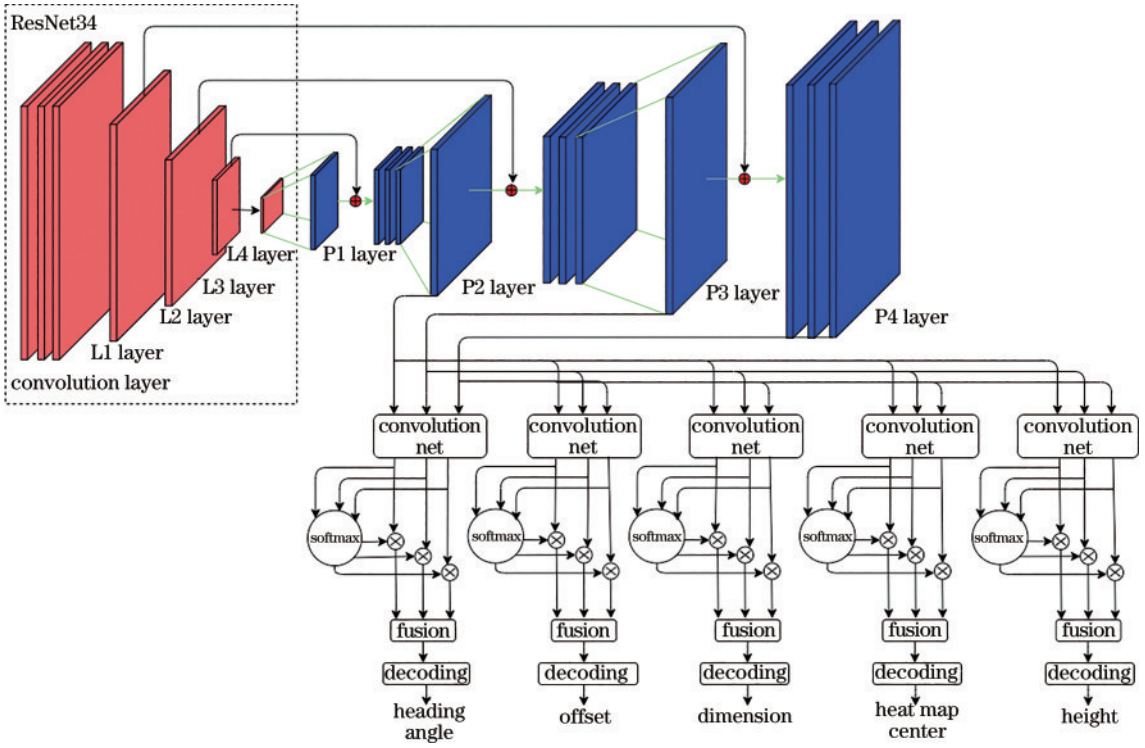

图 5　基于关键点特征金字塔的 3D 目标检测网络

Fig. 5　3D object detection network based on key point feature pyramid

函数(softmax)来计算出每个新特征图的权重值;再将各新特征图和对应的权重值相乘并累加求和,就得到一张特征图,该特征图将被送入解码模块以回归出边界框长宽高和航向角等信息。

2.4.2　损失函数

将 BEV 图上边界框中心点的真值标签使用高斯核函数表示,设中心点的真值为(x,y),则高斯核函数为 $Y_{xy}=\exp\left[-\frac{\left(x-\tilde{p}_x\right)^2+\left(y-\tilde{p}_y\right)^2}{2\sigma_{\mathrm{p}}^2}\right]$,其中标准差$\sigma_{\mathrm{p}}$根据物体实际大小进行自适应调整。采用焦点损失(Focal loss)函数计算中心点的损失,公式为

$$L_{\mathrm{Focal}}=\frac{-1}{N}\sum_{xy}\begin{cases}\left(1-\hat{Y}_{xy}\right)^{\alpha}\log\hat{Y}_{xy}, & Y_{xy}=1\\ \left(1-\hat{Y}_{xy}\right)^{\beta}\hat{Y}_{xy}^{\alpha}\log\left(1-\hat{Y}_{xy}\right), & Y_{xy}\neq 1\end{cases},\tag{5}$$

式中:α,β均为超参数;Y_{xy}和$\hat{Y}_{xy}$分别表示边界框中心点的真值高斯核函数和预测高斯核函数。设航向角的预测值为$\hat{\theta}$,真值为θ,使用L1损失函数计算航向角在训练时的损失,公式为

$$L_1=\frac{1}{N}\sum_{k=1}^{N}\left|\theta_k-\hat{\theta}_k\right|。\tag{6}$$

采用平衡L1损失(balanced L1 loss)函数计算中心点高度和边界框长宽高的回归损失,公式分别为

$$L=\begin{cases}\frac{1}{N}\sum_{k=1}^{N}\frac{\alpha}{b}\left(b\left|z_k-\hat{z}_k\right|+1\right)\ln\left(b\left|z_k-\hat{z}_k\right|+1\right)-\alpha b\left|z_k-\hat{z}_k\right|, & \left|z_k-\hat{z}_k\right|<1\\ \gamma\left|z_k-\hat{z}_k\right|, & \left|z_k-\hat{z}_k\right|\geqslant 1\end{cases},\tag{7}$$

$$L=\begin{cases}\frac{1}{N}\sum_{q\in\{w,\,h,\,l\}}\sum_{k=1}^{N}\frac{\alpha}{b}\left(b\left|d_k^q-\hat{d}_k^q\right|+1\right)\ln\left(b\left|d_k^q-\hat{d}_k^q\right|+1\right)-\alpha b\left|d_k^q-\hat{d}_k^q\right|, & \left|d_k^q-\hat{d}_k^q\right|<1\\ \gamma\left|d_k^q-\hat{d}_k^q\right|, & \left|d_k^q-\hat{d}_k^q\right|\geqslant 1\end{cases},\tag{8}$$

式中:z表示中心点高度;d^q,$q\in\{w,h,l\}$分别表示边界框的长度、宽度和高度;α和γ是可调节的超参,$b=\mathrm{e}^{\gamma/\alpha}-1$。

3　分析与讨论

所提模型在公开数据集 KITTI[29] 上进行训练和评估,所使用的硬件为配置有 I7-10700K CPU 和英伟达 3080 GPU 的 PC 机。

3.1　模型训练

KITTI 数据集由德国卡尔斯鲁厄理工学院和丰田美国技术研究院共同制作,包含了城市、乡村和高速公路等多个场景下的图像和点云数据,是目前国际上

较大的面向自动驾驶场景的计算机视觉算法评测数据集。本文中的深度补全网络在KITTI深度补全平台上进行训练，深度补全数据集提供了85898张图片用于训练，1000张图片用于评价。目标检测网络在KITTI目标检测平台上进行训练，目标检测训练集总共包含了7481张RGB图片和对应的点云。将目标检测训练集分为两部分：6000张RGB图片和点云用于训练，1481张图片和点云用于测试。

首先训练深度补全网络两次。第一次是在KITTI深度补全数据集上进行训练，数据集提供的真实值标签为半稠密深度图；然后采用知识蒸馏的思想，将数据集上所有的照片和点云送入第一次训练好的深度补全网络，生成稠密的深度图并作为第二次训练的真实值标签，稠密深度图如图6所示。在第二次训练中，首先对数据集中的点云进行降采样，使每份点云个数只有原始点云的10%；然后将极端稀疏化后的点云和图像送入深度补全网络，并以稠密深度图作为标签训练深度补全网络。

图6　深度补全网络输出的稠密深度图

Fig. 6　Dense depth map generated from depth completion network

针对目标检测网络的训练，首先将KITTI目标检测数据集中6000份用于训练的点云随机降采样，使得每帧点云中点的数量只有原始点云的10%；再将6000份极端稀疏化后的点云和对应的RGB图片送入已经训练好的深度补全网络，得到6000份稠密的深度图和对应的置信度图；然后使用所提方法生成稠密深度图和置信度图对应的点云俯视图；最后使用点云俯视图作为目标检测网络的输入，以KITTI目标检测数据集提供的真值作为标签训练目标检测网络。

3.2　实验结果

首先将目标检测测试集的1481份点云随机降采样，使得每帧点云中点的数量只有原始点云的10%，将降采样后的稀疏点云投影到对应的图片上，效果如图7所示。然后将稀疏点云和图片输入到已训练好的深度补全网络，以得到稠密深度图，深度图如图8所示。设稠密深度图中每个像素点的坐标为(u,v,z)，u，v表示像素点在图片坐标系中的像素坐标，z表示像素点的深度，则利用相机的内参和外参矩阵可得到该点

图7　稀疏点云投影到图像上的结果

Fig. 7　Result of sparse point cloud projection on the image

图8　深度补全网络输出的稠密深度图

Fig. 8　Dense depth map generated from depth completion network

的三维坐标(x,y,z)。根据2.3节的论述，将深度补全网络输出的每个像素点的置信度值e作为该点的强度值并附加在三维坐标之后，即(x,y,z,e)。深度图上的所有像素点经上述步骤后即可生成稠密的点云。使用2.3节描述的方法将稠密点云投影到鸟瞰图上，即得到一张三通道的点云BEV图，如图9所示。将点云BEV图输入到已训练好的基于关键点特征金字塔的3D目标检测网络，得到3D目标检测结果。

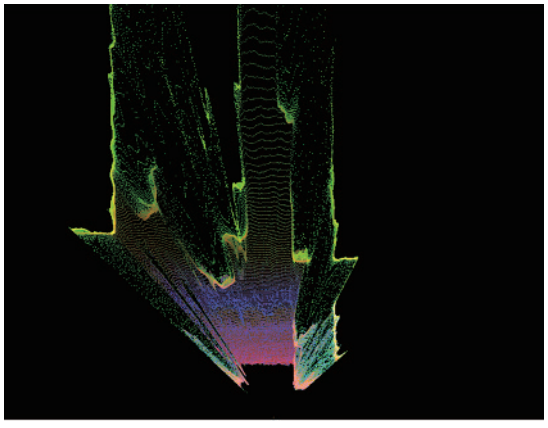

图9　稠密点云经鸟瞰图投影后生成的点云BEV图

Fig. 9　BEV map generated from dense point cloud after aerial view projection

检测结果按照KITTI数据集格式保存，并使用KITTI提供的评估程序计算网络在不同物体类别上的3D平均精度(AP)，结果如表1所示，其中IOU为交并比。在AP计算中，会按照目标类别例如汽车等分别进行计算，并且在每个类别下分别给出容易、中等和

表1　所提算法在KITTI数据集上的目标检测精度

Table 1　Target detection accuracy of proposed algorithm on KITTI dataset　　unit: %

Algorithm	Car (IOU is 0.7)			Person (IOU is 0.5)			Bicycle (IOU is 0.5)		
	Easy	Moderate	Difficult	Easy	Moderate	Difficult	Easy	Moderate	Difficult
Proposed algorithm	87.98	77.14	73.33	45.97	38.94	35.81	68.12	55.25	53.55

困难三个不同难度下的AP值。根据检测结果进行可视化显示，其中图10表示目标边界框在BEV俯视图中的效果，图11表示将目标立体边界框投影到RGB图片上的效果。

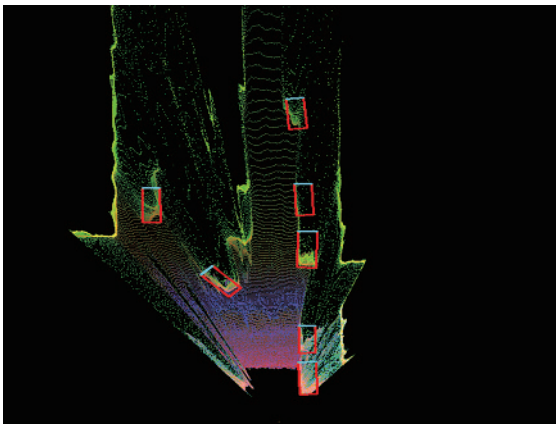

图10　目标检测结果在点云BEV图上的显示效果

Fig. 10　Detection result on BEV map

3.3　算法检测精度影响因素分析与实验

为分析模型输入数据组织形式对检测算法精度的影响，分别进行以下对比实验：1）将稠密点云经降采样后得到的极端稀疏点云直接转为点云BEV图，然后将其输入所提基于关键点特征金字塔的3D目标检测网络，得到3D目标检测结果；2）与所提算法流程的第一部分相同，即将稀疏点云和图片输入到深度补全网络，得到稠密深度图，进而通过相机的内参和外参矩阵得到稠密点云，将点云的三维坐标(x,y,z)以前视图的形式编码成三通道输入到基于关键点特征金字塔的3D目标检测网络，得到3D目标检测结果；3）只将图片输入到深度补全网络，得到稠密深度图，然后将深度图转换为点云BEV图后输入到基于关键点特征金字塔的3D目标检测网络，得到3D目标检测结果；4）将点云按不同比例分别降采样，得到不同稀疏程度的稀疏点云，将稀疏点云和图片输入到深度补全网络，得到稠密深度图，然后将深度图转换为点云BEV图后输入到基于关键点特征金字塔的3D目标检测网络，得到3D目标检测结果。

3.3.1　稀疏点云作为3D目标检测网络的输入

为验证点云数量对所提模型检测精度的影响，将原始的64线激光雷达采集的稠密点云降采样，使得稀疏点云数量只有原始点云的10%。设点云的坐标表示为$P=\left\{\left(x^{(t)},y^{(t)},z^{(t)}\right)\right\}_{t=1}^{K}$，其中$K$表示稀疏点云总个数，$\left(x^{(t)},y^{(t)},z^{(t)}\right)$表示每个点云在激光雷达坐标系下的坐标，其中$z$轴方向垂直向上。通过移除点云的$z$轴坐标，可将点云垂直投影到BEV图上。设BEV图尺寸为$H_{\mathrm{BEV}}\times W_{\mathrm{BEV}}$，将BEV图组织为三通道形式，每个通道分别表示点云的强度、密度和高度。对于BEV图上的一个像素点(x,y)，有$T(x,y)=c^{j}$，其中j表示通道个数，c^{j}表示该通道上对应像素点的值。将稀疏点云生成的三通道BEV图输入到所提基于关键点特征金字塔的3D目标检测网络，得到目标检测结果，在车辆、行人和自行车上的检测精度如表2所示。表2每行数值表示该实验在KITTI数据集上不同物体类别和不同难度条件下的AP，其中表2首行是模型在输入为稀疏点云条件下的平均检测精度，第二行为所提算法的实验结果。

图11　目标检测立体边界框在相机RGB图片上的显示效果

Fig. 11　Display effect of object detection stereo bounding box on camera RGB pictures

表2　在稀疏点云BEV图作为关键点特征金字塔网络输入的条件下的目标检测精度

Table 2　Target detection accuracy under the condition of sparse point cloud BEV as the input of key point feature pyramid network　　unit: %

Input	Car (IOU is 0.7)			Person (IOU is 0.5)			Bicycle (IOU is 0.5)		
	Easy	Moderate	Difficult	Easy	Moderate	Difficult	Easy	Moderate	Difficult
Sparse point cloud	4.50	3.15	2.88	0.96	0.94	0.9	0.82	0.78	0.70
Proposed algorithm	87.98	77.14	73.33	45.97	38.94	35.81	68.12	55.25	53.55

3.3.2 深度补全生成的稠密点云按照前视图形式编码然后作为目标检测网络的输入

为分析点云组织形式对模型检测效果的影响，首先将降采样后的稀疏点云和相机图片输入到所提算法中的深度补全网络，得到稠密深度图。设稠密深度图尺寸为 $H_{\mathrm{Depth}}\times W_{\mathrm{Depth}}$，其中每个像素点的坐标为$(u,v,z)$，其中$(u,v)$表示点的像素坐标，$z$表示该点的深度，则通过式(4)可计算出每个点在相机坐标系下的三维坐标(x,y,z)。为将点云以前视图形式编码，按照稠密深度图尺寸建立三通道图片 $3\times H_{\mathrm{Depth}}\times W_{\mathrm{Depth}}$，将稠密深度图上每个像素点的三维坐标$(x,y,z)$分别填入三通道图片上的每个通道。将此三通道图片输入到所提基于关键点特征金字塔的目标检测网络，其在车辆、行人和自行车上的检测精度如表3所示。表3每行数值表示该实验在KITTI数据集上不同物体类别和不同难度条件下的AP，其中表3首行表示模型在点云以前视图形式编码的条件下的平均检测精度，第二行为所提算法的实验结果。

表3 在以前视图形式编码点云条件下的目标检测精度

Table 3 Target detection accuracy under the condition of coded point cloud in previous view form unit: %

Input	Car (IOU is 0.7)			Person (IOU is 0.5)			Bicycle (IOU is 0.5)		
	Easy	Moderate	Difficult	Easy	Moderate	Difficult	Easy	Moderate	Difficult
Point cloud organized as (x,y,z)	60.25	52.80	45.60	35.86	31.38	29.72	41.82	39.56	36.09
Proposed algorithm	87.98	77.14	73.33	45.97	38.94	35.81	68.12	55.25	53.55

3.3.3 相机图片作为深度补全网络的输入

为分析稀疏点云对所提算法检测精度的影响，在实验中并未将稀疏点云作为输入而是直接将图片单独作为深度补全网络的输入，得到稠密深度图；然后由稠密深度图生成点云BEV图，并将其输入到基于关键点特征金字塔的3D目标检测网络，得到目标检测结果。其在车辆、行人和自行车上的检测精度如表4所示。表4每行数值表示该实验在KITTI数据集上不同物体类别和不同难度条件下的AP，表4首行表示模型在仅将图片作为输入条件下的平均检测精度，第二行为所提算法的实验结果。

表4 在仅将图片作为深度补全网络输入条件下的目标检测精度

Table 4 Target detection accuracy under the condition of only taking the picture as the input of depth complement network unit: %

Input	Car (IOU is 0.7)			Person (IOU is 0.5)			Bicycl e(IOU is 0.5)		
	Easy	Moderate	Difficult	Easy	Moderate	Difficult	Easy	Moderate	Difficult
Image	34.52	21.04	19.03	22.24	13.56	12.26	5.63	3.43	3.10
Proposed algorithm	87.98	77.14	73.33	45.97	38.94	35.81	68.12	55.25	53.55

3.3.4 不同降采样率下的稀疏点云和图片作为深度补全网络的输入

为分析点云的降采样率对所提算法检测精度的影响，在实验中将原始点云按照1%，6%，8%，10%，12%分别降采样，生成不同稀疏程度的稀疏点云。将稀疏点云和图片输入到深度补全网络生成稠密深度图，然后由稠密深度图生成点云BEV图，并将其输入到基于关键点特征金字塔的3D目标检测网络，得到目标检测结果。其在车辆、行人和自行车上的检测精度如表5所示。表5每行表示在特定点云降采样率下，所提算法在三个类别上的检测精度。

表5 不同点云降采样率下的目标检测精度

Table 5 Target detection accuracy under different point cloud down sampling rates

Down sampling rate	Car (IOU is 0.7)			Person (IOU is 0.5)			Bicycle (IOU is 0.5)		
	Easy	Moderate	Difficult	Easy	Moderate	Difficult	Easy	Moderate	Difficult
1%	44.30	32.03	28.32	31.56	22.36	20.79	16.27	12.85	11.50
6%	73.54	60.17	55.80	38.91	33.40	29.95	48.2	40.76	39.80
8%	81.85	69.84	67.80	41.20	35.21	31.90	53.52	47.92	43.77
10%	87.98	77.14	73.33	45.97	38.94	35.81	68.12	55.25	53.55
12%	88.20	78.26	73.65	46.01	39.70	36.11	68.80	55.73	54.02

3.4 结果分析与讨论

3.4.1 目标检测精度分析与讨论

表6显示了其他学者提出的算法和所提算法在KITTI数据集上的实验结果，其中包括汽车、行人和自行车三个目标类别。表6所列其他学者的检测算法分为三类：基于高线数激光雷达点云的3D目标检测算法、基于单目相机的3D目标检测算法、基于高线数激光雷达和相机的双模态算法。所提算法在汽车类目标

表 6 3D目标检测算法在KITTI测试集上的检测结果对比

Table 6 Comparison of 3D object detection algorithms on KITTI dataset unit: %

Algorithm	Modality	Car (IOU is 0.7)			Person (IOU is 0.5)			Bicycle (IOU is 0.5)		
		Easy	Moderate	Difficult	Easy	Moderate	Difficult	Easy	Moderate	Difficult
VoxelNet	64-line LiDAR	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
SECOND	64-line LiDAR	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
PointRCNN	64-line LiDAR	85.94	75.76	68.32	49.43	41.78	38.63	73.93	59.60	53.59
SS3D[30]	Camera	10.78	7.68	6.51	2.31	1.78	1.48	2.80	1.45	1.35
D4LCN[31]	Camera	16.65	11.72	9.51	4.55	3.42	2.83	2.45	1.67	1.36
AVOD	64-line LiDAR+camera	76.39	66.47	60.23	36.10	27.86	25.76	57.19	42.08	38.29
Frustum PointNets	64-line LiDAR+camera	82.19	69.79	60.59	50.53	42.15	38.08	72.27	56.12	49.01
Proposed algorithm	Sparse point cloud+camera	87.98	77.14	73.33	45.97	38.94	35.81	68.12	55.25	53.55

上的平均检测精度，按照三个难易程度分别为87.98%，77.14%，73.33%，大幅超过基于单目相机的3D目标检测算法，并且也超过了经典的基于点云的3D目标检测算法（VoxelNet和SECOND）。经典的激光点云检测算法均使用KITTI数据集上64线激光雷达采集的点云。从表6可知，采用激光雷达作为输入的检测算法效果要大幅优于只采用相机的检测算法，主要是因为相机只能提供二维信息；神经网络使用图片进行3D目标检测时需要回归出物体的中心点坐标等三维信息，是从二维空间到三维空间的映射，因此较为困难；而激光雷达能够提供物体精确的三维坐标信息，从而准确地刻画物体的轮廓信息，使得神经网络的回归任务变为两个三维空间的映射，因此相较于相机效果要好。但是基于激光雷达点云的算法需要较稠密的点云，而稀疏点云难以刻画物体轮廓，增加了卷积核提取特征的难度。因此所提算法利用图片的丰富语义信息和纹理信息，结合稀疏点云提供的精确测距，首先利用深度补全网络输出稠密深度图；然后生成稠密点云并将其用于目标检测网络的输入，因此在检测效果上优于基于相机的算法。此外，当前低成本的低线数激光雷达的点云数量要高于所提算法使用的点云数量，而更多数量的点云将会提升深度补全网络的表现，进而提高整个网络的检测效果，因此在低线数激光雷达和相机条件下，所提算法的精度将得到更进一步提升。

3.4.2 算法运算时间复杂度分析与讨论

可以将模型输出每帧预测结果所消耗的时间作为衡量模型运算时间复杂度的指标，所消耗时间越少，表明运算时间复杂度越低。表7对比了所提算法和其他学者提出的3D目标检测算法的时间复杂度，所提算法输出每帧检测结果所需时间为0.08 s，运算复杂度与其他同类算法相比处于较低的水平。制约模型运算复杂度的主要因素在于模型所采用的卷积层个数及是否采用了双阶段的目标检测，所提算法比SECOND等算法时间复杂度高的原因在于所提算法需要先使用深度补全网络生成稠密点云，而SECOND和SS3D等算法没有此类步骤。此外，所提算法比PointRCNN等算法的时间复杂度低的主要原因在于PointRCNN等算法采用的是两阶段的目标检测，而所提卷积神经网络相比这些算法采用了较小的骨架网络，从而减少了卷积层的个数。目前主流激光雷达的采样频率为10 Hz，而所提算法的运算时间小于激光雷达的两帧采样数据的时间间隔，因此所提算法可以满足自动驾驶领域的实时性要求。

表 7 3D目标检测算法在KITTI测试集上的运行时间对比

Table 7 Running time comparison of 3D object detection algorithms on KITTI dataset

Parameter	VoxelNet	SECOND	PointRCNN	SS3D	D4LCN	Proposed algorithm
Running time /s	0.23	0.05	0.1	0.05	0.2	0.08

3.4.3 算法检测精度影响因素分析

3.3.1节实验目的为论证点云的数量对目标检测效果的影响。其实验条件与所提算法的不同点在于其关键点特征金字塔网络的输入为高线数激光雷达采集的稠密点云经降采样后的极端稀疏点云，而所提算法在实验时的关键点特征金字塔网络的输入为经深度补全后的稠密点云。采用极端稀疏点云的模型的检测精度大幅低于采用稠密点云的模型，原因在于极端稀疏点云中点的数量稀少，导致点云刻画物体外形轮廓的能力严重下降，增加了卷积神经网络的卷积核识别出物体外形轮廓的难度，进而增加了物体被检测出的难度。该实验证明了点云的数量对神经网络检测物体具有重要作用，低线数的激光雷达因点云的稀疏特性导致神经网络不能很好地检测出物体，而所提算法充分利用了相机提供的丰富语义信息和轮廓信息，结合已有的稀疏点云，使用深度补全网络生成稠密深度图并由此生成稠密点云，解决了点云稀疏性的难题。

3.3.2节实验目的为论证点云的组织形式对目标

检测效果的影响，其实验条件与所提算法的相同点在于都是先经过深度补全网络生成同样的稠密深度图并由此生成稠密点云，但不同点在于对点云的组织方式，前者按照前视图形式组织点云而后者将点云转成BEV图形式。实验结果表明，采用前视图组织点云的检测效果要低于采用BEV图的组织形式，因此表明点云不同的组织形式对目标检测网络的效果具有重要影响。BEV图的组织形式要优于前视图的组织形式，原因在于将点云组织成BEV图后，物体的轮廓在BEV图上呈现出明显的角点，角点特征降低了卷积神经网络识别物体的难度；除此之外，采用BEV图的组织方式可以解决前后物体的遮挡问题，而这是采用前视图的组织形式难以解决的。因此在前后物体相互遮挡的场景下，BEV图组织方式的检测效果要显著优于前视图的编码方式。

3.3.3节实验目的为论证稀疏点云对所提算法检测效果的影响，其实验条件与所提算法的不同点在于其深度补全网络的输入仅有图片，而所提算法的深度补全网络的输入为图片和稀疏点云。实验结果表明，仅使用图片作为深度补全网络输入时，算法的目标检测结果要低于所提算法，表明稀疏点云对算法的目标检测精度具有重要影响。其原因在于图片作为二维数据天然缺少第三个维度的信息，导致仅使用图片来预测深度时不仅难度大而且预测误差也较大，不精确的深度图导致点云三维坐标误差较大，最终影响到目标检测效果；而所提算法在深度补全时结合图片和稀疏点云作为输入，利用稀疏点云提供的关于深度方向上的测量值，使得深度补全网络输出的深度图更为精确，从而提高了目标检测的效果。

3.3.4节实验研究点云降采样率对所提算法检测效果的影响。实验结果表明，算法的检测结构与点云的降采样率呈正相关，原因在于稀疏点云的稀疏程度直接影响深度补全网络输出的深度图的准确率，进而影响由深度图生成的点云俯视图。在点云俯视图中，每个点的三维坐标误差决定了模型回归出的检测框的位置和航向角的误差，因此输入的稀疏点云越稠密，模型的检测效果越好。而且实验结果还表明，点云的降采样率对检测结果的提升呈边际递减的效应。

4 结　论

针对自动驾驶场景中3D目标检测任务中存在的高线数激光雷达成本高昂和基于毫米波雷达和相机的检测算法效果一般等问题，提出了基于深度补全网络和关键点检测的3D目标检测算法。该算法采用极端稀疏化后的点云和RGB图像作为输入，利用深度补全网络输出稠密深度图，由深度图生成点云俯视图，再将点云俯视图送入基于关键点特征金字塔的目标检测网络以输出检测结果。网络可以同时预测每个目标的3D边界框和边界框中心点坐标，以及每个目标的类别，例如车辆、行人和自行车等。在KITTI数据集上的实验表明：所提算法在检测精度方面大幅优于基于单目相机的3D目标检测算法，超过了部分经典的基于激光雷达的检测算法；可以应用在使用低成本的激光雷达和相机的场景中，并能取得与基于高线数激光雷达的检测算法相当的检测效果，从而可以降低自动驾驶感知系统的硬件成本。通过一系列对比实验对影响所提算法精度的相关因素进行了研究，实验结果和分析结果表明点云数量、点云组织形式和是否采用稀疏点云对算法的检测精度有重要影响。所提检测网络存在冗余部分，还可进一步进行优化以提升检测速度。在后续工作中，将继续修改网络结构，进一步提高检测精度。

参　考　文　献

[1] 张艳辉, 徐坤, 郑春花, 等. 智能电动汽车信息感知技术研究进展[J]. 仪器仪表学报, 2017, 38(4): 794-805.
Zhang Y H, Xu K, Zheng C H, et al. Advanced research on information perception technologies of intelligent electric vehicles[J]. Chinese Journal of Scientific Instrument, 2017, 38(4): 794-805.

[2] Li H, Fu K, Yan M L, et al. Vehicle detection in remote sensing images using denoizing-based convolutional neural networks[J]. Remote Sensing Letters, 2017, 8(3): 262-270.

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

[4] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[5] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 779-788.

[6] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 6517-6525.

[7] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL]. (2018-04-08)[2021-06-05]. https://arxiv.org/abs/1804.02767.

[8] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 936-944.

[9] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: optimal speed and accuracy of object detection [EB/OL]. (2020-04-23) [2021-02-05]. https://arxiv.org/abs/2004.10934.

[10] Wang C Y, Mark Liao H Y, Wu Y H, et al. CSPNet: a

new backbone that can enhance learning capability of CNN[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 14-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 1571-1580.

[11] Law H, Deng J. CornerNet: detecting objects as paired keypoints[J]. International Journal of Computer Vision, 2020, 128(3): 642-656.

[12] Law H, Teng Y, Russakovsky O, et al. CornerNet-lite: efficient keypoint based object detection[EB/OL]. (2019-04-18)[2021-05-06]. https://arxiv.org/abs/1904.08900.

[13] Zhou X Y, Wang D Q, Krähenbühl P. Objects as points [EB/OL]. (2019-04-16)[2021-05-04]. https://arxiv.org/abs/1904.07850.

[14] 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.

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

[16] 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, 2019: 770-779.

[17] 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)[2021-04-05]. https://arxiv.org/abs/1706.02413.

[18] 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.

[19] Simon M, Milz S, Amende K, et al. Complex-YOLO: an Euler-region-proposal for real-time 3D object detection on point clouds[M]//Leal-Taixé L, Roth S. Computer vision-ECCV 2018 workshops. Lecture notes in computer science. Cham: Springer, 2019, 11129: 197-209.

[20] Chen X Z, Ma H M, Wan J, et al. Multi-view 3D object detection network for autonomous driving[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 6526-6534.

[21] Li B, Zhang T L, Xia T. Vehicle detection from 3D lidar using fully convolutional network[EB/OL]. (2016-08-29) [2021-04-05]. https://arxiv.org/abs/1608.07916.

[22] 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.

[23] 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, 2018: 18392975.

[24] Vora S, Lang A H, Helou B, et al. PointPainting: sequential fusion for 3D object detection[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 4603-4611.

[25] van Gansbeke W, Neven D, de Brabandere B, et al. Sparse and noisy LiDAR completion with RGB guidance and uncertainty[C]//2019 16th International Conference on Machine Vision Applications (MVA), May 27-31, 2019, Tokyo, Japan. New York: IEEE Press, 2019: 18820916.

[26] Romera E, Álvarez J M, Bergasa L M, et al. ERFNet: efficient residual factorized ConvNet for real-time semantic segmentation[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(1): 263-272.

[27] Li P X, Zhao H C, Liu P F, et al. RTM3D: real-time monocular 3DDetection from object keypoints for autonomous driving[M]//Vedaldi A, Bischof H, Brox T, et al. Computer vision-ECCV 2020. Lecture notes in computer science. Cham: Springer, 2020, 12348: 644-660.

[28] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 770-778.

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

[30] Jörgensen E, Zach C, Kahl F. Monocular 3D object detection and box fitting trained end-to-end using intersection-over-union loss[EB/OL]. (2019-06-19)[2021-04-05]. https://arxiv.org/abs/1906.08070.

[31] Ding M Y, Huo Y Q, Yi H W, et al. Learning depth-guided convolutions for monocular 3D object detection [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 11669-11678.