计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20200595
58(12):2748-2759,2021

3D 物体检测的异构方法

吕 卓[1,2,3,4] 姚治成[1,2,5] 贾玉祥[4] 包云岗[1,2,5]

[1](中国科学院计算技术研究所 北京 100190)

[2](计算机体系结构国家重点实验室(中国科学院计算技术研究所) 北京 100190)

[3](数学工程与先进计算国家重点实验室 郑州 450001)

[4](郑州大学信息工程学院 郑州 450001)

[5](中国科学院大学 北京 100049)

(lvzhuo11@163.com)

A Heterogeneous Approach for 3D Object Detection

Lü Zhuo[1,2,3,4], Yao Zhicheng[1,2,5], Jia Yuxiang[4], and Bao Yungang[1,2,5]

[1](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[2](*State Key Laboratory of Computer Architecture* (*Institute of Computing Technology, Chinese Academy of Sciences*), *Beijing* 100190)

[3](*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou* 450001)

[4](*School of Information Engineering, Zhengzhou University, Zhengzhou* 450001)

[5](*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract 3D object detection is an important research direction of computer vision, and has a wide range of applications in areas such as autonomous driving. Existing cutting-edge works use end-to-end deep learning methods. Although it has achieved good detection results, it has problems such as high algorithm complexity, large calculation volume, and insufficient real-time performance. After analysis, we found that the deep learning method is not suitable for solving "partial tasks" in 3D object detection. For this reason, this paper proposes a 3D object detection scheme based on heterogeneous methods. This method uses both deep learning and traditional algorithms in the detection process, and divides the detection process into multi-task stages: 1) Use deep learning methods to obtain information such as the mask and object category of the detected object from the detected picture; 2) Based on the mask, use the fast clustering method to filter out the surface radar points of the target object from the radar point cloud space; 3) Use the information such as the object's mask, category and radar point cloud to calculate the object's orientation, border and other information to finally realize 3D object detection. We have implemented this method systematically, which we call HA3D (a heterogeneous approach for 3D object detection). Experiments show that on the 3D detection data set KITTI for cars, the method in this paper is within the acceptance range of

收稿日期:2020-08-05;**修回日期**:2021-04-25

基金项目:军科委基础加强项目(2019-xCxQ-xD-172-00);广东省普及型高性能计算机重点实验室项目(2017B030314073);国家自然科学基金项目(62090020, 61672499);中国科学院青年促进创新会项目(2013073);中国科学院战略性先导科技专项(XDC05030200)

This work was supported by the Foundation Enhancement Project of Commission of Science and Technology of the CMC (2019-xCxQ-xD-172-00), the Guangdong Province Key Laboratory of Popular High Performance Computers (2017B030314073), the National Natural Science Foundation of China (62090020, 61672499), the Youth Innovation Promotion Association of Chinese Academy of Sciences (2013073), and the Strategic Priority Research Program of Chinese Academy of Sciences (XDC05030200).

通信作者:姚治成(yaozhicheng@ict.ac.cn)

detection accuracy decline (2.0%) compared with the representative 3D object detection method based on deep learning, the speed is increased by 52.2%. The ratio of the accuracy to the calculation time has increased by 49%. From the perspective of comprehensive performance, this method has obvious advantages.

Key words　deep learning; autonomous driving; instance segmentation; clustering; 3D object detection

摘　要　3D物体检测是计算机视觉的一个重要研究方向,在自动驾驶等领域有着广泛的应用.现有的前沿工作采用端到端的深度学习方法,虽然达到了很好的检测效果但存在着算法复杂度高、计算量大、实时性不够等问题.经过分析发现3D物体检测中的"部分任务"并不适合使用深度学习的方法进行解决,为此提出了一种基于异构方法的3D物体检测方法,该方法在检测过程中同时使用深度学习和传统算法,将检测过程划分为多任务阶段:1)利用深度学习方法从被检测图片中获取被检测物体的mask、物体类别等信息;2)基于mask,利用快速聚类方法从雷达点云空间中筛选出目标物体的表面雷达点;3)利用物体mask、类别、雷达点云等信息计算物体朝向、边框等信息,最终实现3D物体检测.对该方法进行了系统实现,称之为HA3D(a heterogeneous approach for 3D object detection).经实验表明:在针对汽车的3D检测数据集KITTI上,该方法与代表性的基于深度学习的3D物体检测方法相比,在检测精度下降接受范围内(2.0%),速度提升了52.2%,精确率与计算时间的比值提升了49%.从综合表现上来看,方法具有明显的优势.

关键词　深度学习;自动驾驶;实例分割;聚类;3D物体检测

中图法分类号　TP391

3D物体检测是计算机视觉的一个重要研究方向,其主要任务是预测物体的尺寸、世界坐标系下的坐标以及朝向等信息,从而提供物体所处的3D空间.3D视觉识别对于机器人感知外界环境、理解周围场景和完成特定任务十分重要[1].3D物体检测在自动驾驶、机器人和目标追踪等场景中都有所应用.在自动驾驶领域,3D物体检测获取到的相关信息可以帮助汽车完成路径规划、避免碰撞等任务,自动驾驶需要3D物体检测来保证驾驶安全性,因此,如何更有效地得到精确的3D物体检测结果成为近些年来研究的热点.

当前的3D物体检测方法基本都难以同时满足高精度、快速度和低成本这3个要求[2].如图1所示,当前方法在速度-精度图中的分布基本都在曲线附近,精度较高的方法速度较慢,速度较快的方法精度较低,而理想的3D物体检测需要同时兼顾速度和精度.当前的3D物体检测方法大多以RGB图像、RGB-D数据、雷达点云等作为网络的输入,采用端到端的深度神经网络进行相关计算,最终输出预测的物体3D边框.然而直接使用端到端深度神经网络来解决3D物体检测这种复杂的任务,存在着网络结构复杂、计算量大、实时性差等问题.

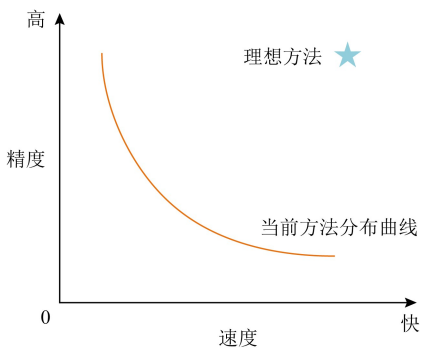

Fig. 1　3D object detection method status diagram

图1　3D物体检测方法现状示意图

本文提出了一种基于异构的3D物体检测方法,该方法以深度学习与传统算法相结合的形式,来进行3D物体检测.该方法的主要思路是将整个3D物体检测过程划分为多阶段:在预处理阶段,使用实例分割等深度学习的手段从RGB图片等原始数据中获取所需信息后,在后续的阶段中,使用聚类算法、图形处理算法等传统的方法来进行物体表面雷达点的获取、物体坐标及朝向的计算等.本文所提的方法适合用于检测汽车等可以在现实世界中获取到具体型号及对应尺寸的物体.

本文的主要贡献有4个方面：

1) 从全新的角度来考虑3D物体检测问题，将传统算法应用到检测过程中，与深度学习方法相结合，实现了一种采用异构形式进行3D物体检测的方法；

2) 提出的雷达点云筛选方法能够从巨大的雷达点云空间中，有效地筛选出目标物体的表面雷达点，并且去除其中存在的干扰点，在减少了雷达点计算量的同时，提升了计算精度；

3) 提出的“最小点边距外接矩形算法”，以及“物体所在高度计算方法”，在汽车坐标的计算中显著提升了计算速度和精度；

4) 经实验表明，本文方法与代表性的基于深度学习的3D物体检测方法相比，具有明显的优势.

1 相关工作

基于雷达点云的3D物体检测方法以激光雷达获取的点云数据作为输入，此外还有部分方法将RGB图像等数据作为额外的输入来帮助更好地进行检测，最终得到物体的3D边框，如图2所示：

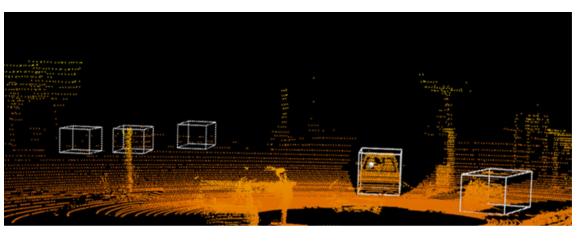

Fig. 2 3D bounding box of objects in radar point cloud

图2 雷达点云中的物体3D边框

在最初的阶段，由于卷积神经网络(convolutional neural network, CNN)需要规则的输入，因此基于雷达点云的3D物体检测通常先将不规则的雷达点云转化为规则数据格式，再输入网络进行相关检测. 例如Zhou等人[3]提出了一个端到端的将不规则点云转化为规则3D体素(voxel)，进而检测3D物体的网络，该网络由特征提取层、3D中间卷积层和区域生成网络[4](region proposal network, RPN)组成，其中的特征提取层先将点云转化为规则的3D体素，并对其中点数量较多的体素进行随机采样以减少计算量和体素间差异，然后再由3D中间卷积层进行特征提取，最后由RPN进行分类检测和位置回归，得到检测结果.

除了将不规则点云转化为体素这种形式外，还有一些方法将点云转化为多视角图像几何，例如Chen等人[5]提出了一种基于多视角的3D物体检测方法，该方法分别将雷达点云投射到鸟瞰图上，通过鸟瞰图获取候选框并将其分别投影到鸟瞰图、前视图和RGB图像这3个视角上，之后对各个视角的特征进行联合，进而预测目标类别并回归出3D边框；Ku等人[6]则以RGB图像和雷达点云数据投射生成的鸟瞰图作为输入，通过特征提取得到2个相应的特征图，经融合后使用RPN生成无方向区域建议，并使用子网络生成有方向的3D边框，完成3D检测.将雷达点投射到鸟瞰图的3D物体检测方法可以避免物体遮挡所带来的问题，并且投射到鸟瞰图的物体能够保留原始尺寸，但是投射的过程中不可避免地会损失一部分点云信息，且不适应垂直方向有多个物体的场景.

随着能够直接处理点云数据的深度网络[7-8]的出现，一些3D物体检测方法基于原始雷达点云数据进行检测.例如Charles等人[9]提出了一种基于2D对象检测器和3D深度技术的3D物体检测方法，该方法首先使用2D检测器构建对象建议并据此定义3D视锥区域，然后基于这些视锥区域中的3D点云，使用PointNet[7]/PointNet++[8]实现3D实例分割和非模态3D边界框估计；Shi等人[10]提出了第1个仅使用原始点云作为输入的多阶段3D对象检测器，该检测器也利用了PointNet++，直接从原始点云中生成3D方案，再根据语义信息和局部特征等进行优化.

在基于规则数据的3D物体检测方法中，将不规则点云转化成规则格式需要额外的计算工作，并且存在不可避免的信息损失；而直接基于原始雷达点云进行3D物体检测，则需要处理巨大的点云空间，对目标的分类也较为复杂；而且无论是基于规则数据转换还是基于原始雷达点云的方法，基本上都使用了结构较为复杂的深度神经网络，从而可能导致计算时间较长，时间成本较高.

2 基于异构的3D物体检测方法

现有的3D物体检测大多采用端到端的深度神经网络，使用这种方式来解决3D物体检测这样复杂的问题，无疑会增加深度神经网络的复杂度，进而导致计算量增大、实时性不够等问题，而且不是3D物体检测中的所有步骤都适合使用深度学习的方法，为此本文提出了一种基于异构的3D物体检测

方法,该方法的核心思想是将深度学习与传统算法相结合来进行检测,将整个检测流程划分为不同的子模块,分别承担不同的任务:在使用实例分割等深度学习手段从 RGB 图片等数据中获取信息后,根据深度学习获取的信息,采用传统算法,来进行点云筛选、坐标计算等任务.本文对该方法进行了实现,将其称为 HA3D(a heterogeneous approach for 3D object detection).

HA3D 由 5 个模块组成:数据预处理、雷达点云筛选、尺寸预测、坐标及朝向计算和结果展示,系统构成如图 3 所示.其中部分任务采用了非深度学习的方法来进行计算,例如雷达点云筛选使用了聚类的方法;坐标及朝向计算则使用了传统的计算机图形算法.以汽车的检测过程为例,HA3D 的整个方法流程如图 4 所示,主要划分为 4 个步骤:

1) 数据预处理.首先对原始数据进行处理,使用实例分割模型对 RGB 图像进行预测;读取雷达点云等原始数据并进行格式转换.

2) 尺寸预测.对汽车这类物体构建尺寸数据库,通过简单分类神经网络获取物体种类,根据种类查询数据库获取物体准确尺寸.

3) 雷达点云筛选.利用预处理得到的实例分割结果,结合聚类算法,从整个雷达点云空间中筛选出目标物体的表面雷达点,并去除其中干扰点,以用于下一步计算.

4) 坐标及朝向计算.根据前面基础模块获得到的雷达点云、物体尺寸等数据,采用图形以及点云处理算法,计算物体的坐标以及朝向,得到最终的检测结果.

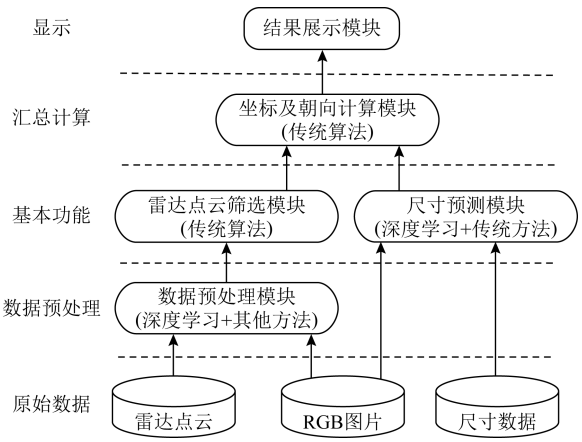

Fig. 3　System composition diagram of HA3D

图 3　HA3D 系统组成图

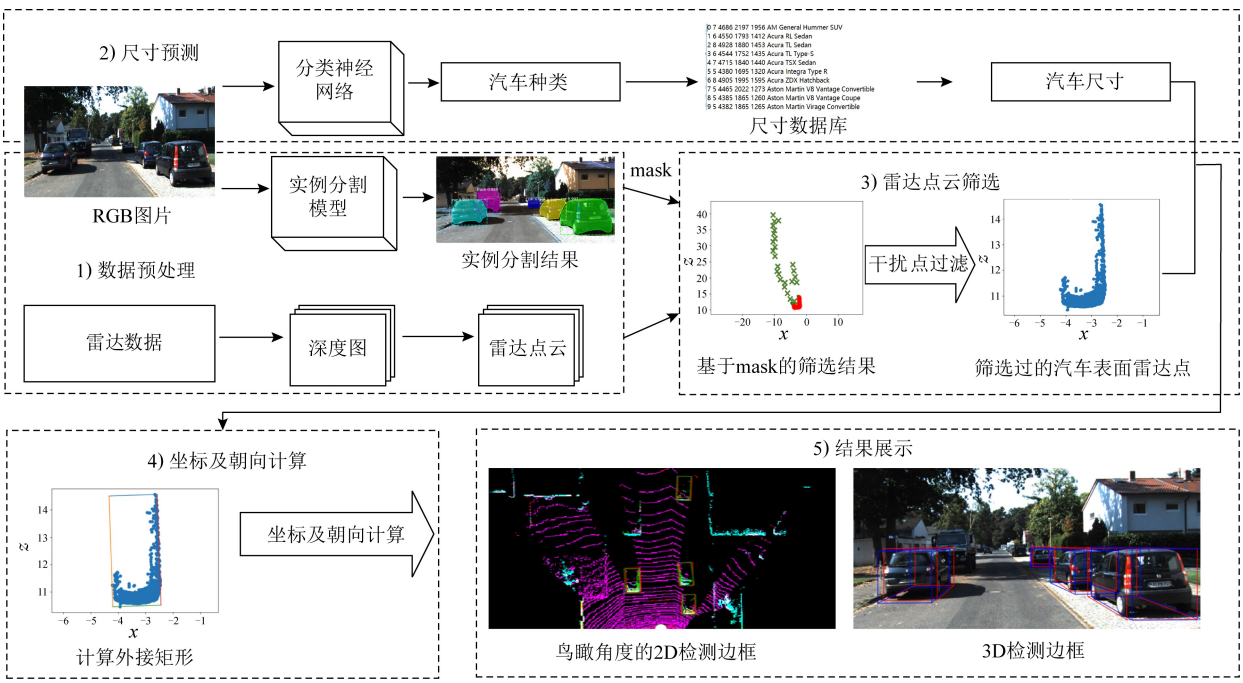

Fig. 4　Flow chart of heterogeneous 3D object detection

图 4　基于异构的 3D 物体检测流程图

2.1　数据预处理

数据预处理部分主要进行 2 部分工作:1)处理原始数据集,原始 3D 物体检测数据集中提供了 RGB 图像、相机参数、雷达数据和 3D 标注,在这里读取原始雷达数据,根据相机参数等对原始雷达数据进行格式转换、坐标系转换,并生成深度图和坐标

转换矩阵,为进一步的计算提供数据;2)获取图像分割结果,在这里使用实例分割模型对RGB图片进行预测,获取图片中物体的种类、2D检测边框、mask等数据,用于下一步的尺寸预测和雷达点云筛选,过程如图5所示.本文中使用的实例分割方法有Mask R-CNN[11]和YOLACT[12],据文献[12]中所述,前者检测结果的精确率较高,在COCO test-dev上的mask AP比后者高出5.9%,但是后者的检测速度比前者提高了3.9倍,可以帮助我们更快地获取相关信息,从而提升计算速度.

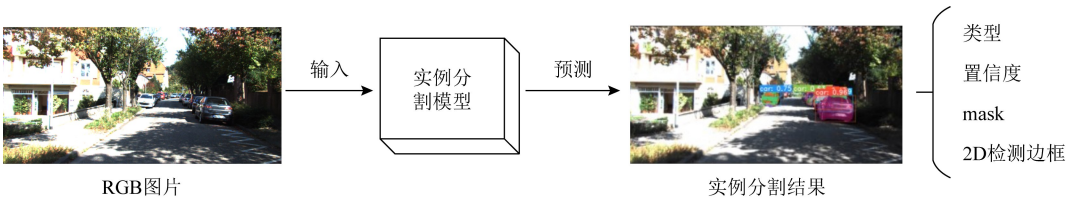

Fig. 5 Flow chart of instance segmentation

图5 实例分割流程图

2.2 尺寸预测

尺寸预测首先需要构建尺寸数据库,在现实世界中,汽车的型号及对应尺寸是可以获取的,例如,新浪汽车、edmunds等国内外各大汽车网站中都提供汽车的详细信息.基于此现实,我们可以搜集数据并构建汽车的数据库,其中存储汽车的具体车型类别和对应长宽高尺寸,并根据汽车尺寸划分小型车、中型车和大型车等尺寸类别.接下来将数据集中的汽车图片及对应具体车型类别作为训练数据,用于训练分类神经网络,可以基于ResNet[13],DenseNet[14]等结构较为简单的模型来构建分类网络,这样就能以较小的计算量预测汽车的具体车型类别.

构建完尺寸数据库并训练好分类神经网络后,就可以进行尺寸预测的工作,具体流程如图6所示,首先根据实例分割预测所得的2D检测边框,从RGB图像中裁剪出汽车的图片;然后对图片进行缩放等预处理操作后,输入分类神经网络进行种类预测;最后根据预测的具体类别查询尺寸数据库,获取汽车尺寸;当预测所得具体车型类别的置信度不高时,还可以根据该汽车具体车型所属的尺寸类别,查询出该尺寸类别汽车的平均尺寸,作为该汽车的尺寸.这种尺寸预测方法适合用于预测汽车等物体,在现实世界中,能够查询到这类物体的准确型号、对应尺寸和图片.

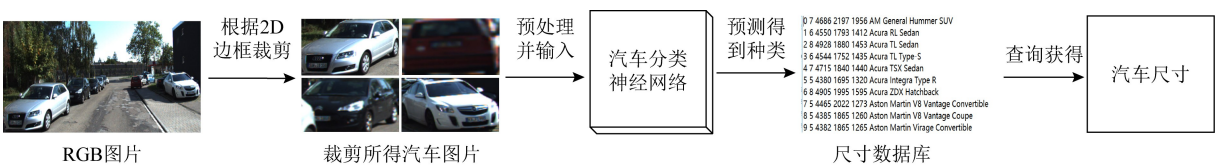

Fig. 6 Flow chart of size prediction

图6 尺寸预测流程图

2.3 雷达点云筛选

整个雷达点云空间的数据规模非常大,如果直接对其进行搜索和处理,需要花费大量的时间,从而降低了检测速度.因此,本文实现了基于mask的区域筛选和密度聚类筛选这2个雷达点云筛选方法,用于从雷达点云空间中快速找到较为纯净的目标物体表面雷达点,以获取汽车表面雷达点为例,流程如图7所示.

基于mask的区域筛选被用于从较大的点云空间中快速确定物体表面雷达点分布的空间范围.该方法首先要获得目标物体mask内雷达点的像素坐标,由于整张图片的尺寸较大,搜索整张图片会花费大量的时间,因此这里只遍历目标种类物体2D检测边框内的像素点,将位于物体mask范围内的雷达点保留下来;然后利用数据预处理得到的坐标转换矩阵,得到这些雷达点的相机坐标系坐标,即真实坐标,这样就初步获得了mask内的雷达点集。通过对图7中该筛选方法的结果进行观察,可以看到该方法能够快速锁定物体表面雷达点的大致分布范围,大大缩小了后续所需处理的雷达点数量.

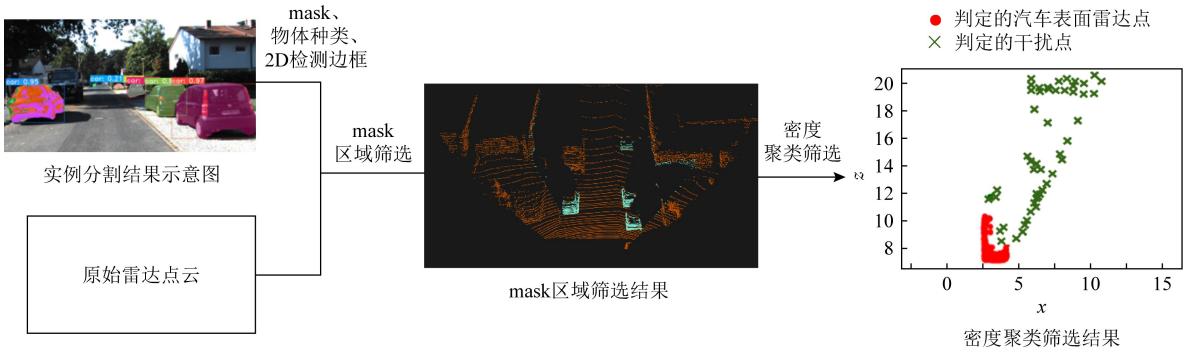

Fig. 7　Flow chart of radar point cloud screening

图 7　雷达点云筛选流程图

密度聚类筛选被用于进一步去除物体表面雷达点中的干扰点.由于物体遮挡、mask 存在误差等原因,经过初步筛选得到的雷达点中除了汽车表面雷达点外,还存在一部分干扰点.同一物体的表面雷达点分布较为集中,密度较高,因此使用 Scikit-learn[15] 所实现的密度聚类算法 DBSCAN[16] 来去除其中的干扰点.具体的聚类筛选操作为:首先,对获得的雷达点集进行 DBSCAN 聚类计算,得到 m 个聚类簇和 1 个离群异常簇 P_{outlier},将聚类簇的集合记为 $P=\{P_1,P_2,\cdots,P_m\}$,并记录下每个聚类簇中包含的点的数目 $N_{\text{pts}}=\{n_1=|P_1|,n_2=|P_2|,\cdots,n_m=|P_m|\}$;然后,从 m 个聚类簇中选出一个簇作为汽车表面雷达点集 P_{res},选取方法如式(1)所示:

$$P_{\text{res}}=\begin{cases}P_i(p_{\text{center}}\in P_i), & \text{若 } p_{\text{center}}\text{存在},\\ P_j(j=index(\max(N_{\text{pts}}))), & \text{其他},\end{cases}\tag{1}$$

其中 p_{center} 是位于汽车 mask 中心位置的雷达点.这样选取的原因是:位于汽车 mask 中心的点 p_{center} 大概率为汽车表面雷达点,因此包含 p_{center} 的聚类簇 P_i 大概率就是汽车表面雷达点集;如果表面雷达点过于稀疏导致 p_{center} 不存在,由于汽车 mask 范围内的主要物体就是汽车,汽车表面雷达点在整个点集中所占比例较高,所以此时选取 P 中点数量最多的簇 P_j,来作为最后筛选出的汽车表面雷达点集.

这里将部分筛选结果投射到 xOz 平面来进行观察,如图 8 所示,可以清晰地观察到:在 xOz 平面上,圆点分布明显符合汽车顶部的矩形形状,而叉点分布过于离散,明显不是汽车表面的雷达点.由此可见,基于 mask 的聚类筛选和密度聚类筛选 2 种筛选方法能较为有效地保留物体表面雷达点,去除其他干扰点,可以为进一步计算输入较为纯净的物体表面雷达点,从而达到减少计算量、提升计算精度的效果.

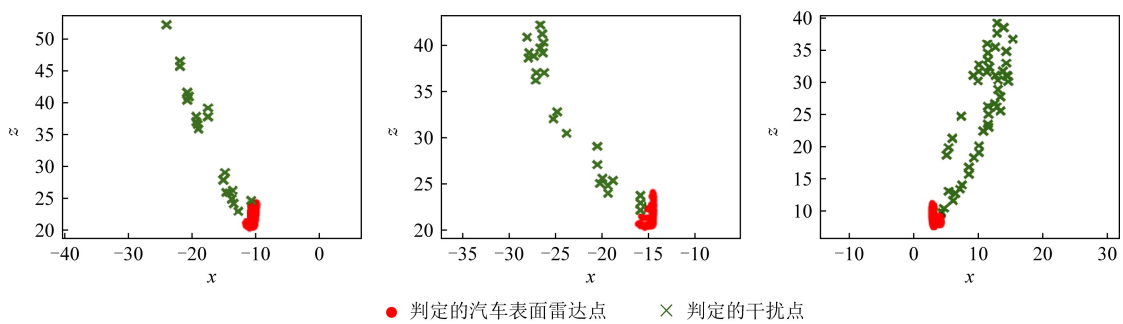

Fig. 8　Display of radar point cloud filtering results

图 8　雷达点云筛选结果展示

2.4　坐标及朝向计算

需要利用 2.3 节获取到的物体表面雷达点等数据,通过传统算法来替代深度神经网络,计算出物体的坐标以及朝向,其中包括 xOz 平面坐标的计算以及物体所在高度的计算.

2.4.1　xOz 平面坐标计算

以计算汽车的 xOz 坐标为例,整体流程如图 9 所示,接下来对该计算方法进行详细的描述:

1) 将雷达点投射到 xOz 平面并求得外接矩形

由于汽车从 xOz 平面观察呈现矩形形状,因此

将所获汽车表面雷达点投影到 xOz 平面后,可以求出这些雷达点的外接矩形,来代表汽车在 xOz 平面的分布情况,外接矩形需要尽量贴近雷达点的边缘点,以便更准确地描述汽车的朝向以及所在位置.

本文最初采用的是 OpenCV 实现的最小面积外接矩形,该方法最终得到的是各个角度的外接矩形中具有最小面积的外接矩形,因此在部分情况下,该方法所求得的外接矩形并不能贴合物体表面雷达点在 xOz 平面的边缘点,计算效果不理想.为了进一步提升该方法的效果,本文对原始雷达点云进行线性填充,使用稠密雷达点进行最小面积外接矩形计算,虽然提高了检测效果,但填充雷达点云和去除干扰点的过程带来了巨大的时间成本.因此本文希望通过原始雷达点云就可求得效果较好的外接矩形,于是对外接矩形算法进行改进.首先尝试了分段折线拟合法来求外接矩形,但该方法难以确定分段的转折点,不适用于本文方法.最终本文提出了最小点边距外接矩形,该算法求得的外接矩形能够较好地贴合边缘点,更能代表汽车在 xOz 平面的分布情况,效果最好.

"最小点边距外接矩形"中的"点"是指物体表面雷达点的凸外包点,"边"是指外接矩形的边.该方法的思路是:先求出物体表面雷达点的凸外包点集 $P_{\text{convex}}=\{p_1,p_2,\cdots,p_n\}$,紧接着将初始外接矩形 $rect$ 旋转不同角度 $\theta(0°\leqslant\theta\leqslant180°)$ 得到 $rect_\theta$,将点 $p_i(1\leqslant i\leqslant n)$ 到矩形 $rect_\theta$ 边框的欧氏距离记为 $dis(p_i,rect_\theta)$,找到其中的最佳旋转角度 θ_{best},该角度满足的条件为

$$\theta_{\text{best}}=\arg\min_{\theta}\sum_{i=1}^{n}dis(p_i,rect_\theta),\quad p_i\in P_{\text{convex}},\ 0°\leqslant\theta\leqslant180°,\tag{2}$$

旋转角度为 θ_{best} 的外接矩形就是我们所需要的最小点边距外接矩形,由于该方法的原理可以计算得到较为贴近凸外包点的外接矩形,并且只对凸外包点进行相关计算,时间开销较低,满足本文方法对外接矩形的要求,综合效果最佳.

2) 补全外接矩形

以汽车为例,由于受到雷达探测器所处位置和物体遮挡等因素的影响,雷达一般只能检测到汽车部分表面,导致获取到的物体表面雷达点不完整,进而导致求得的外接矩形也不"完整".也就是说,所求得的外接矩形的长和宽可能与汽车真实的长和宽存在一定的差距,所以还需要使外接矩形的长和宽分别等于汽车的真实长和宽,即补全外接矩形,如图 9 所示:

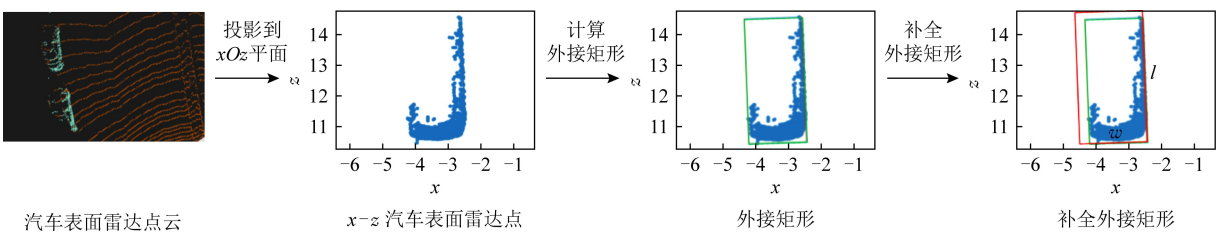

Fig. 9 Flow chart of calculation method of xOz plane coordinates

图 9 xOz 平面坐标计算流程图

按照图 9 中方法补全的外接矩形,可以更好地描述汽车在 xOz 平面的分布情况.补全的外接矩形的中心坐标就是汽车在 xOz 平面的坐标,补全的外接矩形的长边与相机坐标系 x 轴的夹角即汽车的朝向.

2.4.2 物体所在高度计算

需要计算物体底面所在高度,即物体在相机坐标系下的 y 轴坐标.为了提升计算速度,本文提出了一种新的物体所在高度的计算方法,该方法的思路就是将计算物体底面的 y 轴坐标,转化为计算物体 2D 边框底边的 y 轴坐标.以汽车底部所在高度的计算为例,计算示意图如图 10 所示.在数据预处理中已经获得了汽车的 mask,2D 检测边框及其像素坐标($x_{\min},y_{\min},x_{\max},y_{\max}$),在尺寸预测中获取到了汽车尺寸($l,w,h$),并且已经通过前面的雷达点云筛选方法获得了汽车表面雷达点;从 mask 中心区域随机选取 1 个汽车表面雷达点 p,其相机坐标系坐标(真实坐标)是已知的,记为(x_c,y_c,z_c)(此处下标 c 代表 camera),该点在单目 RGB 图像中对应的像素坐标(x_p,y_p)(此处下标 p 代表 pixel)也是已知的.

点 p 到 2D 边界框底边的像素距离占边界框像素高度的比例,等于点 p 到 2D 边界框底边的真实距离占汽车高度 h 的比例,据此可以计算出边界框底边的相机坐标系 y 轴坐标,即汽车的所在高度 y_{car}:

$$y_{\text{car}}=y_c+\frac{y_{\max}-y_p}{y_{\max}-y_{\min}}\times h.\tag{3}$$

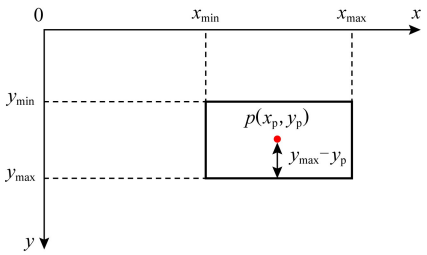

Fig. 10　Schematic diagram of calculating the height of the car

图 10　汽车所在高度计算示意图

该方法能够较快地求得物体所在高度,但是较为依赖实例分割模型预测所得物体 2D 检测边框的准确性,实例分割模型预测的 2D 边框越准确,该方法所求的物体所在高度也越准确.

3　实验与结果

3.1　实验环境和相关设置

本文实验全部都在服务器上进行,服务器系统版本为 CentOS Linux release 8.1.1911(Core),使用的 CPU 为 Intel® Xeon® CPU E5-2650 v4@2.20 GHz.实验所使用的主要依赖库及版本信息如表 1 所示:

Table 1　Dependency Library and Version Information Used in the Experiment

表 1　实验所用主要依赖库及版本信息

依赖库	版本号
tensorflow	1.12.0
keras	2.2.4
numpy	1.18.1
cython	0.29.16
scipy	1.1.0
torch	1.4.0+cpu
torchvison	0.5.0+cpu
pillow	7.1.1
scikit-learn	0.22.2.post1
opencv-python	4.2.0.32

实验以 KITTI[17] 作为 3D 物体检测数据集.由于 KITTI 数据集中未标出汽车种类,所以汽车分类神经网络缺少训练数据,暂时无法获得效果较好的汽车分类神经网络,因此无法通过网络预测车型,进而查询获得尺寸.所以在这里,我们假设能够通过本文所提出的尺寸预测方法获取汽车的尺寸,在实验中暂时使用 KITTI 中人工标注的汽车尺寸作为替代.

为了证明本文提出的尺寸预测方法的可行性,我们查询了相关资料,发现当前车型识别的相关研究工作已经较为成熟,根据车型识别精度的排行榜[18],其中提及的 29 个车型识别方法在 Car Dataset[19] 上的车型识别的精度均在 90%以上,并且最高精度已经达到了 96.2%,所以如果下一步能够获得较为充足的训练数据,我们应该可以训练得到较为准确的车型识别的分类神经网络,实现本文所提出的尺寸预测思路.

3.2　评价指标

HA3D 采用的是异构的方法,最终的检测结果没有置信度输出,而传统的 3D 物体检测评估方法会将置信度也作为评估因素,不适合用于评估本文提出的方法,所以本文对传统 3D 物体检测评估方法进行改造,形成新的评估方法.

本文使用交并比(intersection over union, IoU)来描述检测结果和标注数据中边框的重叠率,使用 IoU 阈值来描述当检测结果中的 3D 边框与标注中边框的重叠率大于何值时,方为合格的检测结果.此外还根据标注中物体的遮挡程度、截断指数、2D 边界框高度、物体到相机的距离等设置评估范围,在评估范围内的标注数据和检测结果视为有效的数据,进行下一步的评估,不满足该条件的标注数据和检测结果不予评估.

参考 KITTI[17] 的 3D 物体检测评估程序,本文使用 2 个概念来描述检测结果:真正例(true positive, TP),以及假正例(false positive, FP).TP 表示与有效标注数据的 IoU 大于检测阈值的有效检测结果数量;FP 表示与任意有效标注数据的 IoU 都不大于检测阈值的有效检测结果数量.由于 KITTI 标注中存在'DontCare'标注,这类标注数据未对物体的 3D 空间进行标注,因此不应该将该类区域的检测结果纳入评估范围,在这里使用 $STUFF$ 来表示与'DontCare'标注的 2D 检测边框的重叠率大于检测阈值的有效检测结果数量.

明确定义之后,使用精确率($Precision$)对检测结果进行评估,精确率也叫查准率,它用来描述检测出的 TP 占检测出的所有正例(TP 和 FP)的比例,计算为

$$Precision=\frac{TP}{TP+FP}. \tag{4}$$

但是为了避免将对应'DontCare'标注的检测结果的数量计入 FP,需要将 $STUFF$ 从检测出的 FP 中去除,重新定义评估方法中所使用的精确率:

$$Precision=\frac{TP}{TP+(FP-STUFF)}. \tag{5}$$

为了评估方法的检测速度,本文使用平均每张图片所需的计算总时间以及每秒处理图片数目(frames per second, fps)作为检测速度的评价指标,同时为了更准确地评估3D检测的效率,将总时间细分为平均每张图片所需的数据预处理时间以及3D检测时间.为了同时评估方法的精度和速度,将平均每张图片所需的计算时间记为 t,使用精确率与计算时间的比值(ratios of precision and time, RPT),作为评价方法综合表现的指标,比值越大,说明检测效果越好,其计算方法为

$$RPT=\frac{Precision}{t}. \tag{6}$$

3.3 实验结果和分析

3.3.1 不同外接矩形算法

在相同的CPU实验环境下,从KITTI[17]训练集中随机选取1 000张图片,分别使用OpenCV的最小面积外接矩形、稠密点云下的最小面积外接矩形、分段折线拟合法和最小点边距外接矩形这4种算法对其中的汽车进行检测,并使用相同的评估范围,检测阈值设置为0.7,评估不同方法的精确率和速度的综合表现,评估结果如图11所示.

从图11中可以发现:虽然稠密雷达点云下的最小面积外接矩形算法,相对于稀疏雷达点云下的最小面积外接矩形算法,精确率高出5.81%,但检测速度极慢;而分段折线拟合法显然不适用于本文方法,检测结果的精确率仅仅只有4.88%;最小点边距外接矩形算法的检测结果的精确率远高于其他方法,并且其检测速度也非常快.因此,本文所提出的最小点边距外接矩形算法所求矩形更为贴合边缘点,计算的速度较快、精度最高,在HA3D中应用效果最佳.

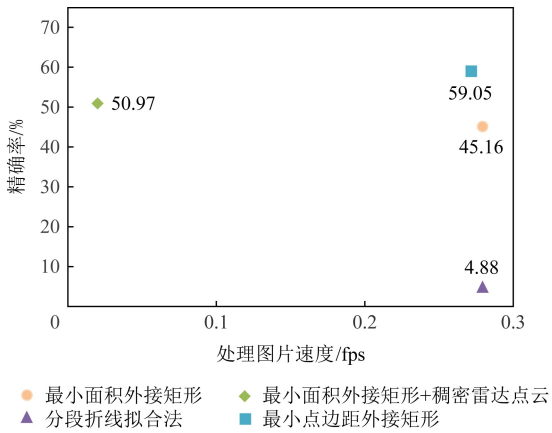

Fig. 11 FPS-Precision scatterplot of different bounding rectangle algorithms

图11 不同外接矩形算法FPS-Precision散点图

3.3.2 与其他3D物体检测方法对比

本节将对比HA3D和部分具有代表性的3D物体检测方法.在相同的CPU环境下,我们对所有方法的检测结果使用相同的评估范围,测试每张图片所需的预处理时间、3D检测时间以及总的计算时间,同时对比检测阈值分别为0.7和0.5时的精确率,以及检测阈值为0.5时的 RPT 综合指标,评估结果如表2所示:

Table 2 Comparison with Other 3D Object Detection Methods

表2 与其他3D物体检测方法对比

方法	平均每张图片所需时间/s			精确率/%		RPT (IoU阈值=0.5)
	数据预处理时间	3D检测时间	合计时间	IoU阈值=0.7	IoU阈值=0.5	
VoxelNet[3]	0.24	5.01	5.25	36.24	56.53	10.77
AVOD[6]	**0.14**	5.46	5.60	48.73	**83.11**	14.84
F-PointNet[9]		**0.49**		56.59	70.02	
F-ConvNet[20]				**63.78**	77.63	
Mask R-CNN[11]+HA3D(本文方法)	15.68	2.21	17.89	59.05	80.10	4.48
YOLACT[12]+HA3D(本文方法)	2.90	0.78	**3.68**	59.05	81.41	**22.12**

注:黑体表示在该项指标下表现最好的结果.

VoxelNet的代码与预处理模型来源为github项目[21];AVOD代码来源为官方实现,预训练模型来自Pseudo-LiDAR[22];F-PointNet和F-ConvNet的代码及预训练模型为官方实现.在这里还绘制了散点图来对各个方法的综合表现进行直观的对比(其中处理图片速度是根据各个方法3D检测部分

的平均时间进行计算,*Precision* 是 IoU 为 0.5 时的精确率),如图 12 所示:

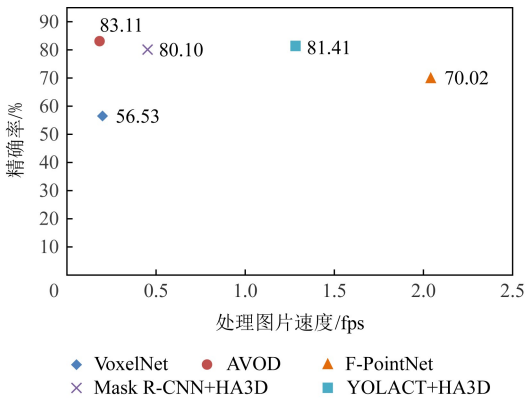

Fig. 12　FPS-Precision scatterplot of different 3D object detection methods

图 12　不同 3D 物体检测方法 FPS-Precision 散点图

表 2 中的 F-PointNet 和 F-ConvNet 均未给出完整的 3D 检测代码,两者均直接使用了提前生成的 2D 检测边框和 3D 视锥体点云,导致本文无法对其进行完全复现,进而无法测试其完整的计算时间.此外,F-PointNet 检测流程中的视锥体提案生成部分采用了基于 FPN[23] 的神经网络,传统的 FPN 网络模型通常比较复杂,GPU 推理时间在 100 ms 以上,而本文中用到的 YOLACT[12] 网络 GPU 推理时间约 30 ms,比传统 FPN 网络快 3 倍以上,类比到 CPU 设备,结合表 2 中的数据,可推断出 F-PointNet 的数据预处理时间大于 8.7 s,由此可以推测其检测总用时约 9.19 s,是本文工作 HA3D 检测总时长的 2.5 倍,因此 F-PointNet 的检测时间并不占优势.而且 F-PointNet 在 3D 物体检测部分使用的是轻量级的 PointNet[7],虽然减少了少量的 3D 检测时间,但造成了其精确率的下降,相对于 HA3D,其精确率下降了 11.39%,因此 HA3D 的综合表现更优.因此,本方法相对其他基于深度学习的方法(例如 F-PointNet)无论是在总体检测速度,还是检测精度上都有明显优势.

本文通过对比各个方法的检测时间来衡量各个方法的复杂度,从表 2 中可以观察到 HA3D 的 3D 物体检测时间远远少于 VoxelNet 和 AVOD,相对于 AVOD 的检测时效提升了 600%,且通过分析,可以得知 F-PointNet 的检测时间并不占优势.因此,本文利用传统图形算法来进行 3D 边框的计算,相对于部分利用深度神经网络进行 3D 边框回归的检测方法,所需要的检测时间更少.这说明了 HA3D 方法进行 3D 物体检测的算法复杂度更低、计算规模更小,因此进行检测的速度更快.

从表 2 和图 12 中综合比较各个方法的检测速度和精确率,其中 AVOD 方法的精确率比 HA3D 方法高 1.7%,但它采用了较为复杂的网络结构,导致计算量较大,总体运行时间约为 HA3D 方法的 1.5 倍;F-PointNet 处理图片的速度虽高于 HA3D 方法,但从之前的讨论中可以得知这是因为它采用了轻量级的网络,也因此导致了精确率的大幅度下降.从图 12 中也可以看到本文提出的 YOLACT+HA3D 方法位于右上角,这说明 HA3D 方法的综合表现最好.经结果对比分析可得出:本文提出的 HA3D 方法在检测精度下降可接受范围内,大幅度地提高了总体检测速度,相比于传统的基于端到端的深度学习方法,更适合于自动驾驶等实时性要求高的工作场景.

因此,在汽车的 3D 检测中,相对于部分具有代表性的端到端的 3D 物体检测方法,HA3D 在精确率和速度的综合方面上表现突出,这说明了传统方法确实适合用来完成 3D 物体检测过程中的部分任务,使用聚类算法等来进行雷达点云筛选和坐标计算,取得了很好的检测精度,并且减少了计算量,提升了检测速度.

4　总结与展望

本论文提出的 HA3D 方法从全新的异构角度来考虑 3D 物体检测问题,将传统算法应用到 3D 物体检测中,实现了"深度学习+传统算法"的异构 3D 物体检测方法.从实验结果中来看,本文方法相对于部分基于端到端深度学习的 3D 物体检测方法,在速度和精确率的综合表现上具有优势(速度提升 52.2%,*RPT* 指标提升 49%),该结果说明传统算法确实适合用于解决 3D 物体检测中的坐标计算等问题.这样将深度学习和传统算法相结合进行 3D 物体检测的方法,相对于直接使用端到端的深度神经网络,不仅保证了较高的准确率,还具有更快的检测速度,并且该方法通过多模块的形式来实现,具有更加灵活的结构和更好的可拓展性.

本文提出的方法还有进一步的优化空间,例如:1)本文方法还可以通过算法优化和 GPU 加速,进一步提升计算速度;2)可以尝试通过路面检测来获取物体所在高度,从而避免 2D 检测边框准确性对结果的影响,可能会进一步提高 3D 检测结果的精确率;

3)如果能够获得充足的汽车分类神经网络的训练数据,就可以训练出汽车分类神经网络,进而预测获得汽车尺寸数据,实现本文所提出的汽车尺寸预测思路.

参 考 文 献

[1] Jiang Shuqiang, Min Weiqing, Wang Shuhui. Survey and prospect of intelligent interaction-oriented image recognition techniques [J]. Journal of Computer Research and Development, 2016, 53(1): 113-122 (in Chinese)
(蒋树强, 闵巍庆, 王树徽. 面向智能交互的图像识别技术综述与展望[J]. 计算机研究与发展, 2016, 53(1): 113-122)

[2] Wang Yongsen. Study progress of advances in 3D object detection technology [C] //Proc of the 23rd Annual Network New Technology and Application Conf of China Computer Users Association Network Application Branch in 2019. Beijing: Beijing Union University Beijing Key Laboratory of Information Service Engineering, 2019: 177-182 (in Chinese)
(王永森. 3D目标检测技术的研究进展[C] //中国计算机用户协会网络应用分会2019年第二十三届网络新技术与应用年会论文集. 北京: 北京联合大学北京市信息服务工程重点实验室, 2019: 177-182)

[3] Zhou Yin, Tuzel O. Voxelnet: End-to-end learning for point cloud based 3D object detection [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 4490-4499

[4] Ren Shaoqing, He Kaiming, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [C] //Proc of the Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2015: 91-99

[5] Chen Xiaozhi, Ma Huimin, Wan Ji, et al. Multi-view 3D object detection network for autonomous driving [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 1907-1915

[6] Ku J, Mozifian M, Lee J, et al. Joint 3D proposal generation and object detection from view aggregation[C/OL] //Proc of the Int Conf on Intelligent Robots and Systems. Piscataway, NJ: IEEE, 2018[2020-06-01]. https://arxiv.org/pdf/1712.02294v3.pdf

[7] Charles R Q, Su Hao, Mo Kaichun, et al. PointNet: Deep learning on point sets for 3D classification and segmentation [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 652-660

[8] Charles R Q, Yi Li, Su Hao, et al. PointNet++: Deep hierarchical feature learning on point sets in a metric space [C] //Proc of the Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2017: 5099-5108

[9] Charles R Q, Liu Wei, Wu Chenxia, et al. Frustum PointNets for 3D object detection from RGB-D data [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 918-927

[10] Shi Shaoshuai, Wang Xiaogang, Li Hongsheng. PointRCNN: 3D object proposal generation and detection from point cloud [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 770-779

[11] He Kaiming, Gkioxari G, Dollár P, et al. Mask R-CNN [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 2961-2969

[12] Bolya D, Zhou Chong, Xiao Fanyi, et al. YOLACT: Real-time instance segmentation [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 9157-9166

[13] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770-778

[14] Huang Gao, Liu Zhuang, Maaten L, et al. Densely connected convolutional networks [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 4700-4708

[15] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python [J]. Journal of Machine Learning Research, 2011, 12: 2825-2830

[16] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C] //Proc of the 2nd Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 1996: 226-231

[17] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2012: 3354-3361

[18] Facebook AI Research. Fine-grained image classification on stanford cars [EB/OL]. 2020 [2020-06-01]. https://paperswithcode.com/sota/fine-grained-image-classification-on-stanford

[19] Krause J, Stark M, Deng J, et al. 3D object representations for fine-grained categorization [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2013: 554-561

[20] Wang Zhixin, Jia Kui. Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection [C] //Proc of the Int Conf on Intelligent Robots and Systems. Piscataway, NJ: IEEE, 2019: 1742-1749

[21] Qianguih. Voxel [CP/OL]. (2018-04-15) [2020-06-01]. https://github.com/qianguih/voxelnet

[22] Wang Yan, Chao Weilun, Garg D, et al. Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 8445-8453

[23] Lin Tsungyi, Dollar P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 2117-2125

Lü Zhuo, born in 1998. Master candidate. Her main research interests include computer vision and deep learning.

吕　卓,1998 年生.硕士研究生.主要研究方向为计算机视觉和深度学习.

Yao Zhicheng, born in 1989. PhD candidate, engineer. Member of CCF. His main research interests include computer vision, deep learning and computer system.

姚治成,1989 年生.博士研究生,工程师,CCF 会员.主要研究方向为计算机视觉、深度学习和计算机系统.

Jia Yuxiang, born in 1981. PhD, lecturer, master supervisor. Member of CCF. His main research interests include natural language processing, machine learning and data mining.

贾玉祥,1981 年生.博士,讲师,硕士生导师,CCF 会员.主要研究方向为自然语言处理、机器学习、数据挖掘.

Bao Yungang, born in 1980. PhD, professor and PhD supervisor. Member of CCF, ACM, IEEE. His main research interests include computer architecture, operating system, system performance modeling and evaluation.

包云岗,1980 年生.博士,研究员,博士生导师,CCF,ACM,IEEE 会员.主要研究方向为计算机体系结构、操作系统、系统性能建模与评估.