

电子科技大学  
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

# 硕士学位论文

MASTER THESIS



论文题目 基于 RGB 和 LIDAR 数据的 3D 目标检测

算法研究

学科专业 信号与信息处理  
学号 201821011618  
作者姓名 何庆东  
指导教师 王正宁 副教授

分类号 \_\_\_\_\_ 密级 \_\_\_\_\_

UDC <sup>注1</sup> \_\_\_\_\_

# 学 位 论 文

## 基于 RGB 和 LIDAR 数据的 3D 目标检测算法研究

(题名和副题名)

何庆东

(作者姓名)

指导教师

王正宁

副教授

电子科技大学

成 都

(姓名、职称、单位名称)

申请学位级别 硕士 学科专业 信号与信息处理

提交论文日期 2021.03.25 论文答辩日期 2021.05.25

学位授予单位和日期 电子科技大学 2021 年 06 月

答辩委员会主席 \_\_\_\_\_

评阅人 \_\_\_\_\_

注 1：注明《国际十进分类法 UDC》的类号。

# **Research on 3D Object Detection Based on RGB and LIDAR Data**

**A Master Thesis Submitted to  
University of Electronic Science and Technology of China**

**Discipline:** Signal and Information Processing

**Author:** Qingdong He

**Supervisor:** Prof. Zhengning Wang

**School:** School of Information and  
Communication Engineering

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名： 何庆东 日期： 2021 年 5 月 30 日

## 论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构递交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后应遵守此规定)

作者签名： 何庆东 导师签名： 王玲

日期： 2021 年 5 月 30 日

## 摘要

近年来，自动驾驶有了广泛的应用，多个传感器会被安装在自动驾驶汽车中，如激光雷达（LIDAR）和视觉传感器等。激光雷达能够捕获精确的深度信息，而详细的语义信息却保留在视觉传感器捕获的数据中。在自动驾驶中，一项比较关键的任务是对目标进行基于深度学习的3D检测与定位。深度学习在目标检测和实例分割等2D计算机视觉任务上已取得了显着进步。但是基于点云数据的3D目标检测仍然面临着巨大挑战，如数据格式不规则、搜索空间的自由度较大、LIDAR点云具有的无序性和不规则性等特点在处理上较为复杂，而RGB图像由于缺少了空间信息而很难达到较高的检测准确度等等。针对上述问题，本文在充分研究了现有3D目标检测算法的基础之上，结合LIDAR点云数据和RGB图像数据的特点，提出了系列的改进方案，并通过实验充分证明所设计的网络结构的可行性。本文主要研究内容及贡献包括：

（1）首先对使用不同数据进行三维目标检测的不同算法进行系统全面的研究，对单独使用RGB图像数据、单独使用LIDAR点云及将RGB图像数据与LIDAR点云进行结合这三类算法进行了详细划分，对每一类算法的算法结构、网络框架等进行了充分理解与研究。

（2）提出了一种使用双目RGB图像数据和激光雷达点云数据融合的三维目标检测网络。针对单目RGB图像对目标定位不够准备的问题，提出了新的双目RGB图像融合方案，且通过加入边缘卷积与残差注意力模块的方式生成更加紧凑的点云分割方案，以将目标点更准备的与背景分割开来，并提出一种新的3D框编码方案进一步提高检测精度，最后通过实验证明了所设计的网络的有效性。

（3）提出了一种基于体素图表示方法的三维目标检测网络。该网络以原始点云作为输入并输出目标的类别和边界框信息，并主要由体素图网络模块和稀疏到密度回归模块组成。体素图网络旨在为每个体素构造局部完整图，为所有体素构造全局KNN图。局部完整图和全局KNN图充当注意机制，可以为每个点的特征向量提供参数监视因子。这样，可以将局部特征与全局特征逐点组合。所设计的稀疏到密集的回归模块，通过融合处理不同尺度的特征图来预测类别和3D边界框。最终的实验结果验证了基于图表示方法的三维目标检测的有效性。

**关键词：**自动驾驶，3D目标检测，双目RGB图像，点云分割，多尺度

## ABSTRACT

In recent years, autonomous driving has developed by leaps and bounds. Modern autonomous vehicles are also equipped with multiple sensors, such as lidar and vision sensors. The laser scanner has the advantage of accurate depth information, while the camera retains more detailed semantic information. In autonomous driving, a more critical task is to perform 3D detection based on deep learning. Deep learning has made significant progress in 2D computer vision tasks such as object detection and instance segmentation. However, there are great challenges in 3D object detection with point clouds, such as irregular data format, greater freedom of search space, disorder and irregularity of LIDAR point clouds, etc., and the processing is more complicated, while RGB images lack spatial information, it is difficult to achieve high detection accuracy and so on. In response to the above problems, this article has fully studied the existing 3D object detection algorithms, combined with the characteristics of LIDAR point cloud data and RGB image data, and proposed a series of improvement algorithms, and fully proved the feasibility of the designed network structure through experiments. The main research contents and contributions of this thesis are as follows:

(1) First of all, a comprehensive introduction to the current mainstream 3D object detection algorithms is carried out, and the three types of algorithms are described in detail for the use of RGB image data alone, the use of raw point clouds alone, and the combination of the two different data. The algorithm structure and network framework of each type of algorithm have been fully studied, and the processing methods for RGB images and LIDAR point cloud data have been deeply understood and researched.

(2) A 3D object detection network fused with stereo RGB image data and LIDAR point cloud data is proposed. Aiming at the problem that monocular RGB images are not sufficiently prepared for object positioning, a new stereo RGB image fusion scheme is proposed, and a more compact point cloud segmentation scheme is generated by adding edge convolution and residual attention modules to separate the object point from the background more accurately. Then, a new 3D coding scheme is proposed to further improve the detection accuracy. Finally, experiments on KITTI have proven the effectiveness of the designed network.

---

## ABSTRACT

---

(3) An end-to-end 3D object detection network represented by graph pair irregular point cloud is proposed. The network takes the raw point cloud as input and outputs the object category and bounding box information, and is mainly composed of voxel graph network module and sparse-to-density regression module. The voxel graph network aims to construct a local complete graph for each voxel and a global KNN graph for all voxels. The local complete graph and the global KNN graph act as an attention mechanism and can provide parameter monitoring factors for the feature vector of each point. In this way, local aggregated features can be combined with global point-by-point features. The designed sparse-to-dense regression module predicts categories and 3D bounding boxes by fusing feature maps of different scales. The final experimental results demonstrate the effectiveness of 3D object detection with graphs for irregular data representation of point clouds.

**Keywords:** autonomous driving, 3D object detection, stereo RGB image, point cloud segmentation, multi-scale

## 目 录

<b>第一章 绪论 .....</b>	<b>1</b>
1.1 研究背景及意义 .....	1
1.2 国内外研究现状 .....	3
1.2.1 RGB 图像数据处理 .....	6
1.2.2 LIDAR 点云数据处理 .....	7
1.3 论文的主要研究内容 .....	8
1.4 论文的结构安排 .....	9
<b>第二章 基于深度学习的 3D 目标检测算法概述 .....</b>	<b>11</b>
2.1 基于 RGB 图像数据的 3D 目标检测算法 .....	11
2.1.1 基于单目 RGB 图像数据的 3D 目标检测算法 .....	12
2.1.2 基于双目 RGB 数据的 3D 目标检测算法 .....	15
2.2 基于 LIDAR 点云数据的 3D 目标检测算法 .....	17
2.2.1 体素化 3D 目标检测 .....	17
2.2.2 基于 PointNet 的 3D 目标检测 .....	19
2.2.3 体素化与 PointNet 融合的 3D 目标检测 .....	22
2.2.4 基于图卷积网络的 3D 目标检测 .....	25
2.3 基于不同数据融合的 3D 目标检测算法 .....	27
2.3.1 LIDAR 点云与高精度地图融合 .....	28
2.3.2 LIDAR 点云与 RGB 图像融合 .....	29
2.4 本章小结 .....	31
<b>第三章 基于双目 RGB 和 LIDAR 点云融合的 3D 目标检测方法 .....</b>	<b>32</b>
3.1 双目 2D 建议框融合 .....	33
3.2 3D 点云分割 .....	33
3.2.1 分割网络概述 .....	34
3.2.2 基于注意力机制的特征融合 .....	35
3.2.3 注意力模块 .....	37
3.3 3D 检测框回归 .....	38
3.3.1 3D 建议框优化 .....	38
3.3.2 3D 检测框编码 .....	38
3.4 损失函数 .....	39

3.5 实验及分析 .....	40
3.5.1 实验数据集以及评价指标 .....	40
3.5.2 实验细节 .....	40
3.5.3 实验结果 .....	44
3.5.4 消融实验 .....	48
3.6 本章小结 .....	51
<b>第四章 基于稀疏体素图注意力网络的 3D 目标检测方法 .....</b>	<b>52</b>
4.1 体素图网络构架 .....	53
4.1.1 球形体素分组 .....	53
4.1.2 局部点特征表示 .....	53
4.1.3 局部点注意力层 .....	54
4.1.4 全局注意力层 .....	54
4.1.5 体素图特征表示 .....	55
4.2 稀疏到稠密的回归 .....	55
4.3 损失函数 .....	56
4.4 实验及分析 .....	57
4.4.1 实验细节 .....	57
4.4.2 数据集以及评价指标 .....	58
4.4.3 实验结果及分析 .....	58
4.4.4 消融实验 .....	61
4.5 本章小结 .....	64
<b>第五章 总结与展望 .....</b>	<b>65</b>
5.1 全文总结 .....	65
5.2 后续工作展望 .....	66
<b>致 谢 .....</b>	<b>67</b>
<b>参考文献 .....</b>	<b>68</b>
<b>攻读硕士学位期间取得的成果 .....</b>	<b>75</b>

# 第一章 绪论

## 1.1 研究背景及意义

根据世界卫生组织（WHO）的估计，每年全球平均道路死亡人数约为 120 至 135 万人，遭受非致命伤害和/或残疾的人数约为两千万至五千万<sup>[1-2]</sup>；此外，道路交通撞车是 5 至 29 岁之间人群死亡的主要原因<sup>[1]</sup>。随着计算机技术与通信技术的快速发展，物联网和智能化技术逐渐在车辆上得到使用，学术界和工业界都将自动驾驶汽车视为减少当前交通死亡人数的重要战略。由于自动驾驶汽车的普及，预计目前的交通死亡人数将减少 75-80%<sup>[3]</sup>。但是，自动驾驶汽车的影响将落后于碰撞率，因为自动驾驶汽车还有望大大降低保险成本（道路伤害的经济负担相当于 2015 年至 2020 年对全球国内生产总值的 0.12% 的年度税收<sup>[4]</sup>），提高非燃料驾驶员的出行效率，提高道路效率，并由于提高燃料效率和减少排放而减少人类出行对环境的影响<sup>[5]</sup>。



图 1-1 自动驾驶 Apollo 组件

为了简化沟通并促进技术和政策领域内的协作，美国汽车工程师协会（SAE）于 2014 年发布了名为 J3016 “驾驶自动化水平”的标准。其中，制定了自动驾驶的通用分类法和定义，分为六个级别驾驶自动化的程度是根据驾驶员的干预量和所需的专注度来定义的<sup>[6]</sup>。级别从无驾驶自动化到全面驾驶自动化。该标准指出，车辆感知是使车辆达到自动驾驶最高水平的关键组成部分，车辆感知的发展集中在检测道路标记和检测周围物体（例如其他车辆，行人，骑自行车的人和标志）的任务上。目前，国内外的汽车企业、科研机构、高校都在自动驾驶领域中的车辆感

知中积极开展活动，很多企业已经自动驾驶纳入智能城市的布局之中，如图 1-1 所示是百度的 Apollo 组件，其中包括感知、仿真、高精度地图与定位、决策规划智能控制及数据集搜集等众多任务。

为了有效地执行上述任务，自动驾驶汽车必须从周围环境中收集重要信息，并从中提取相关知识，以便最终根据其语义对数据进行分类，甚至预测其未来状态<sup>[7]</sup>。为此，感知系统可能使用单一的采集技术或多个传感器来连续扫描和监视环境，类似于人类的视觉和其他感官<sup>[8]</sup>，它涉及收集，过滤和处理来自多个传感器的原始数据<sup>[9]</sup>。



图 1-2 环境感知的复杂场景示例

由于 3D 感测技术的飞速发展，基于 LIDAR 的 3D 扫描仪正在变得越来越广泛，该方法使用脉冲激光形式的光来测量距地球的距离，为了精确测量传感器与周围障碍物之间的距离，LIDAR 可以同时提供丰富的几何，形状和比例信息<sup>[10-12]</sup>，该技术测量了发射和检测反射激光之间的时间，以提取物体与传感器之间的度量空间<sup>[13]</sup>。每个 LIDAR 扫描都会生成一个 3D 点云，其中包括周围场景的图形表示，其中每个点都包含有关其欧几里得距离的信息。这种类型的传感器能够在不同的照明条件下提供远程检测能力，高分辨率和良好的性能。但是，有几个因素使对点云的感知任务非常具有挑战性，这大大增加了失败的可能性，如图 1-2 所示，其中一些因素与传感器的局限性和采集场景的复杂性有关，即：（1）环境的多样性，尤其要注意在各种天气条件下光度的变化；（2）物体的遮挡和截断，由于物体之间或物体部分之间的视线遮挡超出了传感器范围，导致物体的部分或全部不可见；（3）由于物体的大小以及相同类别的物体，不同类别的物体的表示形式不同，因为与传感器的物体距离会影响采集技术的输出；（4）具有不同结构和类别的整个驾驶领域的性能可靠性。

尽管 LIDAR 传感器已在自动驾驶应用中被广泛采用，但还有其他传感解决方案。基于相机的解决方案具有提供高密度像素强度信息的优势，该信息可以捕获形状和纹理属性<sup>[13]</sup>。基于摄像机的传感器（例如单眼摄像机）具有缺乏深度信息的缺点，而其他基于摄像机的传感器（例如立体摄像机和飞行时间法（ToF））以昂贵的计算和分辨率为代价提供此信息<sup>[14]</sup>。而且由立体相机提供的深度测量的误差与距离成指数地增加，并且比 ToF 相机对照明条件的变化更敏感。当前，许多传统方法，如 GPS 和无线传感器等方法页已经被采用，但鉴于未来路况的复杂性，在更短的时间内获取大量实时数据信息，才会对无人驾驶技术做出一个更准确、更安全的选择。目前常见目标检测算法常集中在 2D 空间，且一般采用人工特征加分类器模式，使用多尺度的滑动窗口来提取目标区域，但因为道路车辆目标多样性，道路环境的复杂性，很难构造一种囊括所有影响因子，在各种场景下对各种目标都表现出良好检测效果的特征，也因此很难有哪种检测算法对所有目标具有普适性性能。因此基于计算机视觉感知的 3D 目标检测方法在检测性能上仍有很大的提升空间，进行这方面的深入研究对无人驾驶技术的发展将会有着重大的理论意义和应用价值。

## 1.2 国内外研究现状

对自动驾驶来说，比较重要的是要能够最周围环境做出准确的判断。当自动驾驶汽车部署在道路上时，它会利用同时定位和地图绘制（SLAM）算法<sup>[15]</sup>来推断其相对于其他车辆，行人和骑自行车的人的空间位置。这种定位和对场景中物体的理解对于自动驾驶汽车做出有关制动、转弯和换道的关键决策至关重要。用于自动驾驶的 SLAM 和场景理解算法的核心推动力是目标检测。考虑到实时情况和物体运动的不确定性，目标检测算法必须是高度准确和稳健的，即，即使在变化的环境条件和明显的遮挡下，该检测算法也不能提供错误的检测。

为了应对这一问题，百度提出 Apollo 开放式的自动驾驶生态，并联合中科慧眼等公司联合开展基于双目 RGB 图像的定位与检测算法研究，国内的其他公司如智加科技、图森未来、小马智行等也纷纷推出了 L2 到 L4 级别的自动驾驶车开展研究与落地，更有包括商汤、minieye 等在内的众多公司提供算法解决方案。自动驾驶有可能从根本上改变城市景观并挽救许多人的生命，安全导航的关键部分是检测和跟踪车辆周围环境。为了实现这一目标，现代自动驾驶汽车部署了多个传感器以及先进的检测和跟踪算法。这种算法越来越依赖于机器学习，这推动了对基准数据集的需求。为了 3D 目标检测的研究有更多的基准，一些汽车公司和科研机构纷纷利用自己推出的自动驾驶汽车来采集不同场景、不同角度、不同环境下的数据，

将数据作为公开数据集并设置相应的评价体系以吸引众多学者开展研究,表 1-1 所示是当下数据种类为多、评价体系较为完善的四个公开数据集的比较。

表 1-1 自动驾驶中公开数据集中各项指标的比较

数据集 属性	KITTI	NuScenes	Waymo	A2D2
LIDAR 传感器	1(64 channels)	1(32 channels)	1+4 aux. (64 channels)	5(16 channels)
水平视场角 FoV (度)	360°	360°	360°	360°
相机	4(0.7 MP)	6(1.4 MP)	3(2.5 MP) + 2(1.7 MP)	6(2.3 MP)
采集地点	城市, 一个城市 (卡尔斯鲁厄)	城市, 两个城市 (波士顿和新加坡)	3 个城市地区 (美国)	城市, 高速公路, 乡村, 道路 (德国三个城市)
采集时间	白天	白天, 夜晚	白天, 夜晚	白天
天气	晴天, 多云	各种天气	各种天气	各种天气
对象	3D	3D	3D, 2D	3D, 像素
类别数	3(×3)	23	4	14
带标签的帧数	20k	40k	230k	12k
3D boxes	200k	1.4M	12M	N.S.
每一帧平均点云 数量	120k	34k	177k	N.S.
数据发布时间	2012 年	2019 年	2019 年	2020 年

如表 1-1 所示, KITTI<sup>[16]</sup> 最早于 2012 年由 Andreas Geiger 等人发布, 该数据集连同相关的基准是该领域的先驱, 在领域内都具有很大的影响力。如图 1-3(a)所示数据收集车配备了四个高分辨摄像机 (两个彩色, 两个灰度), 一个 3D 激光扫描仪和一个 GPS / IMU 惯性导航系统。数据示例如图 1-3(b)所示, 在诸如 2D 和 3D 目标检测、SLAM、深度预测、跟踪和光流等任务上均发布了一些挑战, 基准测试包括 389 对立体和光流图像对, 39.2 km 长的立体视觉测距序列以及在混乱情况下捕获的 200k 3D 对象注释 (每个图像最多可见 15 辆汽车和 30 个行人)。

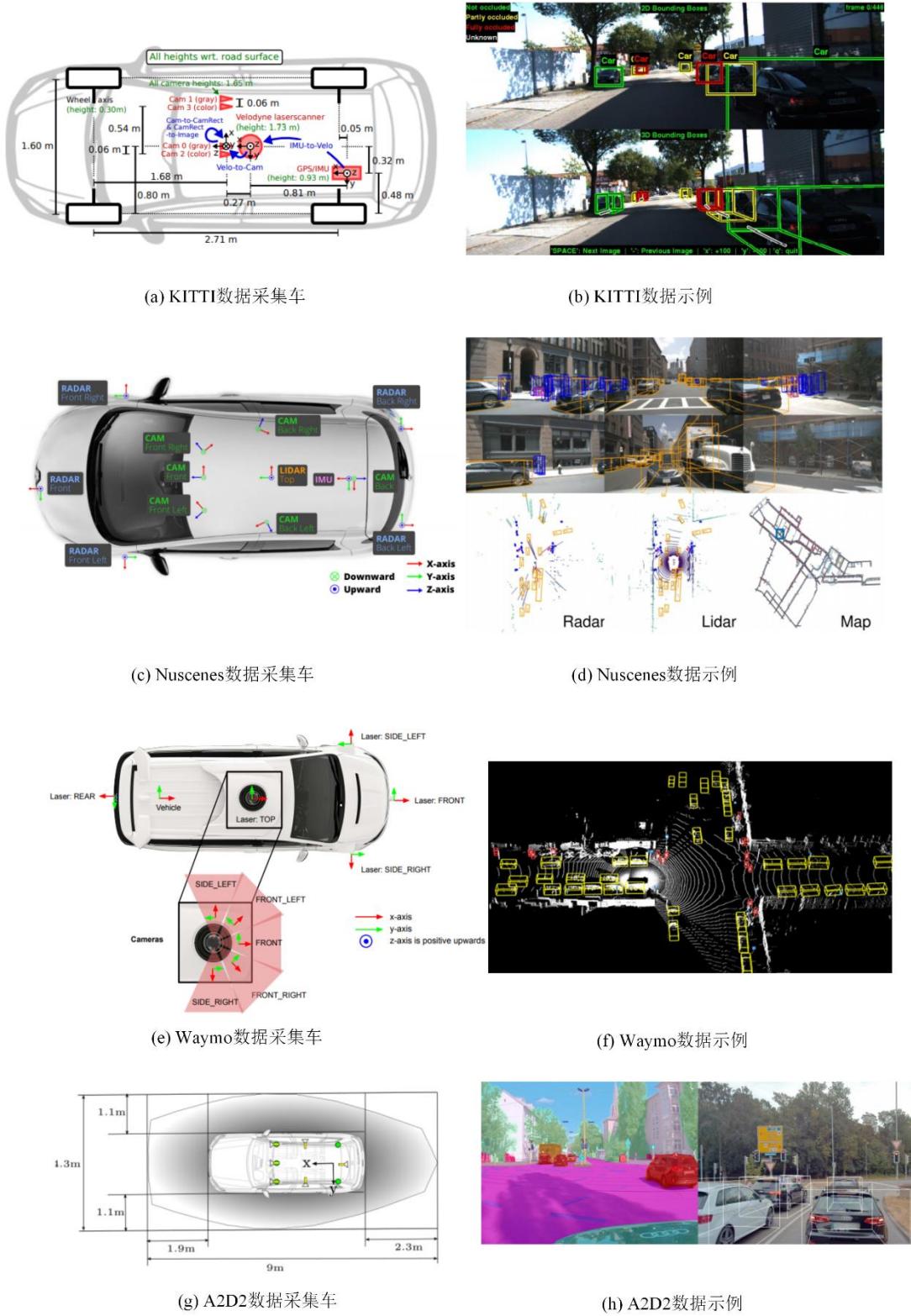


图 1-3 各个数据集的数据采集车与数据示例

NuScenes 数据集由 APTIV 公司于 2019 年正式公布<sup>[17]</sup>, 如图 1-3(c)所示, 这是第一个带有完整的自动驾驶车辆传感器套件的数据集: 6 个摄像头, 5 个雷达和

1 个激光雷达，所有这些都具有完整的 360 度视野。数据示例如图 1-3(d)所示，nuScenes 包含 1000 个场景，每个场景 20 秒长，并用 23 个类别和 8 个属性的 3D 边界框完全注释，它的注释和图像的数量分别是 KITTI 数据集的 7 倍和 100 倍。

由谷歌旗下的 Waymo 公司于 2019 年提供的数据集由 1150 个场景组成，每个场景跨越 20 秒，如图 1-3(e)(f)所示，其中包括在一系列城市和郊区地理区域中捕获的同步良好并经过校准的高质量 LiDAR 和像机数据<sup>[18]</sup>。根据建议的地理覆盖指标，它的多样性是最大的 Camera+LiDAR 数据集的 15 倍。该数据使用 2D（相机图像）和 3D（LiDAR）边界框对这些数据进行了详尽的注释，并在各帧之间使用了一致的标识符，并为 2D 以及 3D 检测和跟踪任务提供了强大的基准。

2020 年，奥迪公布了一个数据集<sup>[19]</sup>，该数据集包括 RGB 图像和 3D 点云，可以用来做 3D 检测，语义分割和实例分割等任务。如图 1-3(g)所示，它的传感器套件包括六个摄像头和五个激光雷达扫描仪，可提供完整的环视覆盖范围。数据示例如图 1-3(h)所示，记录的数据是时间同步的，并且相互注册，注释适用于非顺序帧，具有语义分割图像和点云标签的 41277 帧，其中 12497 帧还具有针对前置摄像头视场内的对象的 3D 边界框注释。

针对上述公开数据集的处理和算法设计也是基于 RGB 和 LIDAR 点云这两种数据的数据特点进行的，而且本文主要是基于 KITTI 中的这两种数据所设计的算法网络结构，下面分布介绍目前常用的针对这两种数据的算法设计处理方案。

### 1.2.1 RGB 图像数据处理

在自动驾驶领域通过相机获得的图像数据一般称为 RGB-D 数据，这里的“D”代表的是 Depth 即深度信息，是由相机所提供的，但是图像 RGB 数据和其他计算机视觉任务中的图像数据基本是一样的，而所携带的这个深度信息往往不会被直接在所设计的网络中使用，只有少数的网络会利用深度信息进行反向投影，如文献 [20-21] 中是利用相机的深度信息在 2D 图像域得到检测框之后映射回 3D 空间，得到目标所在的锥形区域以便进行点云的分割。而在其他的网络中，并没有直接用到此深度信息，因此，对于 RGB 图像的处理和 2D 中图像数据的处理方式是一样的。

在 2D 空间对所得的图像数据进行处理，其目的无非是得到在 2D 空间中的目标检测框或者区域的建议框，以便在后续进行精修或者与点云数据进行融合。在 2D 对 RGB 图像进行处理得到初始建议框的算法中，传统的目标检测算法主要包括目标区域划分，利用人工特征提取算子进行特征提取<sup>[22-23]</sup>，最后利用 SVM<sup>[24]</sup> 等分类器进行类别划分。随着深度学习的发展，2D 目标检测的各种检测网络也层出不穷，按照检测框架的特点往往分为 anchor-based 的方法和 anchor-free 的方法。

在 anchor-based 的方法中又可按照检测阶段划分为两阶段检测网络和单阶段检测网络。两阶段检测网络最典型的是 RCNN 检测算法系列，该系列算法的核心思想是利用一些区域搜索算法先得到一些目标候选区域，再对这些目标候选区域进行 refine，从而生成具有较高鲁棒性的目标框。其中，RCNN 使用的是 RoI + CNN 代替传统的手工提取特征的方法，在提取 Region Proposal 的过程中使用的是 selective search（选择性搜索）<sup>[34]</sup>来提取 2000 多个区域。在 SPP-net<sup>[25]</sup>提出之后，Fast RCNN<sup>[11]</sup>借助 SPP-net 的思想，在网络最后的卷积中加入 RoI Pooling，用 softmax 将 SVM 取而代之，并且提出了一个多任务 multi-stak 的损失函数，将边框回归的过程放在网络之中进行训练，通过这种共享卷积层的方法减少网络的耗时。但是，不难发现，虽然 Fast RCNN 改变了特征分类和特征提取的方案，但是在提取候选框时，仍然使用的是选择性搜索这种与主干网络分离的离线候选框提取方法，这在很大程度上限制了网络的速度。于是，为了进一步加速，Ren 等人于 2015 年提出了 Faster RCNN 网络<sup>[12]</sup>。Faster RCNN 首先将输入的图片送到 CNN 网络中提取 feature map，得到的特征图用 RPN 代替 selective search 得到 region proposal 的特征信息，这里通过设计 9 个不同尺度的 anchor 的形式来对应到原始特征图的所有位置，然后再用分类器判断候选框提取出的特征属于的类别，最后再对候选框进行精修。可以看出，Faster RCNN 通过设计的 RPN 将提取候选框的过程也加入到网络之中进行训练，而不是像 Fast RCNN 那样将提取候选框和精修的过程分开处理，这样全都融合进网络进行训练的方式很大程度上是加快网络速度的关键所在。而对于单阶段检测网络，代表的是 YOLO 系列<sup>[26-28]</sup>、SSD 系列<sup>[29]</sup>等网络。这类端到端的网络本质是使用了回归的思想，输入取的是整张 RGB 图像，将目标在图像上的位置边框及目标的类别同时在图像的多个位置进行预测。不同之处在于，SSD 将回归思想和 anchor 机制进行了完美的结合，既保留了 YOLO 系列算法速度快的特点，又能够使预测的 bounding box 和 Faster RCNN 一样精准。

在 anchor-free 的目标检测算法中，以 CenterNet<sup>[30]</sup>、CornerNet<sup>[31]</sup>等网络为代表，这一类算法又称作是基于点的目标检测，其出发点是为了消除上述 anchor-based 的算法中对于所设计的 anchor 的依赖，将对于目标所在区域边框的检测转化为对于目标所在位置的成对中心点或者角点的检测，再通过这些关键点的几何坐标关系计算出目标的 bounding box。

### 1.2.2 LIDAR 点云数据处理

由于自动驾驶场景的特殊性，在这一场景下需要考虑和可使用的另一大类数据即为由 LIDAR 获得的点云数据。对于来自欧式空间的点的集合来说，它具有三

方面的属性：a) 无序性。点云的表现形式是不存在指定顺序的。换句话说，消耗  $N$  个 3D 点集的网络必须对  $N!$  个输入集按数据馈送顺序的排列保持不变。b) 点之间的相互作用性。所有的点都来自欧氏空间，它们之间是靠距离进行度量的。这就是说，意这些点都不是相互独立的，相邻的点会形成聚合的子集。因此，所设计的网络需要具有判断局部结构的能力，以及计算出局部结构之间的关系。c) 变换下的不变性。作为立体空间的点，所设计的表示形式不能随着变换方法的改变而改变。例如，一起旋转和平移点都不应修改全局点云类别或点的分割。

考虑到点云的上述特性，目前比较经典的用于自动驾驶场景下的点云处理方法为 Pointnet<sup>[32]</sup> 和 Pointnet++<sup>[33]</sup>。除了 Pointnet++ 中使用 MLP 的结构之外，DGCNN<sup>[56]</sup> 引入了边缘卷积的方法并行化处理点云，该网络包含两个分支，一个是分类，另一个是分割。分类模型在加入的所谓边缘卷积，其本质上是考虑到了点云所在的邻域空间，将点云的邻域信息以最大值及最大值池化的形式考虑到了当前点云所在位置中，因此这种表现形式相对于原始的 MLP 是更加符合目标点云的原始分布的。分割模型通过将一维全局描述符和每个点的所有边缘卷积输出（用作局部描述符）进行串联来扩展分类模型。它为  $p$  个语义标签输出每点分类分数。对于点云变换模块：点云变换模块旨在通过应用估计的  $3 \times 3$  矩阵将输入点集与规范空间对齐。为了估计  $3 \times 3$  矩阵，使用了一个张量，该张量将每个点的坐标及其  $k$  个相邻点之间的坐标差连接在一起。边缘卷积模块：边缘卷积模块以形状为  $n \times f$  的张量作为输入，通过应用多层感知器将层神经元的数量定义为  $\{a_1, a_2, \dots, a_n\}$ ，并在相邻边缘特征之间合并后生成形状为  $n \times a$  的张量。

### 1.3 论文的主要研究内容

本文主要围绕自动驾驶场景中的 3D 目标检测问题展开研究，从所利用的数据种类出发，研究了基于 RGB 图像数据和 LIDAR 点云数据的 3D 目标检测算法的算法特点。在此基础之上，首先提出了一种双目 RGB 与 LIDAR 点云融合的算法结构，从数据融合的角度提高检测精度；其次，针对点云数据的处理，提出了一种新的点云特征更新方案，使基于 LIDAR 点云的 3D 目标检测网络更符合点云的真实分布。本文具体的研究内容如下：

1、本文深入研究了自动驾驶领域当下主流的 3D 目标检测算法，并对每一类算法按照所使用的算法结构的不同进行分类归纳与研究。对基于 RGB 图像数据、基于 LIDAR 点云数据和基于单目 RGB 与 LIDAR 数据融合的三维目标检测算法分别了细致的划分与研究。

2、从自动驾驶中可以获取的数据的角度出发，本文深入研究了 RGB 与 LIDAR 点云数据的数据特点。RGB 图像数据带有丰富的语义信息，且有相机提供的深度信息可以作为附件信息可供利用，且双目 RGB 图像相对于单目 RGB 来说，由于其针对同一目标所包含的区域的不一致性，语义信息可被同时使用。LIDAR 点云数据由于具有丰富的空间信息，所带来的检测增益是不同凡响的，因此这两种数据的有效处理与融合将共同对 3D 目标检测的效果带来正向作用。

3、从双目 RGB 数据与 LIDAR 点云数据融合的角度出发，研究设计一种新的基于两种数据融合的 3D 目标检测方案。双目 RGB 图像的共同监督，可以使得在图像域得到的目标候选框更符合物体的真实分布，而增加对点云邻域关系的建立也会更贴合物体的真实分布。

4、从全面构建 LIDAR 数据的邻域关系的角度出发，研究设计一种新的点云特征向量的更新学习方案。考虑到真实场景下物体点云的实际分布，从不同维度建立点云的局部及全局的特征约束关系，从而对每一个点云特征的更新提供监督约束因子，以得到更加聚合的点云特征。

## 1.4 论文的结构安排

本文是基于自动驾驶场景中的 RGB 和 LIDAR 点云数据进行 3D 目标检测算法的研究，论文的结构安排如下：

第一章 绪论。主要介绍自动驾驶的国内外大环境，并阐述 3D 目标检测在自动驾驶场景中的重要意义；分别介绍了当下针对 3D 目标检测任务国内外各个研究机构与公司所开展的挑战，并从自动驾驶中数据处理的角度介绍针对 RGB 和点云数据的主流算法，最后概述本文的主要研究内容与技术路线。

第二章 基于深度学习的三维目标检测算法概述。针对在 3D 目标检测中所能利用的数据，分别介绍了利用 RGB 图像数据、LIDAR 点云数据以及将两种数据进行融合所设计的当下主流的网络结构，并细化了每一种处理方法的算法流程。

第三章 基于双目 RGB 和 LIDAR 点云融合的 3D 目标检测方法。在深入研究进行两种数据融合处理的算法基础之上，设计提出了一种利用双目 RGB 图像数据与 LIDAR 点云数据融合的 3D 目标检测方法，提出了新的构建点云邻域信息的点云分割方案，并利用线性变换优化了检测框回归中使用的冗余向量问题，最后将实验结果与当下主流算法进行比较并设计相应的消融实验进行验证。

第四章 基于稀疏体素图注意力网络的 LIDAR 点云 3D 目标检测方法。为了更好的构建点云的邻域关系并对点云的特征进行全面的监督，本章提出了一种新的点云特征更新方式，从局部和全局两个维度分布引入注意力机制，在两个层面为点

云的特征更新分别学得一个约束参数；在分类和回归网络上，设计了一个多层级、多尺度特征向量融合方法。最后将实验结果与当下主流算法比较，并设计消融实验进行验证。

第五章 总结与展望。对论文的研究成果进行全面的总结，对当下遇到的问题进行分析并提出未来可能的解决方案与研究方向。

## 第二章 基于深度学习的 3D 目标检测算法概述

近年来，随着自动驾驶如火如荼的发展，在该领域内的 3D 目标检测任务也吸引了越来越多研究者的注意，众多自动驾驶公司和该方向的研究机构纷纷发布了数据集和挑战，其中吸引人数最多、影响最为广泛且评价体系最为完善的是 Geiger 等人于 2012 年发布的 KITTI 这一 benchmark<sup>[16]</sup>，这一挑战于 2012 年最初发布，于 2015 年进行了数据集更新，于 2019 年统一更换了评价指标，并于 2020 年还发布了 KITTI-360 环式雷达扫描数据。如图 2-1 所示，基于 KITTI 这一数据挑战，在近几年发表于各大顶会和期刊的 3D 目标检测相关的论文层出不穷，各个算法从数据利用与处理的角度分别设计了不同的网络结构，并达到不错的效果。本章将从只利用 RGB 图像数据、只利用 LIDAR 点云数据及 RGB 图像数据与 LIDAR 点云数据融合这三个角度出发，分别介绍挑出各个类别下当下主流的算法思想与网络结构框架。

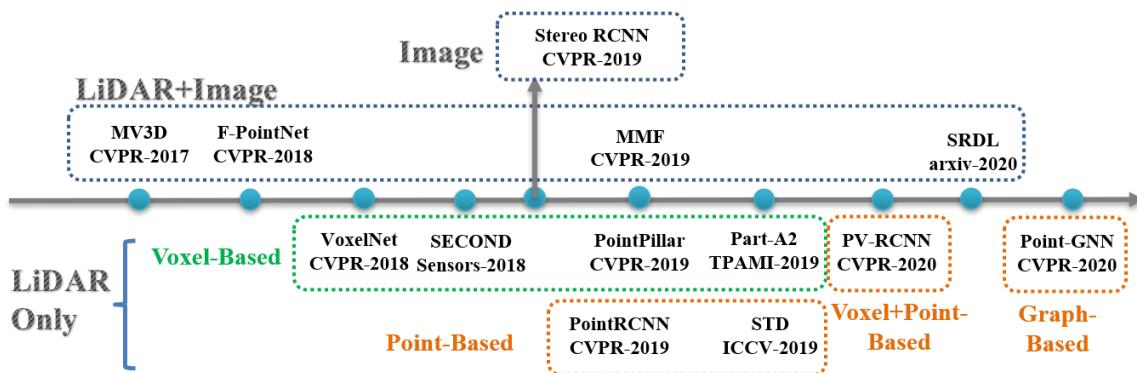


图 2-1 基于 KITTI 数据集的 3D 目标检测算法分类

### 2.1 基于 RGB 图像数据的 3D 目标检测算法

由视觉相机获得的 RGB 图像数据，由于具有丰富的语义信息，且可以结合成熟的 2D 中对于 RGB 图像处理的各种 CNN 网络被广泛应用在 3D 目标检测中。但是由于在 3D 目标检测任务中需要估计三维的几何位置信息，RGB 图像数据相对于 LIDAR 点云数据又不占优势，因此只利用 RGB 图像数据进行 3D 目标检测算法设计的论文并不多，但还是有一些优秀的工作，从不同角度出发提取图像数据的特征完成 3D 目标检测任务，本小节将从利用单目与双目 RGB 图像数据出发，分别介绍基于单目 RGB 数据与双目 RGB 数据的 3D 目标检测算法的网络结构、算法的核心思想。

### 2.1.1 基于单目 RGB 图像数据的 3D 目标检测算法

受到二维目标检测网络的启发，常用的基于单目 RGB 图像的三维目标检测一般会设计出相似的结构，而且网络的重点主要在如何在二维图像中提取特征。但是这些特征由于缺少了空间信息而不适用于 3D 检测任务，这也是目前大多数算法的性能并不好的原因之一。发表于 2019 年的 CE-Net<sup>[35]</sup>这篇文章与以前的基于图像的方法侧重于从 2D 图像提取的 RGB 特征不同，该网络在重构的 3D 空间中解决了此问题，以便明确地利用 3D 上下文。如图 2-2 所示，CE-Net 首先利用一个独立的模块将输入数据从 2D 图像平面转换到 3D 点云空间以获得更好的输入表示，然后使用 PointNet 骨干网络执行 3D 检测以获得对象的 3D 位置，尺寸和方向。为了增强点云的判别能力，提出了一种多模式特征融合模块，将互补 RGB 线索嵌入到生成的点云表示中。CE-Net 证明了与图像平面（即 R, G, B 图像平面）相比，从生成的 3D 场景空间（即 X, Y, Z 空间）推断 3D 边界框更有效。

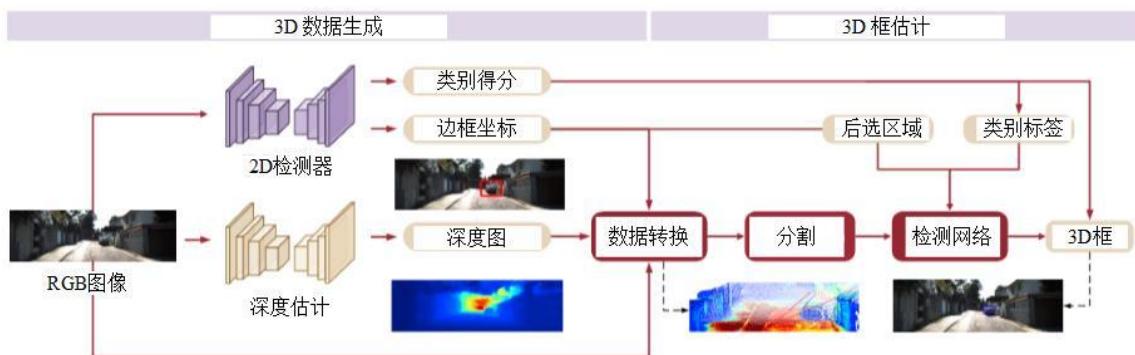


图 2-2 CE-Net 网络结构图

如图 2-2 所示，CE-Net 提出的 3D 检测框架包括两个主要阶段：三维数据生成和 3D 框预测。第一个阶段使用两个 CNN 网络检测出 location 和 depth 信息，并将 depth 信息转移到点云中，从而获得 ROI 的先验信息。在第二个阶段，为了进行准备的边框预测，该网络设计了两个模块，这两个模块一个用于背景点分割，而另一个用于聚合 RGB 图像的信息。最后，为了预测每个检测框的位置，尺寸和方向，该网络使用了 PointNet 作为骨干网络。

#### (1) 3D 数据生成

基于单目图像的三维检测之所以存在较大困难，是因为在很多网络中图像外观没有得到很好的确定。因此，CE-Net 设计了两个深度卷积网络分别完成两项任务，一项是生成深度图，另一项是预测二维的建议框的空间位置信息。

对于输入图像的表示重点在于如何使用深度信息而不是在于如何得到深度信息。因此，CE-Net 将预测出的深度信息由 KITTI 提供的相机矩阵转换为点云，然后将点云再作为输入数据。

具体来说，对于 2D 图像空间中具有深度为  $d$  的像素坐标  $(u, v)$ ，可以将相机坐标系中的 3D 坐标  $(x, y, z)$  计算为：

$$\begin{cases} z = d \\ x = (u - C_x) * z / f \\ y = (v - C_y) * z / f \end{cases} \quad (2-1)$$

其中  $f$  是相机焦距， $(C_x, C_y)$  是相机畸变中心。输入的点云  $S$  可以由深度图和 2D 的 bounding box  $B$  通过如下方式生成：

$$S = \{p \mid p \leftarrow F(v), v \in B\} \quad (2-2)$$

其中  $v$  是深度图中的像素点， $F$  是由公式 (2-1) 带来的转换函数。

### (2) 3D 框预测

通过上述 3D 数据生成阶段，输入数据被编码成了点云，但是这些点云中却有背景点，这些背景点对于精准预测目标的位置来说是会造成干扰的，因此应该被丢弃掉。尽管已经有在第一章中介绍的 PointNet 来解决在 LiDAR 数据中的问题，但是这个算法对 3D 目标的 ground truth 进行预处理以生成分割的标签，而且由于在上一步生成的点是不稳定的，所以即使使用相同的标签生成方法也会带来严重的噪声。因此，CE-Net 基于深度的先验信息提出了一种简单但有效的分割方法。具体来说，首先在每个 2D 边界框中计算深度平均值，以获取 ROI 的近似位置，并将其作为阈值，在  $Z$  通道上所有大于该阈值的点均视作背景点，该处理后的点云集合表示为：

$$S' = \left\{ p \mid p_v \leq \frac{\sum_{p \in S} p_v}{|S|} + r, p \in S \right\} \quad (2-3)$$

这里的  $p_v$  代表的是点云在  $Z$  通道上的值， $r$  是用于校正阈值的偏差。最后，随机选择点集  $S'$  中固定数量的点作为该模块的输出，以确保后续网络输入点数的一致性。

在预测最终的检测框之前，CE-Net 遵循文献[20]中的方法，使用参数量较少的网络预测感兴趣区域的中心  $\delta$ ，并使用它来更新点云，如下所示：

$$S'' = \{p \mid p - \delta, p \in S'\} \quad (2-4)$$

其中  $S''$  是进行最终检测任务的点集。然后再使用 PointNet 作为 3D 检测的 backbone 来编码目标的中心  $(x, y, z)$ ，尺寸  $(h, w, l)$  和仰角  $\theta$ 。

### (3) RGB 信息聚合

为了进一步提高算法的性能和鲁棒性，该网络将互补的 RGB 信息聚合到点云中，具体来说，通过将公式 (2-2) 替换为如下公式的方式添加 RGB 信息：

$$S = \{p \mid p \leftarrow [F(v), D(v)], v \in B\} \quad (2-5)$$

其中  $D$  是对输入的点输出相应 RGB 信息的函数。这样，这些点就被编码为一个六维的向量： $[x, y, z, r, g, b]$ 。然而，用这种简单的融合方式也不可行，因此，该算法使用了如图 2-3 所示的注意力机制来完成信息融合任务。

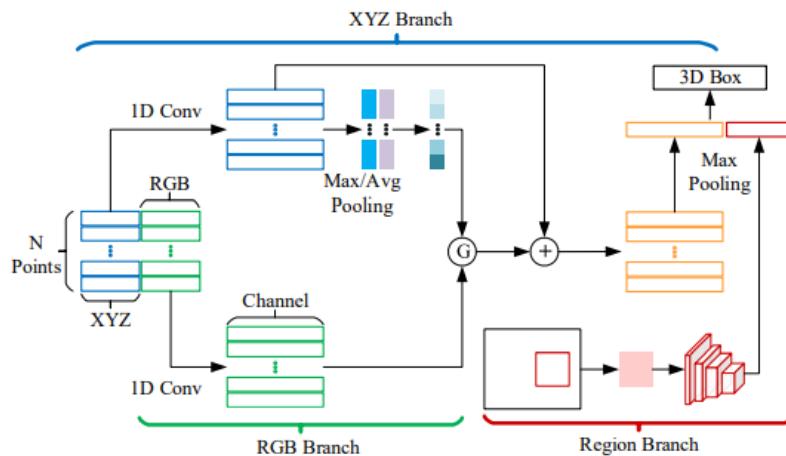


图 2-3 带有 RGB 信息融合的 3D box 检测模块

如图 2-3 所示，该网络借鉴了注意力机制的特点，其目的是传递空间特征和图像数据特征之间的信息。当图像数据的信息传递到对应的点与云上时，三个坐标分支生成的 feature map 会生成 attention 图  $G$ ，这个过程可以用如下所示：

$$G \leftarrow \delta(f([F_{\max}^{xyz}, F_{avg}^{xyz}])) \quad (2-6)$$

其中  $f$  是从一个卷积层中学得的非线性函数， $\sigma$  是 sigmoid 激活函数。然后，信息在注意力图的控制之下通过如下方式传递：

$$F^{xyz} \leftarrow F^{xyz} + G \otimes F^{rgb} \quad (2-7)$$

其中  $\otimes$  表示逐像素相乘。

### 2.1.2 基于双目 RGB 数据的 3D 目标检测算法

不同于单目 RGB 图像数据，为了获得更精确的深度信息，双目摄像机的优势就体现出来了。而和 LiDAR 相机相比，双目相机价格低廉，同时对于具有非凡差异的物体也可达到相当的深度精度。出于这种考虑，Disp R-CNN<sup>[38]</sup>提出了一个实例视差估计网络（iDispNet），该网络仅预测感兴趣目标上特定类别的形状像素的视差。由于所获得的训练数据可能存在人为标注导致视差不一致的情况，该网络提出将目标模型的形状进行计数的方法来生成 groundtruth 的视差，这样就避免了激光点云的使用。

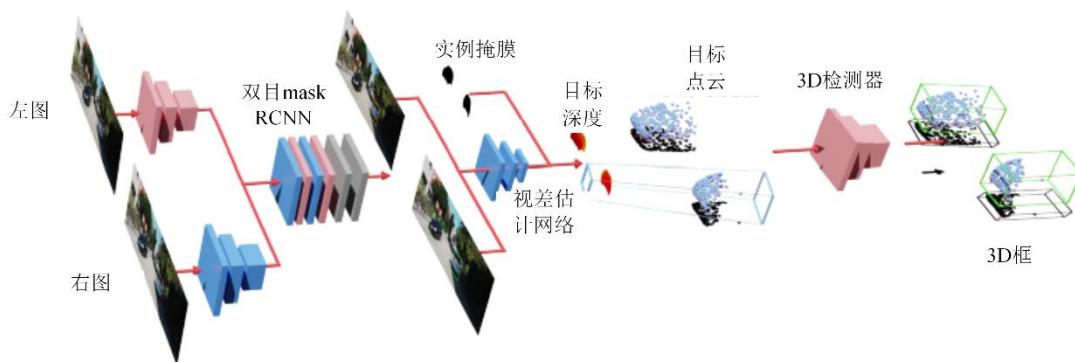


图 2-4 Disp R-CNN 网络结构图

如图 2-4 所示，Disp R-CNN 主要由三个阶段构成。首先生成每个目标的 2D 边界框和实例 mask，然后仅估计属于目标像素的视差，最后使用 3D 检测器从实例点云中预测 3D 边界框。

#### (1) Stereo Mask R-CNN 模块

首先简要描述基础 2D 检测器，该检测器为接下来模块提供必要的输入。该网络扩展了 Stereo R-CNN<sup>[36]</sup>框架，以预测左侧图像中的实例分割 mask。Stereo Mask R-CNN 由两个阶段组成。第一阶段是将[36]中提出的区域提议网络（RPN）进行的改进。第二阶段使用[37]中提出的 RoIAlign 从特征图中提取目标特征。

#### (2) 实例视差估算网络

在使用双目 RGB 图像进行 3D 目标检测中，重点在于如何恢复 3D 数据，而恢复 3D 数据的关键在于视差估计，这直接影响 3D 检测的性能。先前的工作<sup>[48]</sup>应用了现成的视差估计模块，该模块预测整个图像中所有像素的视差图。由于前景对象的区域仅占整个图像的一小部分，因此视差估计网络和目标检测网络中的大多

数计算都是多余的，可以减少。此外，对于大多数车辆上的镜面而言，不能用于立体匹配中用于光度一致性约束的朗伯反射率假设。为了解决这些问题，该网络提出了实例视差估计网络（iDispNet），该网络是一种专门用于 3D 目标检测的视差估计网络。

具体来说，像素  $p$  的全帧视差定义为：

$$D_f(p) = u_p^l - u_p^r \quad (2-8)$$

这里  $u_p^l$  和  $u_p^r$  分别代表了像素  $p$  在左视角上竖直像素的坐标。使用 Stereo Mask R-CNN 产生的 2D 边界框，可以从完整图像中裁剪出左右 RoIs，并在水平方向上对齐它们。将每个 RoI 的宽度( $w^l, w^r$ )设置为较大的值，以使两个 RoI 共享相同的大小。一旦 RoI 对齐，左图像上像素  $p$  的视差位移将从全帧视差变为实例视差，其定义为：

$$D_i(p) = D_f(p) - (b^l - b^r) \quad (2-9)$$

这里  $b^l$  和  $b^r$  分别代表两个视图中边界框左边缘的坐标。我们的目标实质上是为属于感兴趣对象的每个  $p$  学习实例视差  $D_i(p)$  而不是  $D_f(p)$ 。

左右图像中的所有感兴趣区域的大小为  $H \times W$ 。对于实例分割 maks 给出的属于对象实例  $O$  的所有像素  $p$ ，实例 disparity 的损失函数定义为：

$$L_{tdisp} = \frac{1}{|O|} \sum_{p \in O} L_{i,smooth}(\hat{D}_i(p) - D_i(p)) \quad (2-10)$$

$$\hat{D}_i(p) = \frac{D_i(p)}{\max(w^l, w^r)} W \quad (2-11)$$

一旦 iDispNet 输出实例视差，我们就可以计算属于前景的每个像素  $p$  的 3D 位置，作为后续 3D 检测器的输入。3D 坐标  $(X, Y, Z)$  的推导如下：

$$\begin{aligned} X &= \frac{(u_p - c_u)}{f_u} Z, Y = \frac{(v_p - c_v)}{f_v} Z, \\ Z &= \frac{Bf_u}{\hat{D}_i(p) + b^l - b^r} \end{aligned} \quad (2-12)$$

其中  $B$  是左右摄像机之间的基线长度， $(c_u, c_v)$  是对应于摄像机中心的像素位置， $(f_u, f_v)$  分别是水平和垂直焦距。

### (3) 伪背景生成模块

训练立体匹配网络需要大量密集的视差背景，而大多数 3D 目标检测数据集由于手动注释方面的困难而无法提供此数据。在最近的工作中<sup>[48-49]</sup>使用的全帧视差估计模块首先在合成数据集上进行了预训练，然后使用从 LiDAR 点转换而来的稀疏视差真相对实际数据进行了微调。尽管在这种监督下检测性能得到了很大的提高，但由于传感器价格高昂，对 LiDAR 点云的要求限制了现实情况下双目 3D 目标检测方法的缩放能力。

受益于仅需要前向监督的 iDispNet 的设计，该网络提出了一种有效的方法，无需 LiDAR 点即可为真实数据生成大量的密集视差伪背景（pesudo-GT）。该网络使用的形状表示为体积截断符号距离函数（TSDF）<sup>[50-51]</sup>，公式为：

$$\tilde{\phi}(z) = Vz + \mu \quad (2-13)$$

其中  $z$  是形状系数。这里用  $b$  表示双目像机的基线长度。

## 2.2 基于 LIDAR 点云数据的 3D 目标检测算法

本小节将从不同处理点云数据的方法的角度出发，介绍利用 LIDAR 进行 3D 目标检测的各个算法的经典网络。

### 2.2.1 体素化 3D 目标检测

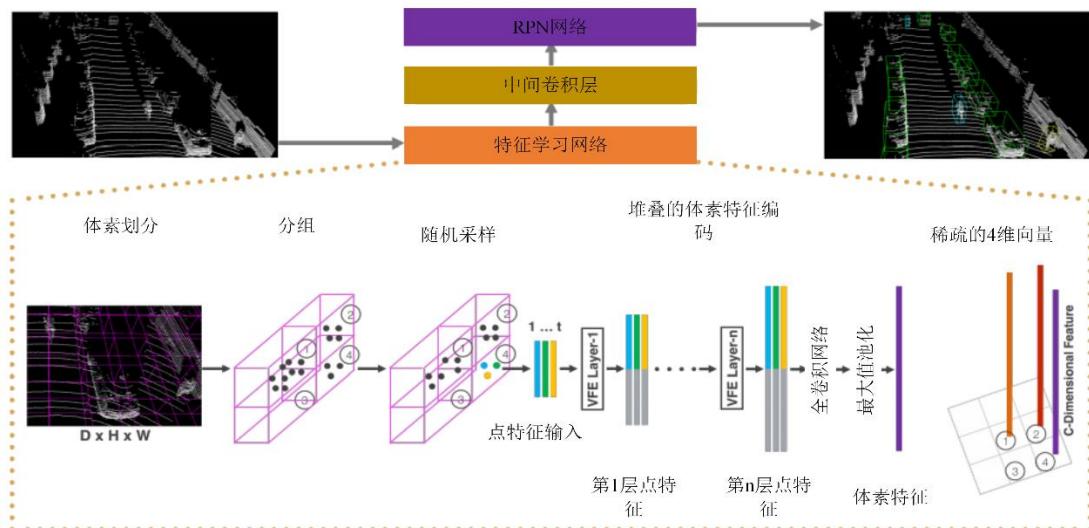


图 2-5 VoxelNet 网络结构图

RPN<sup>[12]</sup>结构在目标检测中已经被高度优化的算法，但是这个算法要求输入数据是以向量的结构紧密且有规律的排列的，这对于无序排列的 LiDAR 点云来说直

接使用是不可行的，因此将点云数据转换为 RPN 网络可以使用的数据结构是众多算法的出发点。其中，将点云进行体素化处理便是一种常用的方法，所谓的体素化是将三维的点云空间进行规则的划分，所切分的小的长方体结构便称作体素。VoxelNet<sup>[39]</sup>便是基于体素化思想的典型算法结构，如图 2-5 所示，VoxelNet 设计了一种新颖的体素特征编码（VFE）层，通过将逐点特征与局部聚合的特征相结合，可以在体素内进行点间交互。最后，RPN 作为检测 head 产生最终的检测结果。

如图 2-5 所示，对于给定的一组点云，首先将 3D 空间划分成均匀的空间体素。假设输入的点云在  $Z, Y, X$  坐标的范围为  $D, H, W$ ，每一个体素的尺寸为  $v_D, v_H, v_W$ ，则最终的 3D 体素大小为：

$$D' = D / v_D, H' = H / v_H, W' = W / v_W \quad (2-14)$$

为了简化，假定  $D, H, W$  可以被  $v_D, v_H, v_W$  整除。

在划分完空间的体素之后，将每个点根据所在的位置进行分组，然而由于距离、遮挡、目标姿态和非统一的采样等问题，空间中的点是稀疏且密度不均匀的，因此所分组的体素之间每一个体素所包含的点的数量也是不一样的。并且对于高分辨率的 LIDAR 点云来说，扫描得到的点通常有约十万个，直接处理如此庞大数量的点在计算平台上增加内存的同时，会极大的影响效率，而且在空间中点的密度的影响下，检测的方向会被改变。为此，定义一个阈值  $T$ ，我们从那些包含多个  $T$  点的体素中随机抽取固定数量的点  $T$ 。

VoxelNet 的关键创新在于所构建的 VFE-layer。假设一个非空的体素包含了  $t \leq T$  个点，则它可以表示为：

$$V = \left\{ p_i = [x_i, y_i, z_i, r_i]^T \in \Re^4 \right\}_{i=1 \dots t} \quad (2-15)$$

其中  $p_i$  表示第  $i$  个包含了 XYZ 坐标的点， $r_i$  表示反射率。首先计算  $V$  中所有点的坐标均值作为当前体素的重心，表示为  $(v_x, v_y, v_z)$ ，然后通过重心来增强每个点  $p_i$  的特征表示，增强后的输入特征集合表示为：

$$\hat{V}_{in} = \left\{ \hat{p}_i = [x_i, y_i, z_i, r_i, x_i - v_x, y_i - v_y, z_i - v_z]^T \in \Re^7 \right\}_{i=1 \dots t} \quad (2-16)$$

对于每一个  $\hat{p}_i$  通过全连接网络 (FCN) 转换到特征空间，从而得到编码了当前体素表面形状的特征  $f_i \in \Re^m$ ，全连接网络由一个线性层，一个 BN 层和一个 ReLU 组成。在获得了逐点的特征表示后，在与  $V$  相关的所有  $f_i$  上使用基于元素的最大值

池化操作，以获取  $V$  的局部聚合的特征  $\tilde{f} \in \Re^m$ 。最终使用  $\tilde{f}$  来增强每一个  $f_i$  以得到基于点的聚合特征：

$$f_i^{out} = \begin{bmatrix} f_i^T, \tilde{f}^T \end{bmatrix}^T \in \Re^{2m} \quad (2-17)$$

这样，输出的特征可以用集合表示为  $V_{out} = \{f_i^{out}\}, i=1\dots.t$ 。

对所有的非空体素进行上述相同的处理，所得到的基于体素的特征是一个 4D 的特征向量，可表示为  $C \times D \times H \times W$ 。接着用  $ConvMD(c_{in}, c_{out}, k, s, p)$  表示 M 维的卷积操作，这里  $c_{in}, c_{out}$  输入与输出通道数， $k, s, p$  是 M 维向量对应的卷积核尺寸、步长和 padding 尺寸大小。每一个卷积的主体网络是 3D 卷积，但是每一个卷积层后面都会紧跟着一个 BN 层和 ReLU 层。这样依次递减的尺度大小会使感受野范围逐渐扩大，但是基于体素的特征和目标形状的语义信息在不断聚合。

## 2.2.2 基于 PointNet 的 3D 目标检测

除了上一小节中的体素化处理点云的方法之外，第二类处理 LIDAR 点云的数据的方法是基于 PointNet 的算法，如在第一章中所介绍的，PointNet 以原始点云作为输入，再基于每个点来预测候选框，这种可以作为骨干网络的算法被很多文章所采用，如文献等等。这些基于 PointNet 的 3D 目标检测网络通常由两阶段组成，在第一阶段，首先利用集合抽象（SA）层进行下采样和提取上下文特征。之后，将特征传播（FP）层应用于上采样，并将特征传递到在下采样期间丢弃的点。然后，将 3D 建议框提取网络（RPN）应用于生成以每个点为中心的投标。基于这些建议框，设计一个优化模块作为第二阶段以提供最终预测。这些方法可以获得更好的性能，但是在许多实时系统中，它们的推理时间通常是无法忍受的。

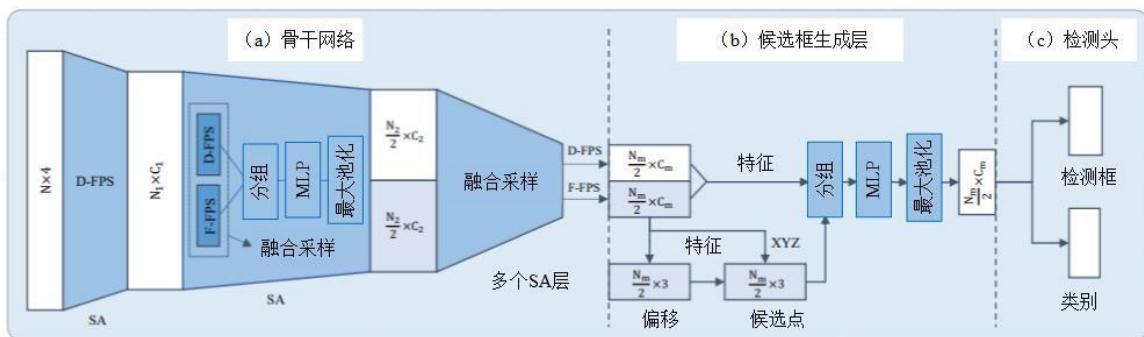


图 2-6 3DSSD 网络结构图

基于对于 3D 目标检测系统中精度与速度的平衡，3DSSD<sup>[40]</sup>这篇算法网络基于 PointNet 做出了重要的改进，如图 2-6 所示，该算法将在所有现有的基于点的方法中必不可少的所有上采样层和优化阶段都被放弃，以减少大量的计算成本。并创新地提出了在下采样过程中的融合采样策略，以使在代表性较小的点上的检测变得可行。

为了保留正样本点（任何实例中的内部点）并消除那些毫无意义的负样本点（位于背景上的点），不仅要在采样过程中考虑空间距离，还要考虑每个点的语义信息。而且可以看到，深度神经网络可以比较不错的捕捉语义信息。因此，利用特征距离作为 FPS (furthest point sampling) 的标准，许多相似的无用负样本点将被去除，例如大量的地面点。由于不同目标点的语义特征不同，因此距离较远的正样本点可以被保留下来。

然而，仅将语义特征距离作为唯一标准将在同一实例中保留许多点，这也引入了冗余。例如，对于一辆汽车，窗户和车轮周围的点的特征之间存在很大差异。结果，将对两个部分周围的点进行采样，而任一部分中的任何点都将为回归提供信息。为了减少冗余并增加多样性，在 FPS 中应用空间距离和语义特征距离作为准则，公式为：

$$C(A, B) = \lambda L_d(A, B) + L_f(A, B) \quad (2-18)$$

这里  $\lambda$  为平衡参数， $L_d(A, B)$  和  $L_f(A, B)$  分别表示两点之间的 L2 XYZ 距离和 L2 特征距离。这篇文章的作者将这种采样方法称为 Feature-FPS (F-FPS)，且证明了比原始的 D-FPS 采样方法在样本保持率上高 20%。

通过 F-FPS，在 SA 层之后可以保留不同实例中的大量正样本点。但是，在固定的总代表点数  $N_m$  的限制下，在降采样过程中会丢弃许多负样本点，这有利于回归任务，但会妨碍分类。换句话说，由于 SA 层中有很多的邻域特征，因此负样本点在缺少邻域信息的情况下即使被很好的分组其感受野也很小。结果，该模型发现难以区分正负样本点，导致分类性能较差。也就是说尽管具有 F-FPS 的模型比具有 D-FPS 的模型具有更高的召回率和更好的定位精度，但它更喜欢将许多负样本点视为正样本点，从而导致分类精度下降。

因此，在 SA 层之后，不仅应尽可能多地采样正样本点，而且还需要收集足够的负样本点以进行更可靠的分类。基于此，该算法提出了一种新颖的融合采样策略 (FS)，其中在 SA 层中同时应用了 F-FPS 和 D-FPS，以保留更多用于定位的正样本点并保留足够的用于分类的负样本点。具体来说，分别使用 F-FPS 和 D-FPS 采样  $N_m / 2$  点，并将这两个集合一起馈送到 SA 层中的分组操作。

在由几个 SA 层实现的骨干网络与融合采样交织在一起之后，便可以从 F-FPS 和 D-FPS 中获得了一个点子集，这些点用于最终预测。在以前的基于点的方法中，应在预测 head 之前应用另一个 SA 层来提取特征。常用的 SA 层包括三个程序，分别为选择中心点，搜索邻域点和生成语义信息。

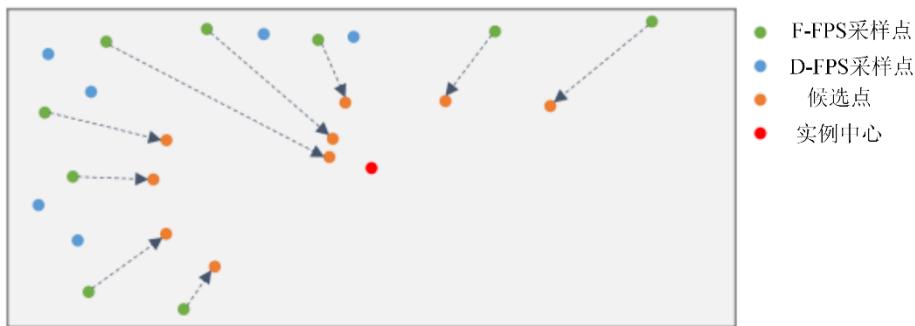


图 2-7 CG 层中移位操作的图示。灰色矩形代表具有 F-FPS（绿色）和 D-FPS（蓝色）的所有正代表点的实例。红点代表实例中心。

为了更好的减少计算成本并完全使用融合采样，该算法在 SA 层的基础之上进行改造，提出了候选生成层（CG），该层是用于预测 head。由于 D-FPS 的大多数代表点都是负样本点，并且在包围盒回归中没有用，因此仅将 F-FPS 的代表点用作初始中心点。实例点的作用是对初始点提供监督因子，以移动初始点的位置。如图 2-7 所示，初始点在移位后被表示为候选点。然后，候选点作为候选生成层的中心点吗，这样的目的是为了优化性能。接下来，从包含 D-FPS 和 F-FPS 的点的整体代表点集中找到每个候选点的周围点，这些点具有预定义的范围阈值，候选点的归一化位置和语义特征被 concatenate 在一起，然后将 concatenate 之后的特征送到多层感知机中用于回归和分类。

通过融合候选生成层和上述的采样策略，该模型可以替代 FP 层和优化模块，并且不会增加额外的时间。在回归 head 中，面临两个选择，基于锚的或无锚的预测网络。如果采用基于锚的头部，必须构造多尺度和多方位的锚，以覆盖具有不同大小和方向的对象。为避免繁琐的多个锚设置，并与轻量级设计保持一致，该算法使用了无锚回归头。

在训练过程中，往往需要一种分配策略来为每个候选点分配标签。在 2D 单阶段检测网络中，通常使用交叉相交（IoU）<sup>[29]</sup>阈值或 mask<sup>[41-42]</sup>为像素分配标签。FCOS<sup>[41]</sup>提出了一个连续的中心度标签，代替了原始的二进制分类标签，以进一步区分像素。与基于 IoU 或基于 mask 的分配策略相比，它将较高的中心度分数分配给更靠近实例中心的像素，从而获得相对更好的性能。但是，直接将中心标签应用

于 3D 检测任务并不可行，因为所有 LIDAR 点都位于物体表面上，它们的中心度标签都非常小且相似，因此无法将良好的预测与其他点区分开。

因此，该算法没有利用点云中的原始代表点，而是求助于预测的候选点，这些候选点受到监督以靠近实例中心。由于越靠近中心所获得的位置信息更准确，3D 中心度标签也越能够轻松的区分它们。对于每个候选点，通过两个步骤定义其中心位置标签。首先确定它是否在实例  $l_{mask}$  中，它是一个二进制值，然后根据其到相应实例的 6 个表面的距离绘制一个中心度标签。中心标签的计算公式为：

$$l_{ctrness} = \sqrt[3]{\frac{\min(f, b)}{\max(f, b)} \times \frac{\min(l, r)}{\max(l, r)} \times \frac{\min(t, d)}{\max(t, d)}} \quad (2-19)$$

其中  $(f, b, l, r, t, d)$  分别表示前、后、左、右、上、下表面的距离最终的分类标签是  $l_{mask}$  与  $l_{ctrness}$  的乘积。

### 2.2.3 体素化与 PointNet 融合的 3D 目标检测

如前所述，体素化可以将点云切分成规则的立方体区域，在每一个规则区域中可以设计不同的采样方法进行标准化处理，而基于 PointNet 的方法可以将其作为一个 backbone，从而提取点云的信息，为了充分利用这两种方法的优点并且加速网络计算，有学者提出了名为 Voxel R-CNN<sup>[86]</sup>的网络，如图 2-8 所示，Voxel R-CNN 包括：（a）3D 骨干网络，（b）2D 骨干网络，后接一个区域提议网络（RPN），以及（c）的 Voxel ROI 池化和检测框精炼网络。在 Voxel R CNN 中，首先将原始点云划分为常规体素，然后利用 3D 骨干网络进行特征提取。然后，将稀疏的 3D 体素转换为 BEV 表示形式，然后在其上应用 2D 骨干网络和 RPN 生成 3D 区域提议。随后，使用 Voxel ROI 池化提取 ROI 特征，这些特征将馈入检测子网以进行检测框的优化。

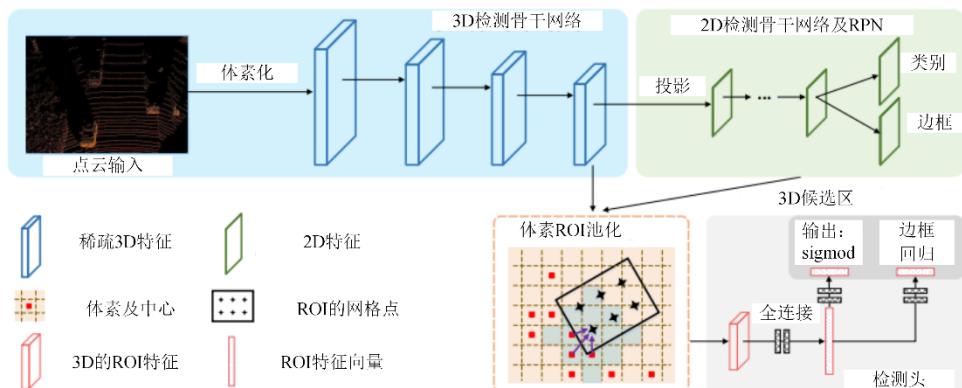


图 2-8 Voxel R-CNN 网络结构图

### (1) Voxel ROI 池化

为了直接从 3D 体素特征量中聚合空间上下文，该网络提出了 Voxel ROI 池化。

**体素体积为点：**该网络将 3D 的稀疏体积空间表示为一组非空的体素中心点集合  $\{v_i = (x_i, y_i, z_i)\}_{i=1}^N$ ，他们对应的特征表示为  $\{\phi_i\}_{i=1}^N$ 。

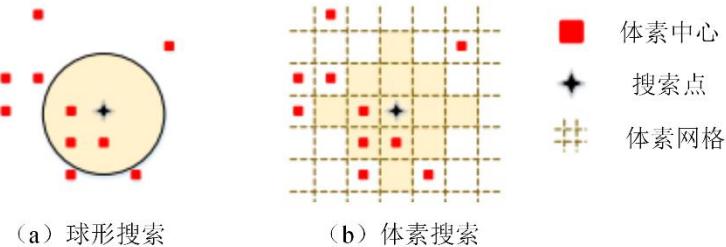


图 2-9 球查询和体素查询

**体素查询：**该网络提出了一个新的操作，称为体素查询，以从 3D 要素中查找相邻体素。与无序点云相比，体素以规则的方式排列在量化的空间中，从而易于邻居访问。例如，可以通过添加三元组偏移量  $(\Delta i, \Delta j, \Delta k)$ ， $\Delta i, \Delta j, \Delta k \in \{-1, 0, 1\}$  在体素指数  $(i, j, k)$  上轻松计算查询体素的 26 个邻域体素。通过利用此属性，该网络设计了体素查询以有效地对体素进行分组。体素查询如图 2-9 所示。首先将查询的目标划分为一个体素，然后可以通过一种搜索方法得到邻域关系。在体素查询中利用曼哈顿距离，并在距离阈值内采样多达 K 个体素。具体来说，在体素  $\alpha = (i_\alpha, j_\alpha, k_\alpha)$  和体素  $\beta = (i_\beta, j_\beta, k_\beta)$  之间的曼哈顿距离  $D(\alpha, \beta)$  表示为：

$$D_m(\alpha, \beta) = |i_\alpha - i_\beta| + |j_\alpha - j_\beta| + |k_\alpha - k_\beta| \quad (2-20)$$

假设 3D 特征量中有 N 个非空体素，并且利用球查询来查找到给定查询点的相邻体素，时间复杂度为 O(N)。尽管如此，进行体素查询的时间复杂度仅为 O(K)，其中 K 是邻居数。邻居感知属性使通过体素查询对邻居体素要素进行分组比通过球形查询对邻居点要素进行分组更为有效。

**Voxel ROI 池化层：**首先将区域建议划分为  $G \times G \times G$  个常规子像素。中心点作为相应子体素的网格点。由于 3D 特征量非常稀疏(非空体素占空间的 3% 以下)，因此无法像在 Fast RCNN 中那样直接利用每个子体素的特征的最大池化。相反，该网络将来自相邻体素的特征集成到网格点中以进行特征提取。具体来说，对于给定的网格点  $g_i$ ，首先利用体素查询将一组相邻的体素分组  $\Gamma_i = \{v_i^1, v_i^2, \dots, v_i^K\}$ ，然后，使用 PointNet 模块汇总相邻的体素特征：

$$\eta_i = \max_{k=1,2,\dots,K} \{\Psi([v_i^k - g_i; \phi_i^k])\} \quad (2-21)$$

其中  $v_i^k - g_i$  代表相关坐标,  $\phi_i^k$  是  $v_i^k$  的体素特征,  $\Psi(\cdot)$  代表了 MLP。 $\max(\cdot)$  代表了最大池化操作。对于每个阶段, 设置两个曼哈顿距离阈值以对具有多个比例的体素进行分组。然后, 将来自不同阶段和规模的聚合特征连接起来, 以获得 RoI 特征。

**局部聚合加速:** 即使使用了该算法前面提出的体素查询, 在 Voxel RoI 池化层中的局部聚合操作仍然包含了巨大的计算复杂度。对于原始 PointNet 模块, 总共有  $M$  个网格点 ( $M = r \times G^3$ , 其中  $r$  是 RoI 数,  $G$  是网格大小), 并且为每个网格点分组了  $K$  个体素。分组的特征向量的维数为  $C + 3$ , 包括  $C$  维体素特征和 3 维相对坐标。当应用 FC 层时, 分组的体素会占用大量内存, 并导致较大的计算 FLOP ( $O(M \times K \times (C + 3) \times C')$ )。

受文献[44,85]的启发, 该算法另外引入了一个加速的 PointNet 模块以进一步降低 Voxel 查询的计算复杂性。具体来说, 对于加速 PointNet 模块, 将体素特征和相对坐标分解为两个流。给定权重为  $W \in \Re^{C,C+3}$  的 FC 层, 将其分为  $W_F \in \Re^{C,C}$  和  $W_C \in \Re^{C,3}$ 。由于体素要素独立于网格点, 因此在执行体素查询之前, 在体素要素上应用带有  $W_F$  的 FC 层。然后, 在体素查询之后, 仅将分组的相对坐标乘以  $W_C$  以获得相对位置特征, 并将它们添加到分组的体素特征中。加速 PointNet 模块的 FLOP 为  $O(N \times C \times C' + M \times K \times 3 \times C')$ 。由于分组的体素 ( $M \times K$ ) 的数量比  $N$  高一个数量级, 因此加速的 PointNet 模块比原始的模块效率更高。

### (2) 场景编码

该文章遵循文献[43,46,84]的类似设计来构建主干网络。3D 骨干网络逐渐将体素化的输入转换为特征量。然后, 沿 Z 轴堆叠输出十个向量, 以生成 BEV 特征图。2D 骨干网络由两个组件组成: 一个具有两个标准  $3 \times 3$  卷积层的自上而下的特征提取子网络, 以及一个对自上而下的特征进行上采样和级联的多尺度特征融合子网络。最后, 将 2D 骨干网络的输出与两个同级  $1 \times 1$  卷积层进行卷积, 以生成 3D 区域提议。

### (3) 检测头

检测头将 RoI 功能作为盒优化的输入。具体而言, 共享的 2 层 MLP 首先将 RoI 特征转换为特征向量。然后, 将展平的特征注入到两个同级分支中: 一个用于 bounding box regression, 另一个用于 confidence prediction。正如文献[45]所推动的那样, 框回归分支预测了 3D 区域建议到地面真值框的残差, 以及置信度分支 预测与 IoU 相关的置信度得分。

### 2.2.4 基于图卷积网络的 3D 目标检测

强大而准确的 3D 检测系统是自动驾驶汽车不可或缺的一部分。传统上，前面介绍的大多数 3D 目标检测算法专注于使用体素网格或鸟瞰(BEV)处理 3D 点云。但是，最近的工作<sup>[53-54]</sup>证明了将图神经网络(GNN)用作 3D 目标检测是一种有效的方法。因此，Thakur S<sup>[55]</sup>等人提出了一种基于注意力的 GNN 特征聚合技术，用于在 LiDAR 扫描中检测物体。该算法首先采用一种距离感知的下采样方案，该方案不仅可以提高算法性能，而且即使对象远离传感器也可以保留其最大的几何特征。在 GNN 的每一层中，除了将每个节点的输入特征映射到相应的更高级别的特征的线性变换之外，还通过为每个节点的第一个环邻域指定不同的权重来掩盖每个节点的注意力。被掩盖的注意力隐含地说明了每个节点的底层邻域图结构，并且还消除了昂贵的矩阵运算的需求，从而在不影响性能的情况下提高了检测精度。

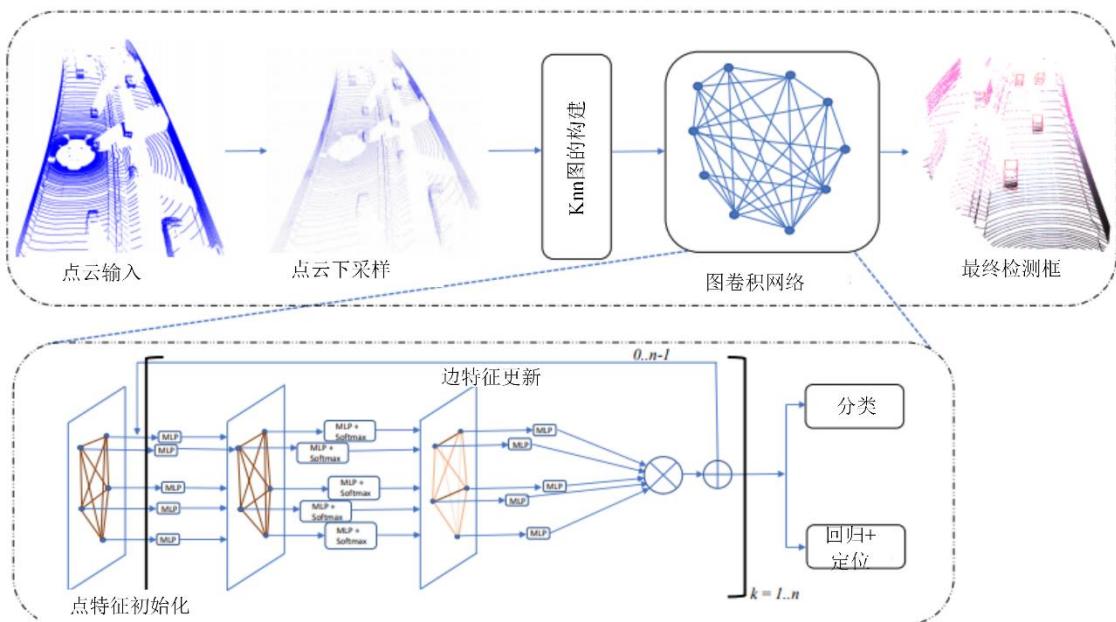


图 2-10 所提出方法的体系结构。方括号中的块构成了所建议 GNN 的一次迭代中的各个步骤

如图 2-10 所示，该算法的网络结构主要由两个模块组成：图构造和基于注意力的 GNN。

#### (1) 图的构建

形式上，点云  $P$  由  $D$  个维度上的  $N$  个点组成，表示为  $P = \{p_i | i = 1, 2, 3 \dots n\} \in \Re^D$ ，其中  $p_i$  是由其坐标  $(x, y, z)$  值和状态值，即反射率值或邻域顶点的编码特征。

距离感知下采样：如[82]中所述，KITTI 数据集中的单点云扫描通常包含数万个点。构造具有如此大量点的图在计算上是过高的。因此，该算法引入了距离感知

的下采样方案来对点  $P$  进行下采样，而不会丢失原始点云扫描中的相关信息。简单的体素降采样使用常规的体素网格从输入点云创建统一降采样的点云。位于扫描中心附近的对象结构密集，而远离中心的对象定义不清。如图 2-11 所示，在一次扫描中没有很好地定义一个远离自我运载工具的物体。使用可变体素大小，具体取决于对象从原点开始的位置。远离原点的点使用较小的体素大小，因此降采样的点云不会丢失原始点云扫描的几何信息，因为较小的体素大小倾向于比较大的体素减少采样点的数量。如图 2-11 所示，即使通过增加体素大小，远距离物体的质量也不会因感知距离而下降。

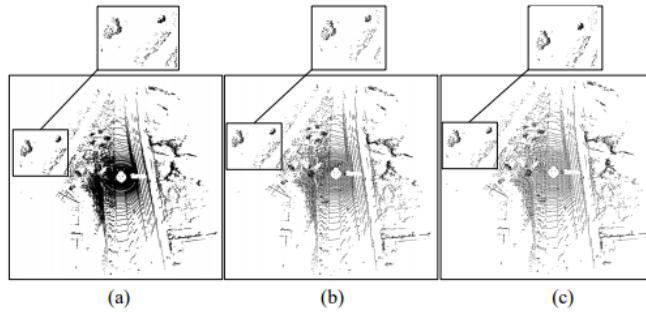


图 2-11 包围盒合并和计分的 NMS 算法

降采样的点云  $P_D$  可以用来从该降采样的点云  $P_D$  构造一个  $k$  最近邻图  $G = \{V, E\}$ ，其中  $V = \{p_1, p_2, p_3, \dots, p_N\}$ ，并且  $E$  由在固定半径内的点  $P_i$  与其相邻顶点之间构成的边组成。

**初始顶点状态：**与[47]相似，该算法使用简化的 PointNet 层嵌入反射强度值和相对坐标。对于每个点，先应用线性层，然后再使用 BatchNorm 和 ReLU，以嵌入反射强度和相对坐标值。该层将逐点特征与局部聚集的特征相结合，允许学习用于表征局部 3D 形状信息的复杂特征。生成的特征用作顶点的初始状态值。然后，将此构造的图传递到 GNN 进行进一步处理。

## (2) 基于注意力的特征聚合

令  $S = s_1, s_2, s_3, \dots, s_N \in \Re^F$  是一组输入特征，在  $k-1$  次迭代时与顶点  $u \in V$  相关。图神经网络(GNN)的单次迭代聚合了给定节点  $u$  的邻域  $N(u)$  中  $k$  个节点的特征，因此在  $k+1$  次迭代中，顶点  $u$  的更新特征  $s'$  由下式给出：

$$s_u^{k+1} = \sigma(W_k(e_{uv}^k, v \in N(u)), b_k s_u^k) \quad (2-22)$$

其中， $W_k$  和  $b_k$  是可训练的权重和偏差矩阵，而  $\sigma$  是引入非线性的激活函数(例如 ReLU)。函数  $e_{uv}^k$  沿边缘聚合特征。此函数更新特征并在每次迭代中重复该过程。

从文献[57]中得到启发，该算法提出了一种基于注意力的聚集方法，以使用权重来细化邻域顶点状态。所提出的方法可以处理无序的点云集和大小波动的邻居关系。令  $\alpha$  为节点  $v$  对节点  $u$  的加权因子（重要性），它计算  $V$  和  $v \in N(u)$  中所有  $u$  的注意力系数  $a_{uv}$ 。在标准 GNN 中， $\alpha = 1/|N(v)|$ 。将  $a_{uv}$  定义为注意机制  $a$  的副产品，该注意机制根据节点对  $u, v$  的消息来计算注意系数  $e_{uv}$ ：

$$e_{uv} = a(W_k s_u^{k-1}, W_k s_v^{k-1}), v \in N(u) \quad (2-23)$$

为了捕获对象的局部结构并使边缘的权重动态适应相似的邻居，将  $e_{uv}$  定义为：

$$e_{uv} = a(\delta x_{uv}, \delta s_{uv}), v \in N(u) \quad (2-24)$$

其中  $\delta x_{uv} = x_v - x_u$ ，表示顶点相对坐标之间的差异。 $\delta s_{uv} = M(s_v) - M(s_u)$  其中  $M$  是特征映射函数，即多层神经网络。顶点之间的相对坐标差学习了  $u$  与邻居  $v$  之间的空间关系。顶点对之间的特征差为相似的邻居分配了更多的权重。这两个术语是使用多层神经网络连接并实现的，因此：

$$e_{uv} = MLP(\delta x_{uv} \parallel \delta s_{uv}) \quad (2-25)$$

在处理了来自  $u$  邻域的不同大小的顶点后，我们使用 softmax 函数对  $a_{uv}$  系数进行归一化，以比较不同邻居之间顶点的重要性，并计算  $\alpha_{uv}$ ：

$$\alpha_{uv} = \frac{\exp(e_{uv}^t)}{\sum_{v \in N(u)} \exp(e_{uv}^t)} \quad (2-26)$$

其中  $a_{uv}$  是第  $k$  次迭代中顶点  $v$  对顶点  $u$  的注意权重。因此，将 GNN 的一个迭代公式表示为：

$$s_u^k = (\sum_{v \in N(u)} \alpha_{uv} * W_k s_v^{k-1}) + s_u^{k-1} \quad (2-27)$$

其中 \* 表示逐像素相乘。使用此最终顶点功能来预测对象的类和定向边界框。

### 2.3 基于不同数据融合的 3D 目标检测算法

在前面两小节中分别介绍了仅使用 RGB 图像数据和 LIDAR 数据的各类典型算法，每一种算法都充分挖掘了所使用数据的数据特点。不难看出，RGB 图像数据具有丰富的语义信息，而 LIDAR 点云数据具有丰富空间信息，因此将二者进行结合让来自不同数据的信息进行融合互补也是一个比较容易想到的方向，包括

前面介绍的 VoxelNet 的作者，在最开始提出 VoxelNet 的网络结构后，于两年后又提出名为 MVX-Net<sup>[58]</sup>的网络，本质就是利用 RGB 图像数据对原始结构进行改进，并取得了不错结果。因此，考虑到 LIDAR 激光扫描仪具有准确的深度信息的优势，而相机则保留了更详细的语义信息。本小节将分别介绍两种不同数据融合的 3D 目标检测任务的典型算法。

### 2.3.1 LIDAR 点云与高精度地图融合

最新的基于多数据融合的方法中，很多都是仅基于 LiDAR 传感器与相机数据的融合。然而，尚未很好地利用地图（例如，高清晰度地图）作为智能车辆的基本基础设施来增强物体检测任务。因此有学者提出了一个简单但有效的框架 MapFusion<sup>[64]</sup>，将地图信息集成到现代 3D 目标检测器网络中。特别地，该网络设计了用于 HD Map 特征提取和融合的 FeatureAgg 模块，以及 MapSeg 模块作为检测主干的辅助分割头。该网络提出的 MapFusion 独立于检测器，可以轻松集成到不同的检测器中。

MapFusion 框架的概述如图 2-12 所示，可以大致分为两部分：标准 3D 目标检测模块和地图特征提取模块。在顶部的红色虚线框中描述了基于 LiDAR 的标准 3d 目标检测流程。输入的 LiDAR 点云被发送到 3d 特征提取器（例如 3D 稀疏卷积）并输出体素的特征。地图特征提取块表示为蓝色虚线框，该框将 HDMap 用作输入。在 2D 要素提取器之后，将提取具有相同体素要素大小的地图要素。然后，对每个体素进行串联操作，将 3D 点云要素和地图信息聚合在一起。然后，包括区域提议，框回归和类别分类的检测头遵循融合的特征。另外，添加了辅助分割头，即 MapSeg，以进一步提高特征提取能力。MapFusion 是一个端到端框架，只需稍作修改即可轻松集成到任何标准 3d 目标检测网络中。

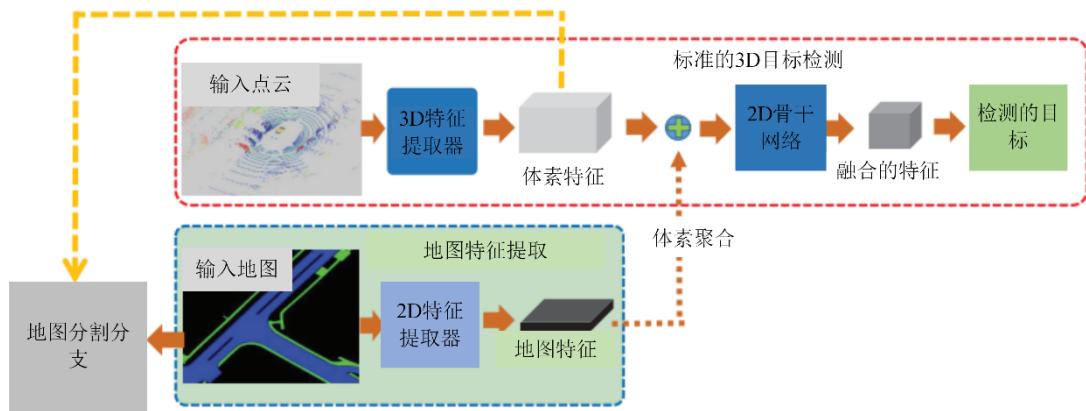


图 2-12 MapFusion 的网络结构图

### (1) HDMap 表示

HDMap 包含有关道路元素的丰富信息，例如可驾驶区域，步行区域和车道。我们通过在图像中心使用 ego car 渲染语义元素来使用栅格表示。对于目标检测任务，这里仅选择三种元素，分别是“可行驶区域”，“人行道”和“停车场区域”。

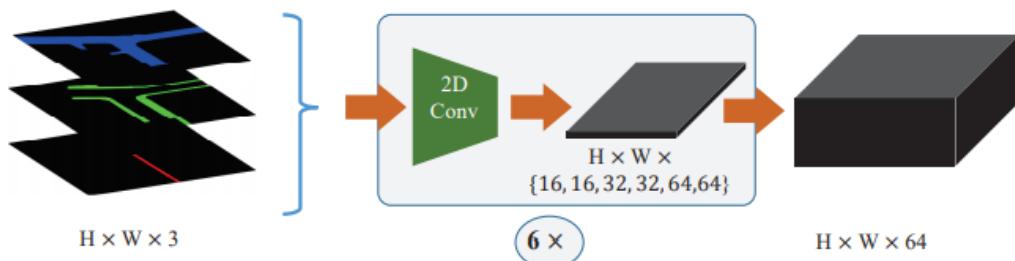


图 2-13 2D 特征提取器结构图

代替直接使用三个栅格图像进行融合，该算法利用 2D Feature Extractor 模块从三个栅格图像中提取高级特征。2D 特征提取器的结构如图 2-13 所示，它是六个相似层的堆栈，包括一个具有  $3 \times 3$  内核的 2D 卷积，批处理归一化和 ReLU 激活功能。六层的过滤器编号分别为 16、16、32、32、64 和 64。具体来说，该算法在 2D Feature Extractor 块之前和之后保持图像大小不变。

### (2) FeatureAgg 模块

FeatureAgg 模块旨在融合提取的地图特征和体素特征。为简单起见，将体素要素和地图要素的大小保持相同，并沿要素通道连接两个张量。尽管操作非常简单，但可以提供令人满意的融合效果。此外，该算法发现，如果在将级联特征发送到下一个检测头之前添加  $1 \times 1$  卷积运算，则可以进一步提高性能。

### (3) MapSeg 模块

MapSeg 模块是一个辅助分割头，它以体素特征作为输入并输出地图分割预测。该预测由真实地图图像监督。该模块的目的是直接从输入点云中学习道路结构信息。实际上，这种信息本质上是可学习的，因为可派生区域与非驾驶区域相比，大多具有独特的结构（例如，平坦的区域）。

## 2.3.2 LIDAR 点云与 RGB 图像融合

为了确定到物体的距离，像 LiDAR 之类的昂贵技术可以提供精确准确的深度信息，因此大多数研究倾向于将重点放在这种传感器上，从而显示出基于 LiDAR 的方法与基于相机的方法之间的性能差距。尽管许多作者已经研究了如何将 RGB 图像与 LiDAR 融合，但还很少有关于在 3D 目标检测任务的深度神经网络中融合 LiDAR 和双目 RGB 图像的研究。因此，有学者提出了 SLS-Fusion<sup>[59]</sup>，这是一种通

过神经网络融合来自 4 光束 LiDAR 和双目相机的数据以进行深度估计的新方法，以实现更好的密集深度图，从而提高 3D 对象检测性能。由于 4 束 LiDAR 比众所周知的 64 束 LiDAR 便宜，因此该方法也被归类为基于传感器的低成本方法。

该算法目标是通过使用双目 RGB 图像和 4 光束 LiDAR 来检测和定位目标的 3D 边界框。给定左右稀疏深度图，通过使用校准参数将稀疏 LiDAR 点云投影在左右图像平面上生成的左右稀疏深度图  $S_l$ ,  $S_r$  和左右 RGB 图像  $I_l$ ,  $I_r$ ，稀疏激光雷达和双目融合网络估计密集整个图像的深度图 D。SLS-Fusion 网络的总体结构如图 2-14 所示。

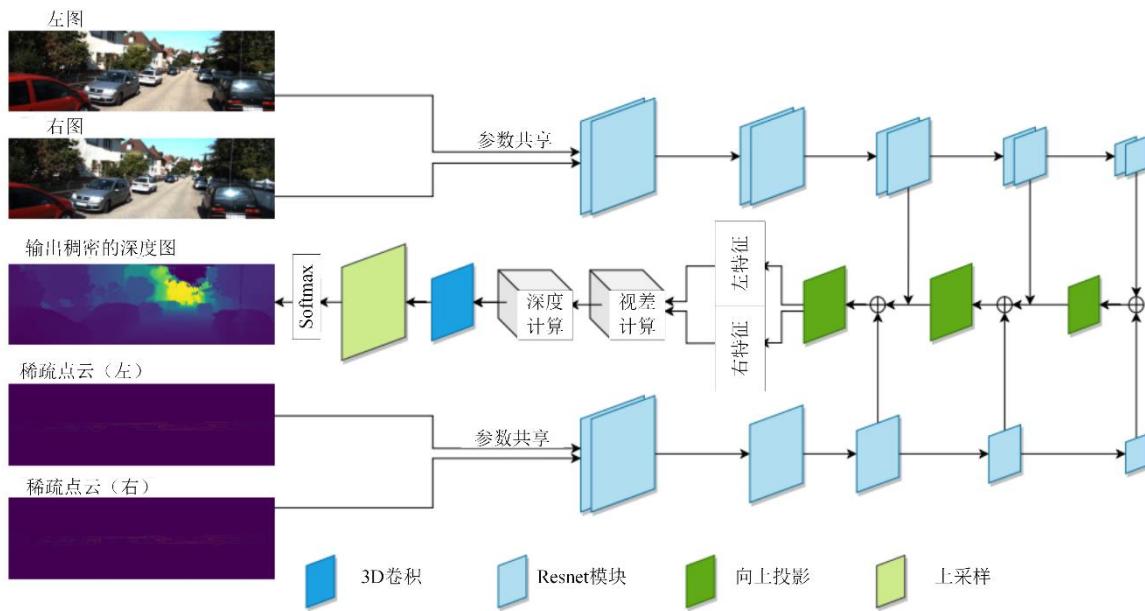


图 2-14 SLS-Fusion 网络结构图

### (1) 网络设计动机

Wang 等人<sup>[48]</sup>提出了伪 LiDAR，它通过更改输入的性质来改善基于 RGB/D 图像的 3D 目标检测方法的性能。You 等人<sup>[49]</sup>创建了他们自己的深度估计网络，以提高深度图的准确性，从而提高 3D 对象检测性能。该算法认为，添加 LiDAR 可以提高模型的性能，该 LiDAR 可以使网络具有强大而准确的深度信息，从而具有更多功能。集成了廉价的 4 光束 LiDAR 系统，而不是使用众所周知但价格昂贵的 64 光束 LiDAR。该假设导致了 SLS-Fusion 网络的设计。

### (2) 网络输入

为了丰富常规双目 RGB 匹配网络的表示形式，已决定加入来自 LiDAR 点云的几何信息。然而，与其像文献[61]中那样，不直接使用来自 LiDAR 的 3D 点云，而是使用校准参数将 4 束 LiDAR 点云重新投影到左右图像坐标，以获得两个稀疏

对应于双目图像的 4 束 LiDAR 深度图。与文献[61]通过将双目图像及其对应的稀疏 LiDAR 深度图集中起来而使用简单的早期融合范例不同，所提出的方法使用了后期融合方法。

### (3) SLS-Fusion 网络

该模型可分为两部分：特征提取和深度成本量。

**特征提取：**Qiu 等人<sup>[62]</sup>提出了一种深度融合单元，它是一种采用后期融合策略的编码器-解码器网络，其中在解码阶段将来自 RGB 图像和 LiDAR 点云的提取特征进行组合。受这项工作的启发，这里提出了一种网络架构，该网络架构在后期融合策略中将 LiDAR 和图像结合在一起。为了进一步利用双目图像输入，如[49,63]中所述，权重共享管道用于 LiDAR 和图像( $I_l, S_l$ )和( $I_r, S_r$ )，而不是左右图像。编码器由一系列 ResNet 块组成，其后是步幅为 2 的卷积层，以 1/16 的输入比例缩小特征图的大小，以获取图像中较小对象的更详细的特征。为了放大特征图并集成来自 LiDAR 和图像的两个编码器的特征，解码器使用了向上投影层<sup>[65]</sup>。但是，与[62]不同，仅使用 3 个向上投影层来获得张量，以在特征分辨率（输入的 1/4）与特征通道数之间取得平衡，然后将其放入深度成本量中，这在计算上是昂贵的。

**深度成本量：**从解码阶段获得的左右特征被传递到 You 等人提出的深度成本量（DeCth）<sup>[49]</sup>学习深度。实际上，这两个特征被馈送到视差成本量（DiCV）<sup>[63]</sup>以形成 4D 成本张量，然后将其转换为 DeCV 以直接优化距离损失而不是视差损失。

## 2.4 本章小结

本章主要是从在 3D 目标检测中所能利用的数据出发，从仅使用 RGB 图像数据到仅使用 LIDAR 点云数据，再到这两种数据结合的这三个角度对主流算法网络进行分类介绍。对于仅使用 RGB 图像数据，介绍了基于单目和双目 RGB 图像数据的两类算法的网络结构；对仅使用 LIDAR 点云数据，从处理这种不规则的点云数据的方法入手，介绍了四种处理方法的算法网络结构，分别为体素化处理、基于 PointNet 处理、体素化与 PointNet 融合处理及使用图神经网络处理；对两种数据融合的算法，分别介绍了点云投影和两阶段处理这两种算法思路。对于每一类算法结构，均挑选了公开发表中具有代表性的网络结构对其进行详细介绍与分析。

### 第三章 基于双目 RGB 和 LIDAR 点云融合的 3D 目标检测方法

3D 目标检测已成为自动驾驶场景中的新兴任务，前面介绍的算法使用基于投影的模型或基于体素的模型处理 3D 点云。但是，这两种方法在都存在一些不足之处。基于体素的方法缺少语义信息，而基于投影的方法投影到不同的视图时会遭受大量空间信息的损失。与上述方法不同，我们观察到双目相机可以从两个视图提供大范围的感知，而 LIDAR 传感器可以捕获准确的 3D 结构，而它们的组合可以利用它们各自的优势，同时弥补其缺点。换句话说，左图像和右图像可以提供更准确的感受野，同时获得可比的深度和位置精度。此外，我们发现最常用的 PointNet [32][33]无法捕获可变比例的局部特征信息，并导致局部特征的丢失，因为它仅独立处理 3D 点以保持排列不变性，这样，它忽略了点之间的距离度量。尽管后来的 SAWnet<sup>[69]</sup>使用共享的多层感知器（MLP）与 DGCNN<sup>[56]</sup>的动态位置信息集成了全局特征，但它无法专注于重要特征并抑制其不必要的特征残余连接<sup>[70]</sup>。

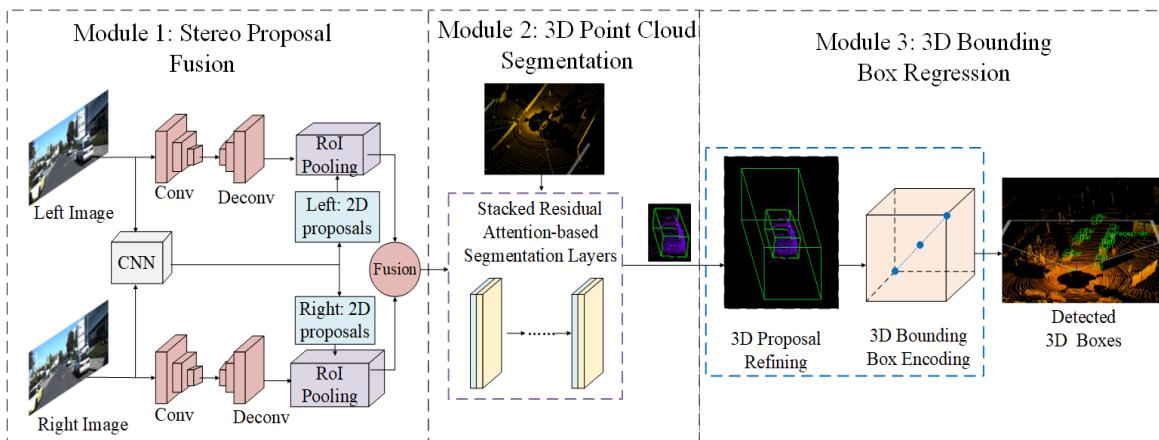


图 3-1 SRDL 的网络结构图

基于这些观察，本章提出了用于 3D 目标检测的 Stereo RGB 和 Deeper LIDAR (SRDL) 网络，从而可以自然地提高 3D 目标检测网络的性能，该网络以双目 RGB 图像和 LIDAR 点云为输入，可以同时利用语义和空间信息并利用注意力机制实现了稳健而准确的 3D 检测，如图 3-1 所示。具体来说，左视图和右视图可以生成从不同角度看并不完全重叠的提议。它们可以相互纠正，并且可以在融合阶段生成更精确的区域。考虑到融合的建议可能会重叠噪声和对象的多余空间，因此设计了 3D 点云中面向特征的分割网络，以从背景中分割出目标点云。给定分割的目标点和裁剪的建议框，本章提出通过以新颖的紧凑方式添加更多约束来对边界框进行

编码。此设计的好处是可以消除更多冗余并更精确地定位对象的大小，同时减小特征尺寸。在 2012 年就公布的 KITTI 检测数据集的实验结果证明了同时利用双目图像和点云进行 3D 目标检测的有效性。

### 3.1 双目 2D 建议框融合

相对于点云来说，RGB 图像具有更多的语义信息。本章将双目图像用作输入，因为双目视觉比单目视觉具有多个优势。首先，双目摄像机可以提供深度信息，这对于遮挡的场景非常有用。其次，从感受野的范围来说，双目图像是更大的。因此，在本章的框架中，将双目图像作为输入，并利用成熟的 2D 目标检测器分别为左右图像生成 2D 目标建议框。同时，在每个视图中应用卷积反卷积以获取更高分辨率的特征。结合 2D 提案框，RoIPooling 用于每个视图以获取相同大小的要素。最后，受[52]的启发，通过逐元素均值运算将 RoIPooling 中输出的两个裁剪特征融合在一起。



图 3-2 左右视图的建议框。左视图和右视图中的建议框没有完全重叠，并且最终融合的建议框比其中任何一个更为准确。

如图 3-2 所示，左右两个分支的输出没有完全重叠。相反，左右图可以产生具有不同的视角的建议框。利用已知的来自 RGB-D 数据的相机投影矩阵来提供准确的深度信息，可以将每个边界框投影到 3D 空间中以形成交叉物体区域。通过最终的逐元素融合，最终建议框包含较少的空间和点云，通过相互监督和更正，该空间和点云比任何一个初始云都更准确。

### 3.2 3D 点云分割

在图 3-1 中，融合的带有深度信息建议框被送入第二阶段，确定了 3D 空间中的大致位置。对于给定的 2D 图像区域及其对应的 3D 位置，本小节为了最终的 3D 坐标回归设计了 3D 分割网络，其主要作用是从背景中分离 3D 点云。

### 3.2.1 分割网络概述

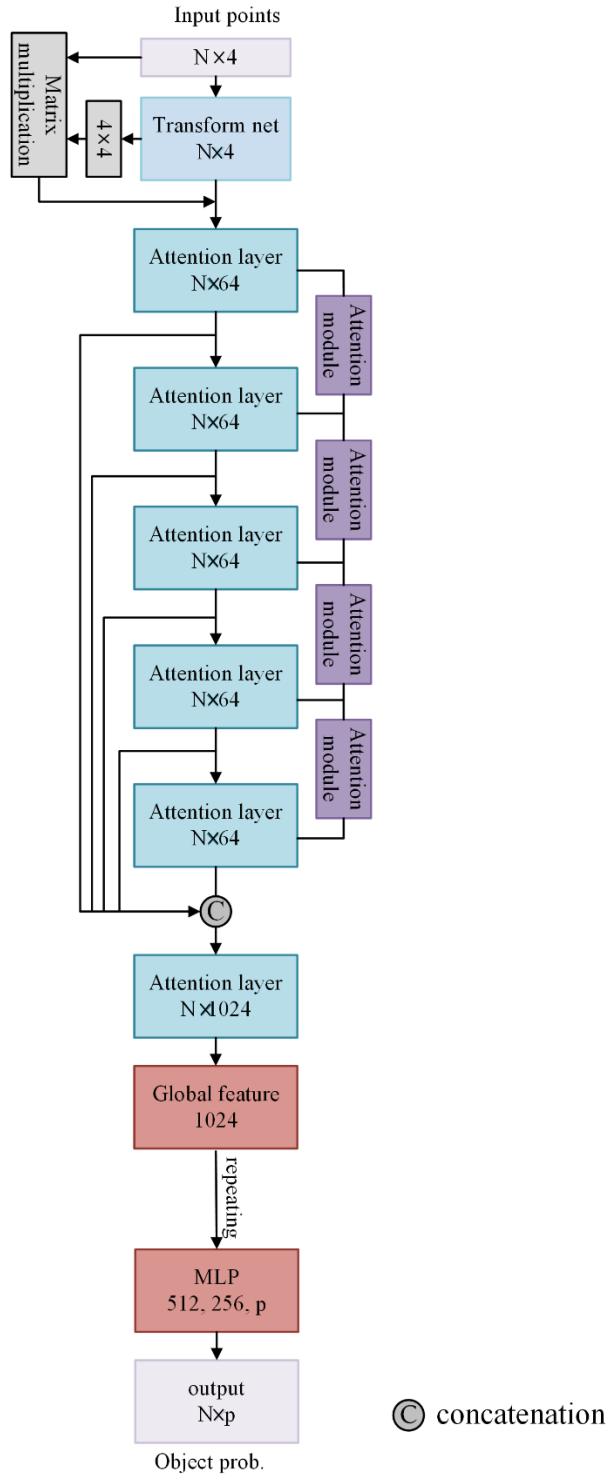


图 3-3 点云分割网络结构

分割网络的功能类似于 F-pointnet<sup>[20]</sup>中提出的功能，它以预搜索范围内的点云作为输入，并给出表示该点属于感兴趣对象概率的分割分数。但是区别在于：(a)

本章使用变换矩阵而不是坐标旋转来保持算法的旋转不变性。（b）本章将 DGCNN<sup>[56]</sup>中的 EdgeConv 与共享 MLP 连接起来，以一种基于残差注意力的方式更好地捕获点云的全局和局部几何特征。这样，可以融合不同级别的要素，并且将通过几个堆叠的基于注意力的层生成更深的点云要素。

分割网络的具体结构如图 3-3 所示。输入被馈送到使用基于注意力层的转换网络以回归  $4 \times 4$  变换矩阵，该矩阵的元素是用于点云对齐的学习仿射变换值。然后将对齐的点送入几个堆叠的基于注意力的层中，以生成这些点的置换不变嵌入。其中，残差注意力模块充当两个相邻层之间的链接桥以传递信息。之后，将之前所有基于 1024 维注意力层的输出串联在一起，并使用最大池来获取点云的最终全局信息聚合。然后将信息送入 MLP 层以预测  $N \times p$  得分矩阵并进行逐点预测。

### 3.2.2 基于注意力机制的特征融合

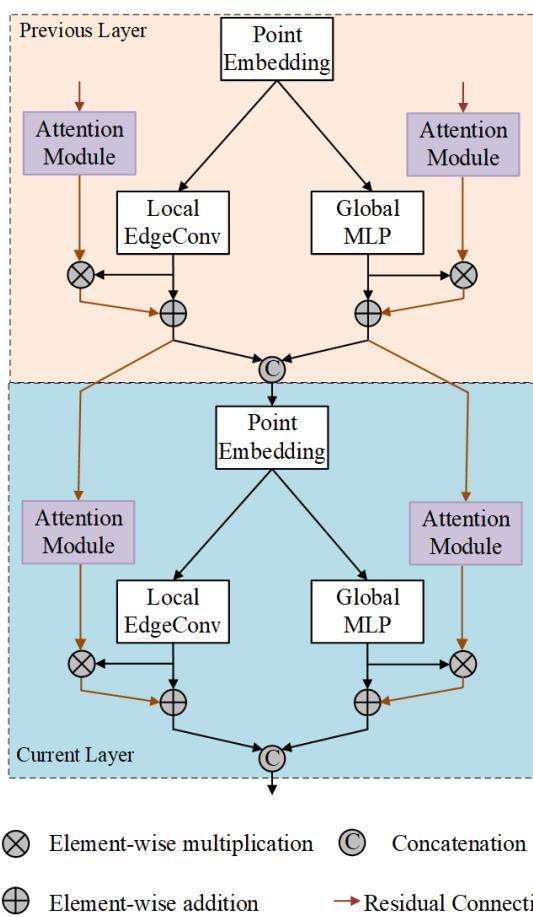


图 3-4 信息在两个相邻层之间传播结构图

基于注意力层的体系结构如图 3-4 所示，其中当前层被视为中间层，其特征不仅通过主流信息流传输，还通过前一层的残差连接。从当前层嵌入的点被输入到两

个并行层中，分别是局部 EdgeConv 层和全局 MLP 层。局部 EdgeConv 层构造一个动态图，并合并  $k$  个最近的局部邻域信息。全局 MLP 层在每个点上独立运行，然后应用对称函数来累积特征。在连接在一起之前，这两层的输出以元素方式连接到上一层的同一分支的输出。两层还使用残差注意力连接将信息分别传输到下一个嵌入层。此外，两层中的每一层分别在其内依次包含两个共享的 MLP。第一个共享的 MLP 通过批处理规范化和 ReLU 计算来估计嵌入，而第二个共享的 MLP 仅具有批处理规范化。

具体来说，考虑  $D$  维嵌入点云中的  $n$  个点，其集合  $P = \{p_1, p_2, \dots, p_n\}$ ，其中  $D$  可以简单地设置为 3，这意味着每个点  $x_n$  包含三个坐标  $(x_i, y_i, z_i)$ 。这些点由每个基于注意力层中的局部 EdgeConv 层和全局 MLP 层并行处理。在局部 EdgeConv 层的分支中， $h(k)$  表示输入，以动态图中的  $k$  个最近邻居表示，而  $e_i (i=1,2)$  表示 Edgeconv 运算，以评估点与其  $k$  个最近邻居依赖性。提取的边缘特征被输入到批处理归一化和 ReLU 计算中。输出可以表示为：

$$E_1 = \text{MLP}(h(k)) = \text{MLP}(e_1(h(k))) = \sigma(W_1(h(k))) \quad (3-1)$$

在应用另一个 Edgeconv-BN 层之后，输出可以表示为：

$$E_2 = \text{MLP}(E_1) = \text{MLP}(e_2(E_1)) = W_2(E_1) \quad (3-2)$$

这里  $\sigma$  代表了 ReLU 激活函数， $W_1$ 、 $W_2$  是两个 MLP 层的权值。在对输出进行最大池化之后，残差模块的意图以逐点方式添加到输出  $E_2$  中，表示为：

$$L = (1 + R_1) \otimes E_2 \quad (3-3)$$

类似地，在全局 MLP 层中， $f(t)$  表示由共享加权 MLP（表示为  $s$ ）对输入点的变换。第一个共享 MLP 层的输出为：

$$M_1 = \text{MLP}(f(t)) = \sigma(W_1(s(k))) \quad (3-4)$$

在应用另一个 MLP-BN 层后，输出为：

$$M_2 = \text{MLP}(M_1) = W_2(s(M_1)) \quad (3-5)$$

此输出以相同方式连接到残差注意力模块中的注意力感知特征：

$$G = (1 + R_2) \otimes M_2 \quad (3-6)$$

其中 $\otimes$ 表示逐元素乘法，并且 $R_i(i \in \{1, 2\})$ 对不同特征的反应在 0 和 1 之间变化。与原始 ResNet 不同，残差注意模块 $R_i(i \in \{1, 2\})$ 的输出用作特征过滤器，以减弱嘈杂的特征并放大良好的特征。注意，来自两个分支的输出 $L, G$ 具有相同的尺寸，并且也被传输到下一层。最后，将它们连接在一起，并将嵌入点作为输入传输到下一层。

### 3.2.3 注意力模块

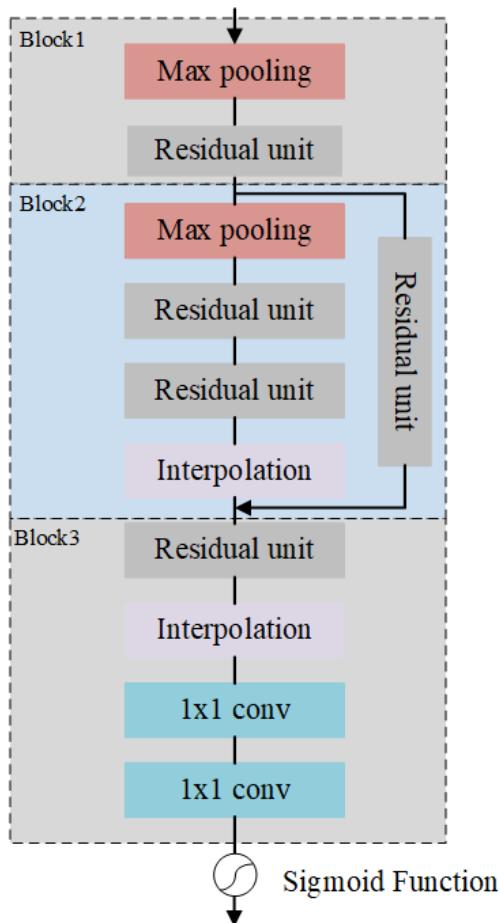


图 3-5 注意力模块的结构，主要由自下而上和自上而下的残差单元组成。

注意力模块不仅尝试强调有意义的特征，而且还增强了目标在某些位置的不同表示形式。本章将注意力模块设计为自下而上和自上而下的结构，如图 3-5 所示。自下而上的操作旨在收集全局信息，而自上而下的操作则将全局信息与原始特征图结合起来。并且使用[71]中的残差单位作为注意力模块中的基本单位。注意模块包含三个块。在 block 1 中，执行最大值池化和残差单元以扩大感受野。在获得最低分辨率后，将采用对称的自上而下的体系结构来推断每个像素以获取 block 2 中的密集特征。此外，在自下而上和自上而下的特征图之间附加了跳过连接，以捕

获不同比例的特征。在 block 3 中，在残差单位之后插入双线性插值以对输出进行上采样。最后，使用 Sigmoid 函数对两个连续的  $1 \times 1$  卷积层进行归一化以平衡尺寸。由于注意力模块精心的设计是轻量的，因此尽管将其多次插入细分网络中，但参数和计算的总体大小几乎可以忽略不计。

### 3.3 3D 检测框回归

给定分割的目标点，这部分在提案框优化后通过更精确的边界框编码方案回归最终的 3D 边界框。

#### 3.3.1 3D 建议框优化

在点云上进行分割操作后，可以从背景中分离出对象点，并在第一个模块的特定位置获取边界框内的点。然而，来自第一模块的预定义建议框和针对这些点的分割网络的组合仅得到相对粗糙的框。因此，本小节提出合并 3D 点及其相应特征以重新调整建议框。对于每个 3D 建议框  $b_i = (x_i, y_i, z_i, w_i, h_i, l_i, \theta_i)$ ，通过向  $w_i, h_i, l_i$  分别添加常数  $\xi$  来定义新的 3D 建议框，以调整建议框的大小。对于每个点，都会执行验证测试来确定它是否在调整大小的框内。如果为真，则将保留该点及其特征，以完善建议框方案。进一步的消融实验将说明该操作在改善性能方面的有效性。

#### 3.3.2 3D 检测框编码

为了确定 3D 边框的方向，通过计算  $(\cos \theta, \sin \theta)$  来解决角度的问题。至于框的编码，目前有几种不同的方法来对边界框进行编码，如图 3-6 所示。在[72]中首先提出了对齐的轴，它用中心和大小对框进行编码。在 MV3D 中<sup>[52]</sup>，Chen 等人声称 8 个角框编码比轴对齐更好。在 AVOD<sup>[60]</sup>中，Jason Ku 等人尝试将 4 个角和 2 个高度替换为 8 个角，以对框进行有效编码。但是，有 8 个角需要 24 维向量来标准化提案框的对角线长度，而忽略了物理约束。4 个角和高度编码方法没有考虑平面内 4 个角之间的物理连接。为了减少更多的冗余并保持物理连接，本小节提出使用三个点（两个角+一个中心）和两个高度对边界框进行编码，两个高度代表从建议框到地平面的偏移量。这三个点在立方体的对角线上，其中  $c_2$  是立方体的中心点。因此，回归目标为  $\{\Delta x_i, \Delta y_i, \Delta z_i, \Delta h_i; i=1, 2, 3; j=1, 2\}$ 。尽管这种 11 维表示矢量略大于 10 维表示矢量，但这种编码方法不仅使用了更少的点，而且还在这些坐标参数之间的流动约束中紧凑地对边界框进行了编码。在回归  $w, l$  时，应考虑的约束为：

$$\begin{aligned} (\Delta x_{c1}, \Delta x_{c3}) &\rightarrow w, (\lvert \Delta x_{c2} - \Delta x_{c1} \rvert, \lvert \Delta x_{c3} - \Delta x_{c2} \rvert) \rightarrow w. \\ (\Delta y_{c1}, \Delta y_{c3}) &\rightarrow l, (\lvert \Delta y_{c2} - \Delta y_{c1} \rvert, \lvert \Delta y_{c3} - \Delta y_{c2} \rvert) \rightarrow l. \end{aligned} \quad (3-7)$$

当回归  $h$  时，应确保以下等式成立：

$$\begin{aligned} \lvert \Delta z_{c3} - \Delta z_{c1} \rvert &= \lvert \Delta h_2 - \Delta h_1 \rvert \rightarrow h \\ \lvert \Delta z_{c2} - \Delta z_{c1} \rvert &= \lvert \Delta z_{c3} - \Delta z_{c2} \rvert = \frac{1}{2} \lvert \Delta h_2 - \Delta h_1 \rvert \rightarrow h \end{aligned} \quad (3-8)$$

这里  $\rightarrow$  表示在回归时存在约束。但是，应注意，这里的 3 个点 + 2 个高度编码方法是通过线性坐标变换获得的。这些约束集是坐标之间的几何约束的派生，因此不会生成新的约束参数。

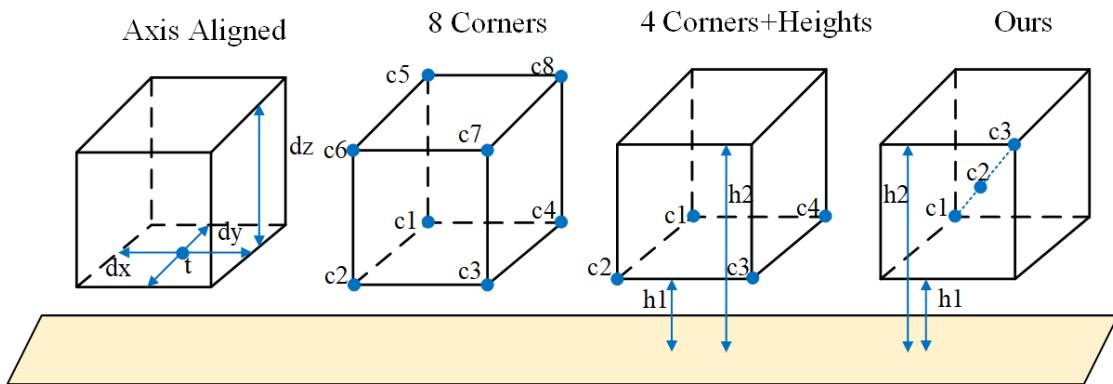


图 3-6 不同边框编码方案的对比

### 3.4 损失函数

本章使用多任务损失函数来训练网络。总损失由三个模块中的三个主要成分组成，即融合损失  $L_{fuse}$ ，分段损失  $L_{seg}$  和边界框回归损失  $L_{box}$ ，总损失表示为：

$$L_{total} = \alpha L_{fuse} + \beta L_{seg} + \chi L_{box} = \alpha(L_{cls\_1} + L_{reg\_1}) + \beta L_{seg} + \chi(L_{cls\_2} + L_{reg\_2}) \quad (3-9)$$

这里  $\alpha$ 、 $\beta$ 、 $\chi$  是用来平衡不同部分之间相对重要性的参数，在实验中被分别设置为 1、4、2。 $L_{cls\_1}$  和  $L_{cls\_2}$  是目标分类损失， $L_{reg\_1}$  和  $L_{reg\_2}$  是边框回归损失。

在本章的实验中，对所有分类损失应用二进制交叉熵，公式为：

$$\begin{aligned} L_{cls} &= \frac{1}{N_{pos}} \sum_i L_{cls}(p_i^{pos}, 1) + \vartheta \frac{1}{N_{neg}} \sum_i L_{cls}(p_i^{neg}, 0) \\ L_{cls}(p, t) &= -(t \log(p) + (1-t) \log(1-p)) \end{aligned} \quad (3-10)$$

在实验中将  $\theta$  设置为 8。至于回归，对所有边界框和方向向量回归 Smooth L1 损失：

$$L_{reg}(x) = \begin{cases} 0.5(\sigma x)^2, & \text{if } |x| < \frac{1}{\sigma^2} \\ |x| - \frac{0.5}{\sigma^2}, & \text{otherwise} \end{cases} \quad (3-11)$$

在实验中  $\sigma$  设置为 4。

对于分割，由于目标点的数量通常比交叉区域的背景点的数量大得多，因此使用 focal loss<sup>[73]</sup> 处理不平衡问题，如下所示：

$$\begin{aligned} L_{seg}(p_t) &= -\alpha_t(1-p_t)^\gamma \log(p_t) \\ p_t &= \begin{cases} p, & \text{object} \\ 1-p, & \text{otherwise} \end{cases} \end{aligned} \quad (3-12)$$

其中我们将参数  $\alpha_t = 0.25$  和  $\gamma = 2$  保留为原始论文一致。

## 3.5 实验及分析

### 3.5.1 实验数据集以及评价指标

本章实验在广泛使用的 KITTI 3D 目标检测基准上评估提出的方法，该基准由 LIDAR 点云和 RGB 图像组成。该数据集包括 7481 个训练图像/点云和 7518 个测试图像/点云，检测的目标分为 car, pedestrian 和 cyclist。对于每个检测的目标类别，将根据三个难度级别评估检测结果：简单，中等和困难。同文献[74]，本章将训练数据按约 1:1 的比例分为训练集（3712 个图像和点云）和一个验证集（3769 个图像和点云），后续的消融实验是在此验证集上完成的。。在训练集上训练我们的模型，并将 SRDL 在验证和测试集上的与 3D 目标检测的最新方法进行比较。

为了进行评估，使用平均精度（AP）度量标准与不同方法进行比较，并针对汽车，骑车人和行人类别分别使用 0.7、0.5 和 0.5 的官方 3D IoU 评估度量。

### 3.5.2 实验细节

#### 3.5.2.1 网络结构

在 2D 建议框生成阶段，两个分支具有相同的体系结构。对于 2D 目标检测器，实验使用 FPN 生成区域建议，并使用 Fast RCNN 预测最终边界框。对于 FPN 部分，将简化的 VGG 作为基础网络，该网络对一半的通道进行采样。然后，将池化

层替换为卷积核大小为  $3 \times 3$ , 步长为 2 的卷积层。对于 Fast RCNN 部分, 从所有卷积层中提取特征并将所有特征连接起来以进行最终预测。预训练使用的数据集是 COCO, 在 fine-tuning 阶段, 使用的数据集是 KITTI 的 2D 目标检测数据集。为了获得高分辨率的特征图, 对于每个视图, 在 RoIpooling 层之前的缩减 VGG-16 中应用  $2\times$  双线性上采样层。降低的 VGG-16 的通道也被采样了一半, 当融合这些功能时, 在原始网络的末尾添加了一个完全连接层。为了减少多余的检测框, 在两个 2D 框上应用了非最大抑制 (NMS), 且 IoU 阈值设为了 0.7。同文献[39,60], 实验训练了两个网络: 一个用于 car, 另一个用于 pedestrian 和 cyclist。实验使用 0.05 的 IoU 阈值在最后阶段移除多余的 box。

表 3-1 汽车、行人和骑行者 KITTI 3D 目标检测测试集性能比较

方法	模式	$AP_{car}(\%)$			$AP_{pedestrian}(\%)$			$AP_{cyclist}(\%)$		
		简单	中等	困难	简单	中等	困难	简单	中等	困难
M3D-RPN <sup>[78]</sup>	M	15.52	11.44	9.62	-	-	-	-	-	-
CE3R <sup>[35]</sup>	M	21.48	16.08	15.26	-	-	-	-	-	-
Stereo rcnn <sup>[36]</sup>	S	49.23	34.05	28.39	-	-	-	-	-	-
MV3D <sup>[52]</sup>	M+L	71.09	62.35	55.12	-	-	-	-	-	-
F-Pointnet <sup>[20]</sup>	M+L	81.20	70.39	62.19	51.21	44.89	40.23	71.96	56.77	50.39
AVOD-fpn <sup>[60]</sup>	M+L	81.94	71.88	66.38	50.80	42.81	40.88	64.00	52.18	46.61
F-ConvNet <sup>[79]</sup>	M+L	85.88	76.51	68.08	52.37	45.61	41.49	79.58	64.68	57.03
MMF <sup>[80]</sup>	M+L	86.81	76.75	68.41	-	-	-	-	-	-
Voxelnet <sup>[39]</sup>	L	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
SECOND <sup>[46]</sup>	L	83.13	73.66	66.2	51.07	42.56	37.29	70.51	53.85	46.9
PointPillars <sup>[47]</sup>	L	79.05	74.99	68.3	52.08	43.43	41.49	75.78	59.07	52.92
Pointrcnn <sup>[10]</sup>	L	85.94	75.76	68.32	49.43	41.78	38.63	73.93	59.6	53.59
STD <sup>[81]</sup>	L	86.61	77.63	76.06	53.08	44.24	41.97	78.89	62.53	55.77
PV-RCNN <sup>[43]</sup>	L	90.25	81.43	76.82	-	-	-	78.6	63.71	57.65
Point-GNN <sup>[82]</sup>	L	88.33	79.47	72.29	51.92	43.77	40.14	78.6	63.48	57.08
<b>SRDL</b>	<b>S+L</b>	<b>87.73</b>	<b>80.38</b>	<b>76.27</b>	<b>47.30</b>	<b>39.43</b>	<b>36.99</b>	<b>77.35</b>	<b>62.02</b>	<b>55.52</b>

对于分割子网络, 将 6 个基于注意力的层与 4 个连接这些层的注意模块堆叠在一起。转换网络包含 3 个基于注意力的层, 维度分别为 64、128 和 1024。最后

一层的输出与合并的全局特征集成，该全局特征由 MLP 层生成，该 MLP 层具有两个大小分别为 512 和 256 的隐藏层。在局部 EdgeConv 层中，为 car 设置  $k = 30$  最近的邻居，为 pedestrian 和 cyclist 设置  $k = 20$  来构造动态图。

### 3.5.2.2 训练细节

以端到端的方式训练网络。使用的优化器为 ADAM<sup>[75]</sup>，其初始学习率为 0.001，每 60K 迭代以指数形式衰减，衰减率为 0.8。默认情况下，模型在单个 GTX 1080 GPU 上训练，批处理大小为 24，总共 160 个 epochs。

表 3-2 汽车、行人和骑自行车者 KITTI 鸟瞰检测测试集性能比较

方法	模式	$AP_{car}(\%)$			$AP_{pedestrian}(\%)$			$AP_{cyclist}(\%)$		
		简单	中等	困难	简单	中等	困难	简单	中等	困难
M3D-RPN <sup>[78]</sup>	M	21.29	15.23	13.16	-	-	-	-	-	-
Stereo rcnn <sup>[36]</sup>	S	61.27	43.87	36.44	-	-	-	-	-	-
MV3D <sup>[52]</sup>	M+L	86.02	76.9	68.49	-	-	-	-	-	-
F-Pointnet <sup>[20]</sup>	M+L	88.70	84.00	75.33	58.09	50.22	47.2	75.38	61.96	54.68
AVOD-fpn <sup>[60]</sup>	M+L	88.53	83.79	77.9	58.75	51.05	47.54	68.09	57.48	50.77
F-ConvNet <sup>[79]</sup>	M+L	89.69	83.08	74.56	58.9	50.48	46.72	82.59	68.62	60.62
MMF <sup>[80]</sup>	M+L	89.49	87.47	79.1	-	-	-	-	-	-
Voxelnet <sup>[39]</sup>	L	89.35	79.26	77.39	46.13	40.74	38.11	66.7	54.76	50.55
SECOND <sup>[46]</sup>	L	88.07	79.37	77.95	55.1	46.27	44.76	73.67	56.04	48.78
PointPillars <sup>[47]</sup>	L	88.35	86.1	79.83	58.66	50.23	47.19	79.14	62.25	56
Pointrenn <sup>[10]</sup>	L	89.47	85.58	79.10	-	-	-	81.52	66.77	60.78
STD <sup>[81]</sup>	L	89.66	87.76	86.89	60.99	51.39	45.89	81.04	65.32	57.85
Point-GNN <sup>[82]</sup>	L	93.11	89.17	83.90	55.36	47.07	44.61	81.17	67.28	59.67
<b>SRDL</b>	S+L	92.01	88.17	85.43	52.42	44.84	42.56	79.64	64.52	57.90

为了确保输入变量的大小一致，以避免梯度爆炸或分散，使用 Xavier 优化器初始化网络参数。在最后一个完全连接的层中使用 Dropout Selu 函数<sup>[76]</sup>，可以缓解梯度的消失并增加网络的非线性拟合能力。在每个参数层之后都加上了块正则化，参数层以 0.5 的衰减率开始，并在每 20K 迭代中以 0.5 的比率逐渐衰减至 0.99。

### 3.5.2.3 数据增强

为了尽可能减轻模型过拟合的问题，在实验中应用了数据增强。由于本章所提的算法将图像和点云都作为输入，因此扩充包括两个分支。对于 2D 框，将应用运动来独立地扰动每个边界框。首先使边界框绕 Z 轴旋转  $[-\pi/2, \pi/2]$  均匀分布。然后，使用平移  $[\Delta X, \Delta Y]$  移动旋转的边界框，并从均值为零且标准偏差为 1 的高斯分布中独立采样  $[\Delta X, \Delta Y]$ 。对点云的具体的增强方法有：（1）随机打乱点云及其对应的标签，以增强网络的鲁棒性。（2）从均匀分布  $[-\pi/4, \pi/4]$  绕任意轴以多个角度旋转点云模型。（3）通过正态分布将来自  $N(0,1)$  的高斯噪声添加到原始点云数据集中，以提高网络的抗干扰能力和推理能力。（4）在 Z 轴方向上随机扰动模型位置以扩大点的深度。（5）将整个块中的点的坐标乘以  $[0.95, 1.05]$  中的均匀采样比例。受文献[47,77]的启发，还执行了几组 ground-truth 增强，包括随机翻转，整体缩放，围绕垂直轴的整体旋转。此外，还将几个新的 ground-truth 真假框及其他场景对应的点放到当前点云的相同位置，以模拟具有各种环境的目标。

表 3-3 KITTI 汽车 3D 目标检测验证集性能比较

方法	模式	$AP_{car}(\%)$		
		简单	中等	困难
M3D-RPN <sup>[78]</sup>	Mono	20.77	17.06	15.21
CE3R <sup>[35]</sup>	Mono	32.33	21.09	17.26
Stereo rcnn <sup>[36]</sup>	Stereo	54.11	36.69	31.07
MV3D <sup>[52]</sup>	Mono+Lidar	71.29	62.68	56.56
F-Pointnet <sup>[20]</sup>	Mono+Lidar	83.76	70.92	63.65
AVOD-FPN <sup>[60]</sup>	Mono+Lidar	84.41	74.44	68.65
Cont-Fuse <sup>[83]</sup>	Mono+Lidar	86.32	73.25	67.81
F-ConvNet <sup>[79]</sup>	Mono+Lidar	89.02	78.8	77.09
Voxelnet <sup>[39]</sup>	Lidar	81.97	65.46	62.85
SECOND <sup>[46]</sup>	Lidar	87.43	76.48	69.1
PointRCNN <sup>[10]</sup>	Lidar	88.88	78.63	77.38
STD <sup>[81]</sup>	Lidar	89.70	79.80	79.30
<b>SRDL</b>	Stereo+Lidar	90.28	79.82	78.65

### 3.5.3 实验结果

本实验将根据 KITTI 测试集的 3D 检测基准和鸟瞰检测基准评估本章的方法。对于表 3-1 中所示的 3D 目标检测测试基准，“M”表示单目 RGB 图像输入，“S”表示双目 RGB 图像输入，“L”表示激光雷达点云输入。本章提出的方法在所有难度级别上的所有类别中都大大优于其他 Mono + Lidar 方法。对于汽车，与这些基于 LIDAR 的方法相比，由于结合了图像，SRDL 大幅增加，尤其是在中等强度和困难强度条件下。对于表 3-2 中所示的鸟瞰视图测试基准，本章的方法可以与大多数方法取得近似的结果。对于行人来说，它的性能通常比单传感器差，可能的原因是行人的稀疏点云和较小的尺寸，因此从不同视角进行的投影比直接在其上进行操作可以捕获更多的特征。

表 3-4 KITTI 汽车鸟瞰图检测验证集性能比较

方法	模式	$AP_{car}(\%)$		
		简单	中等	困难
M3D-RPN <sup>[78]</sup>	Mono	26.29	21.18	17.90
CE3R <sup>[35]</sup>	Mono	43.75	28.39	23.87
Stereo rcnn <sup>[36]</sup>	Stereo	68.50	48.30	41.47
MV3D <sup>[52]</sup>	Mono+Lidar	86.55	78.10	76.67
F-Pointnet <sup>[20]</sup>	Mono+Lidar	88.16	84.02	76.44
F-ConvNet <sup>[79]</sup>	Mono+Lidar	90.23	88.79	86.84
Voxelnet <sup>[39]</sup>	Lidar	89.60	84.81	78.57
SECOND <sup>[46]</sup>	Lidar	89.96	87.07	79.66
Fast PointRCNN <sup>[77]</sup>	Lidar	90.12	88.10	86.24
PointRCNN <sup>[10]</sup>	Lidar	90.50	88.50	88.10
<b>SRDL</b>	<b>Stereo+Lidar</b>	<b>90.67</b>	<b>88.41</b>	<b>87.11</b>

对于最重要的汽车类别，实验还对比了本章的方法在 KITTI 验证集上的性能，包括 3D 目标检测和鸟瞰图检测，如表 3-3 和表 3-4 所示。可以看到，本章的方法优于大多数先前状态 3D 检测任务的最新方法，特别是在简单和中等的情况下。具体而言，与最有效的 Mono-LIDAR 方法相比，SRDL 在 3D 目标检测和鸟瞰视图检测上的所有三个类别均实现了大幅提高，这表明了本章提出的方法的有效性。

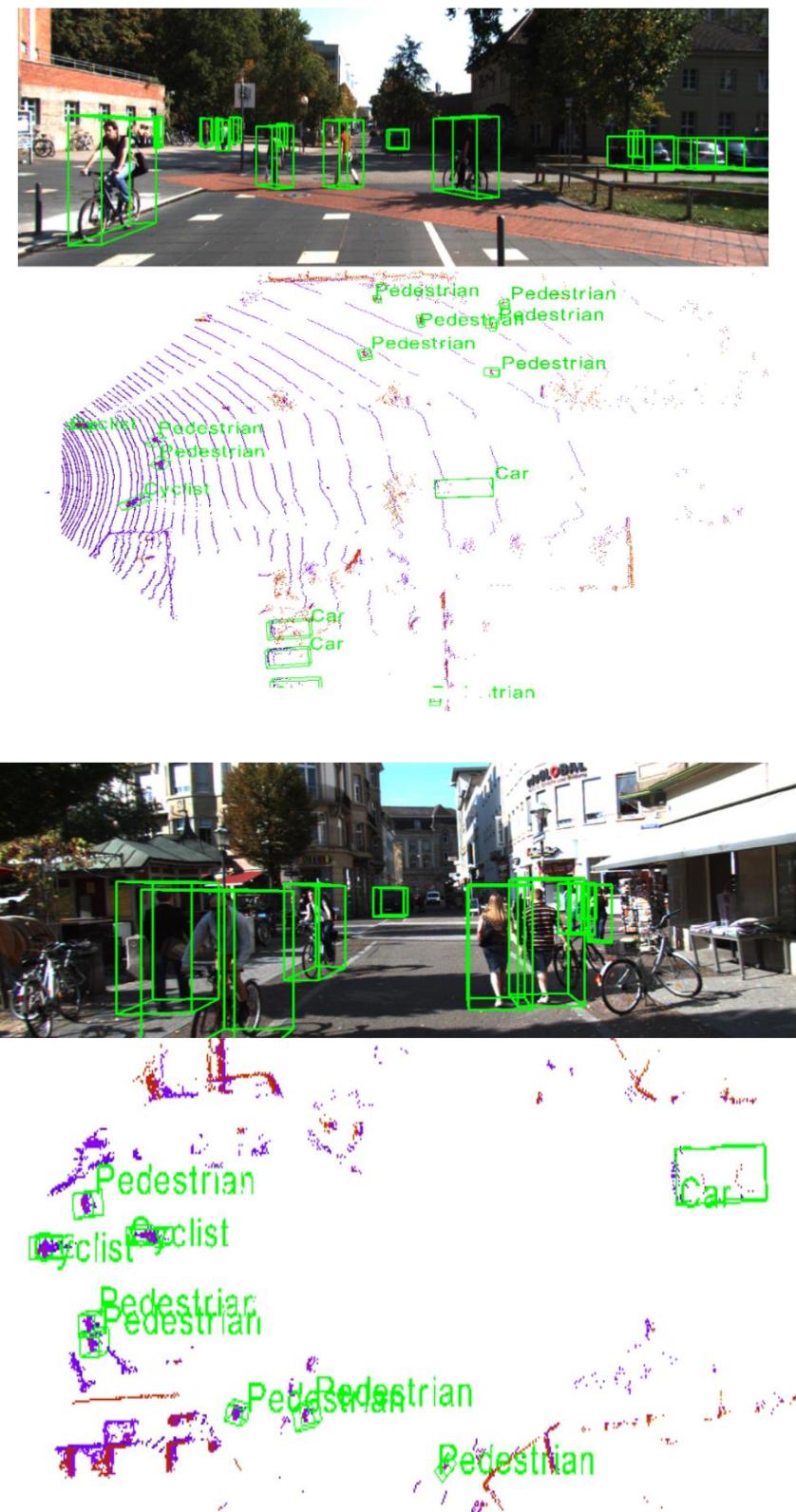


图 3-7 SRDL 在 KITTI 测试集上的定性 3D 检测结果。显示的检测到的对象带有绿色 3D 边界框和相关标签。每个图像的上一行是投影到 RGB 图像上的 3D 对象检测结果，下一行是相应点云中的结果。场景的目标主要是行人和骑自行车的人。

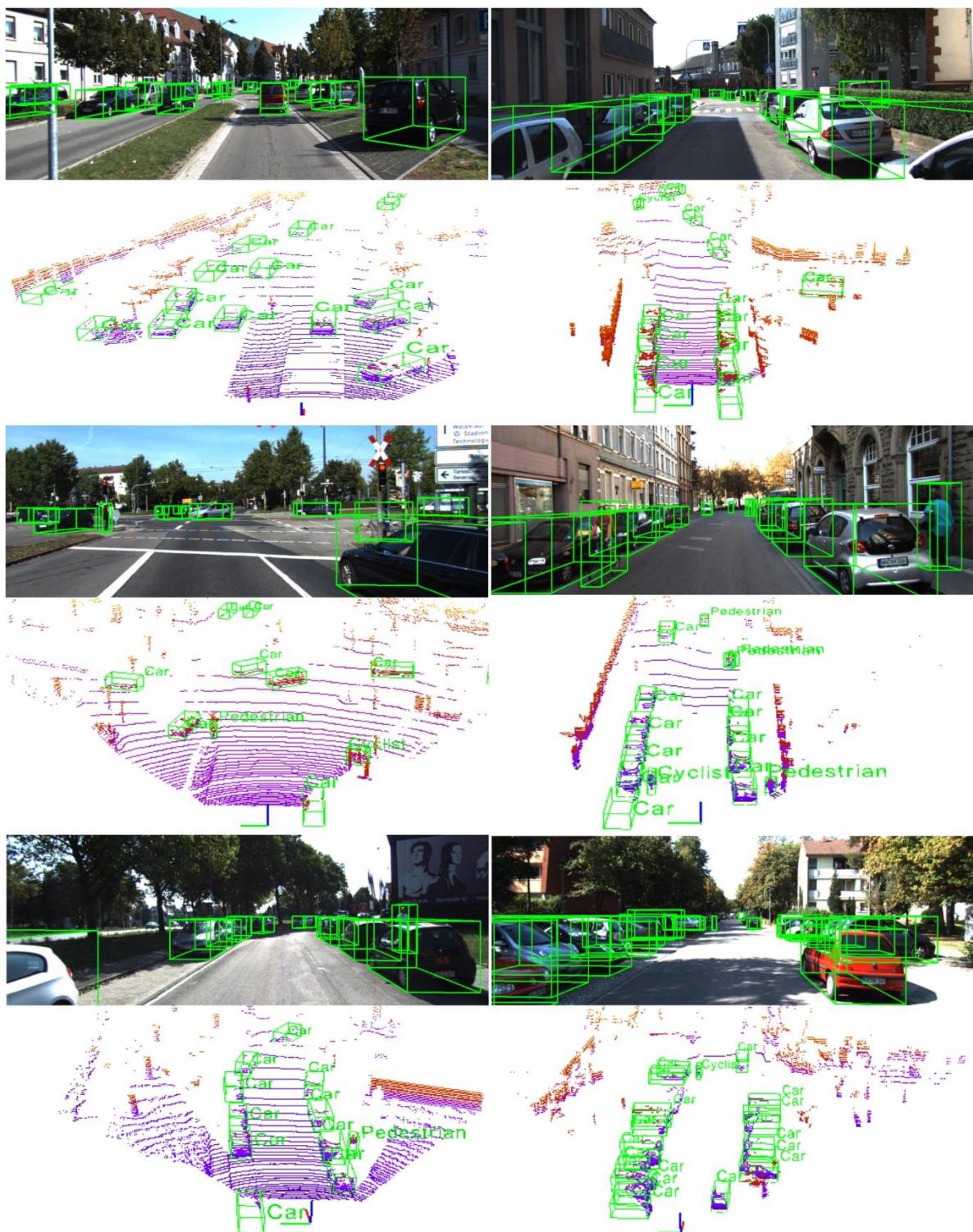


图 3-8 SRDL 在 KITTI 测试集上的定性 3D 检测结果。场景主要由汽车组成。图像的排列方式与图 3-7 相同。

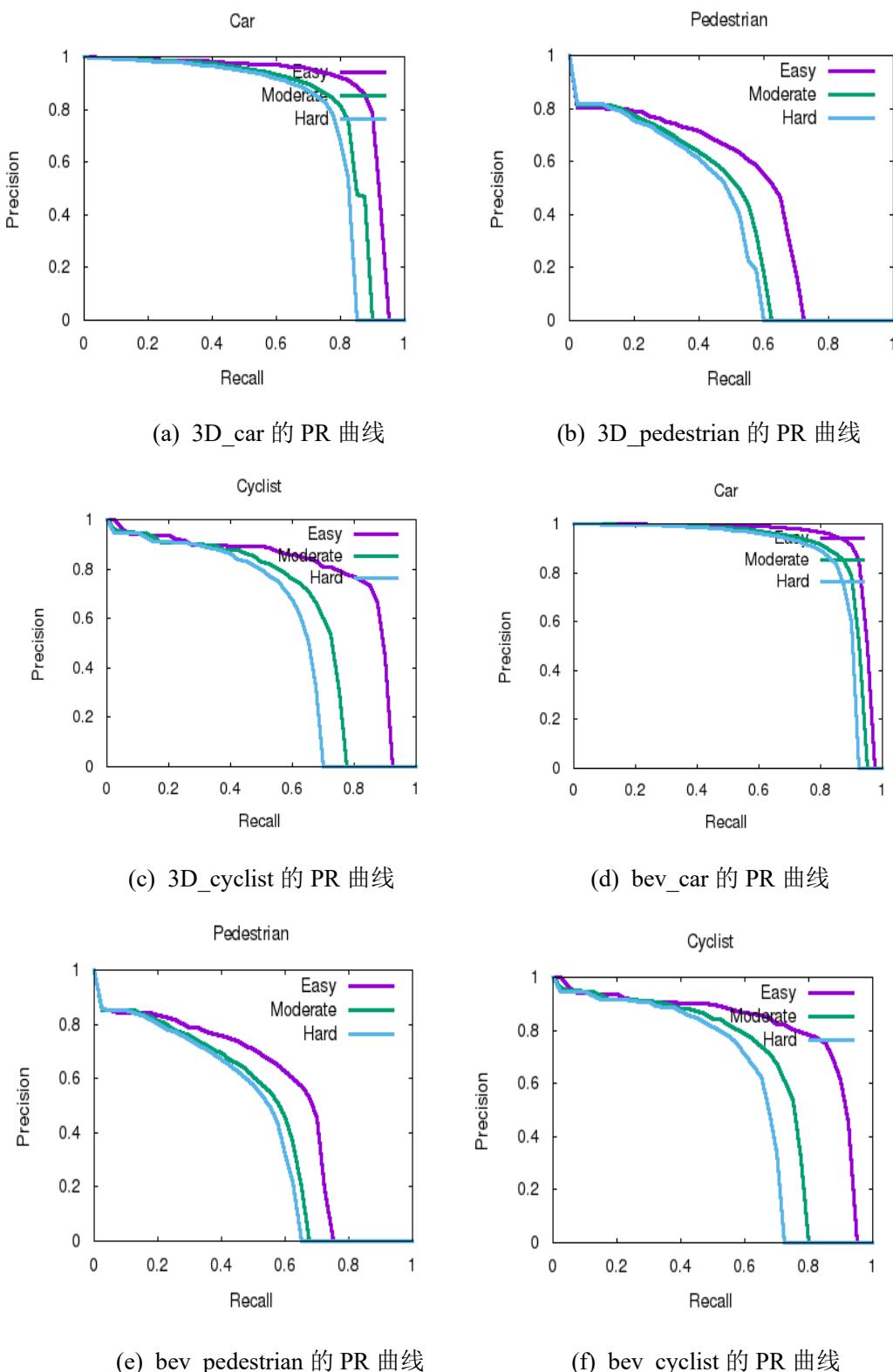


图 3-9 SRDL 在 3D 检测和鸟瞰图 (bev) 检测任务上的 PR 曲线

图 3-7 和图 3-8 所示，在 KITTI 数据集上的测试集中，展示了针对行人/骑车人和汽车的 SRDL 网络的定性结果。从图中可以看出，本章提出的网络可以估计不同情况下的准确 3D 边界框场景。出乎意料的是，观察到即使在非常稀疏的点云和严重遮挡的情况下，本章的方法仍然可以达到令人满意的检测结果。

除此之外，还绘制了在 3D 目标检测和鸟瞰图检测的 PR 曲线，如图 3-9 所示，其中每一个 PR 曲线下的面积即为表中 AP 值的大小。

### 3.5.4 消融实验

在本节中，将通过对 KITTI 的验证集进行广泛的消融实验，来更改本章的 SRDL 的组件和变体。遵循惯例并使用包含最多训练样本的汽车类别。评估指标是验证集上的平均精度（AP%）。

#### 3.5.4.1 不同设计选择对整个网络的影响

如表 3-5 所示，通过删除一部分并将所有其他部分保持不变来说明网络中不同组件的重要性。如果没有输入双目图像（缺少双目代表单目图像），SRDL 的性能将急剧下降。这表明双目可以提供丰富的特征信息来定位目标。同样，AP 在简单，中等和困难程度分别显着降低了 11.65%，12.27%，16.33%，这证实了 3D 边界框编码的不可或缺性。而且由于没有局部或全局卷积进行分割而导致的性能下降证明，只有将它们组合才能产生最佳结果。

#### 3.5.4.2 注意力模块的影响

为了展示注意力模块的重要性，这里在三种不同的设计中添加了注意力模块。如表 3-6 所示，通过注意模块在连接的层之间转移要素，融合模型在中等难度下分别比原始模型高出 1.24%，5.44%，8.49%。对于这三种困难程度，最终的具有注意力机制的融合方法的性能比其他方法分别高 5.77%，8.49%，9.86%。

表 3-5 移除网络不同部分的性能。×表示删除，√表示保留

Stereo	Local	Global	Encoding	简单	中等	困难
×	√	√	√	83.64	73.59	67.48
√	×	√	√	86.77	73.32	71.65
√	√	×	√	88.46	76.71	75.69
√	√	√	×	78.63	67.55	62.32
√	√	√	√	90.28	79.82	78.65

### 3.5.4.3 不同数量的最近邻居 $k$ 的影响

为了在局部 EdgeConv 层中构建最佳动态图，需要为这三个类别选择适当的最近邻居数  $k$ 。如表 3-7 所示，以 5 至 40 范围内的不同数量的最近邻居进行实验，以获取中等难度下的 AP 值。具体来说，对于汽车，当  $k$  为 30 时，AP 达到最大值。但是对于行人和骑自行车的人，当  $k$  设置为 20 时，AP 值达到峰值。可以看到，汽车和行人/骑自行车的人的  $k$  值不同，并且我们认为这可能是由于不同类别中的点稀疏所导致的。此外，虽然没有用所有可能的  $k$  进行测试，但发现随着  $k$  的增加，性能并没有持续改善，这对于所有三个类别都是一致的。原因可能是较大的  $k$  会破坏由欧几里得距离带来的点之间的几何形状。

表 3-6 具有注意机制的不同融合方法的性能比较

融合方法	简单	中等	困难
Global	83.85	71.08	65.73
Local	83.89	71.27	66.54
Global+local	84.51	71.33	68.79
Globa+attention	86.77	73.32	71.65
Local+attention	88.46	76.71	75.69
Global+local+attention	90.28	79.82	78.65

表 3-7 不同数量的最近邻居的性能比较

Number of k	$AP_{moderate}(\%)$		
	Car	Pedestrian	Cyclist
5	80.65	48.96	46.32
10	81.60	51.17	48.76
15	84.75	55.48	50.12
20	87.34	57.64	50.85
25	89.63	55.72	50.45
30	90.28	53.88	49.52
35	89.57	51.56	48.16
40	87.64	50.04	47.22

### 3.5.4.4 不同精修尺寸 $\xi$ 的影响

在 3.3.1 节中，本算法提出通过在框的大小上添加常数  $\xi$  来完善建议框。表 3-8 显示了不同大小的结果。证明  $\xi = -0.5m$  在网络中表现最佳，这意味着应将原始盒子缩小 0.5m。请注意，当放大盒子的尺寸时，尤其是超过 1m 时，AP 的值会急剧下降。这表明原始盒子已经包含多余的空间，并且继续扩大盒子将仅包含更多无关的区域。同时，太大的  $\xi$  不能使框收缩也将导致性能下降，因为小的区域也可能会排除相关区域。

表 3-8 在 3D 盒中采用不同大小的  $\xi$  的性能

精修尺寸	简单	中等	困难
1.5m	72.62	69.86	68.57
1.0m	79.43	70.53	70.82
0.8m	84.59	72.58	71.94
0.5m	87.26	76.25	74.85
0m	89.47	78.84	77.62
-0.5m	90.28	79.82	78.65
-0.8m	89.12	78.76	77.38
-1.0m	87.69	77.17	75.93
-1.5m	84.81	73.91	73.11

### 3.5.4.5 包围盒编码方法的影响

表 3-9 不同边框编码方法的性能比较

编码方法	简单	中等	困难
Axis	79.13	68.42	65.36
8 corners	83.09	74.51	70.82
4 corners + 2 heights	89.61	78.37	77.64
3 points + 2 heights	90.28	79.82	78.65

如第 3.3.2 节所述，有多种边界框编码方法，包括本章提出的方法。这里使用四种不同的方法对网络中的框进行编码。从表 3-9 中，注意到尽管 4 个角+2 高度方法消耗了一些尺寸，但是其性能却比我们的方法差。一方面，“4 个角+2 高度”方法没有考虑四个角之间的坐标关系，因此点数是多余的。另一方面，不能建立拐

角的坐标与高度之间的约束关系。而本章的方法可以建立四组约束关系来分别约束长度，宽度和高度。

### 3.6 本章小结

本章提出了一种新颖的双目 RGB 和更深的 LIDAR (SRDL) 网络，用于自动驾驶场景中的 3D 目标检测。所提出的方法充分利用了双目 RGB 图像和点云的优点，形成了一个端到端的框架。来自双目图像的语义信息和来自点云的空间信息的组合共同有助于提高性能。在公开数据集 KITTI 的 3D 目标检测任务上进行的大量实验充分证明了本章方法的有效性。在未来的研究中，将优化推理速度，更多地关注于集成 RGB 和逐点特征，并且将在点云上添加不同的操作以进一步改善本章的检测框架。

## 第四章 基于稀疏体素图注意力网络的 3D 目标检测方法

随着激光雷达传感器在自动驾驶领域和增强现实领域的广泛应用，基于点云的 3D 目标检测已成为主流研究方向。与 RGB 图像相比，点云能够提供精确的深度和几何信息，可用来定位目标并描述目标的形态。然而，由于点云具有无序性、稀疏性和相关性等特点，直接利用点云进行三维目标检测是一项具有挑战性的任务。正如在第二章中介绍的，针对于基于点云的 3D 目标检测任务，处理点云数据的主要方法是投影法和体素法。然而，由于投影本身不能很好地包含物体的几何信息，同时，仅使用体素化的方法并不能很好地利用点云的特性，并且随着分辨率的提高会带来巨大的计算负担。考虑到点云的不规则性，将图（Graph）应用于 3D 目标检测的优越性便有了体现。事实上，在点的分割与分类任务上，图的方法已经进行了深入研究。但是，很少有研究利用图这种表示方法对点云进行 3D 目标检测。在第二章中介绍的 Point-GNN 可能是第一个将图神经网络引入到 3D 目标检测中来。Point-GNN 引入了自动配准机制来减少平移误差，并设计了图卷积神经网络。然而，在特征提取过程中，点集之间的关系没有很好的建立，大量的矩阵运算会带来沉重的计算负担和存储开销。

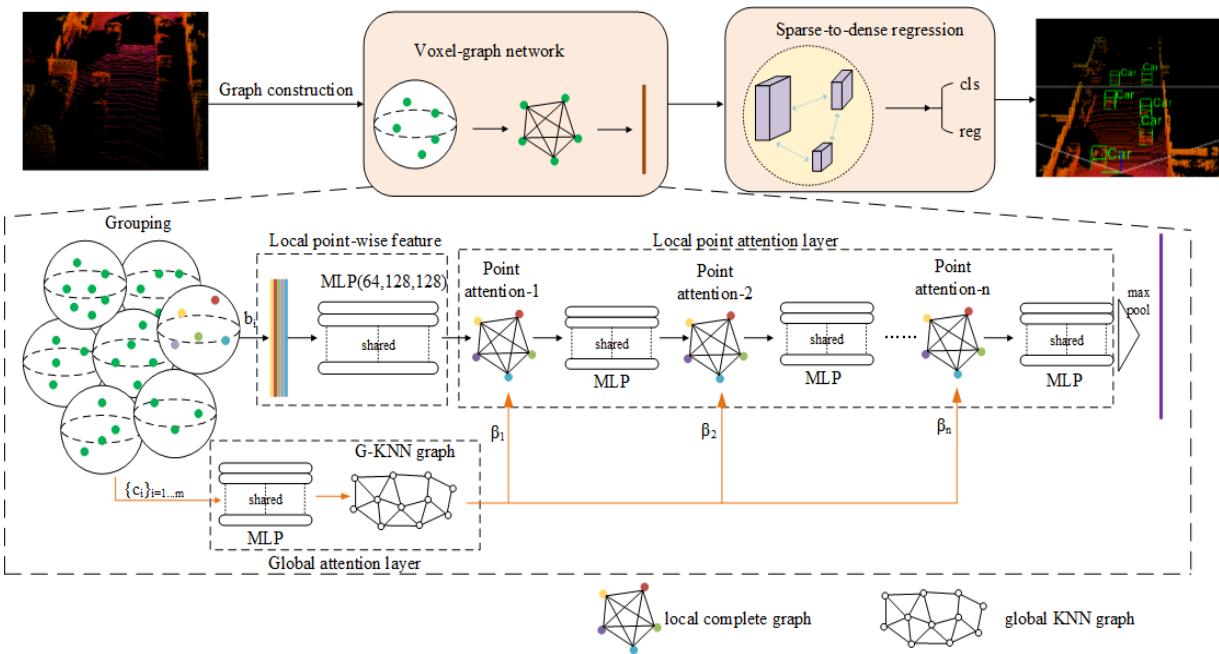


图 4-1 SVGA-Net 网络整体结构图

基于图的表示和原有体素化方法存在的问题，本章提出了一种新的用于 3D 目标检测的网络：稀疏体素图注意网络（Sparse Voxel-Graph Attention Network, SVGA-Net）。SVGA-Net 是一种端到端的可训练网络，它的输入是原始点云，输出是目标的类别和边界框信息。具体来说，SVGA-Net 主要由体素图网络模块和稀疏到稠密回归模块组成。与常规的矩形体素不同，我们将点云划分为半径固定的三维球面空间。体素图网络的目标是为每个体素构造局部完全图，为所有体素构造全局 KNN (K-邻近) 图。通过作用于局部和全局的注意力机制，为点云中每个点的特征向量提供参数监督因子。这样，局部聚集特征就可以与全局点特征相结合。然后通过对不同尺度特征的处理，设计稀疏到稠密的回归模型对目标类别和边界框进行预测。

## 4.1 体素图网络构架

### 4.1.1 球形体素分组

如图 4-1 所示，网络接收到原始点云数据后，首先构建球形体素。假设原始点云表示为  $G=\{V, D\}$ ，其中  $V=\{p_1, p_2, \dots, p_n\}$  表示  $D$  维空间中的  $n$  个点。本文中的  $D$  为 4，该空间中每一个点被定义为  $v_i=[x_i, y_i, z_i, s_i]$ ，其中  $x_i, y_i, z_i$  表示沿 X、Y、Z 轴的每个点的坐标值，第四维是表示为  $s_i$  的激光反射强度。为了更好地覆盖整个点集，我们使用迭代最远点采样<sup>[33]</sup>来选择  $N$  个最远点  $P = \{p_i = [x_i, y_i, z_i, s_i]^T \in \mathbb{R}^4\}_{i=1,2,\dots,N}$ 。根据  $P$  中的每个点，在固定半径  $r$  内搜索其近邻点，形成局部体素球：

$$b_i = \{p_1, p_2, \dots, p_i, \dots, p_j, \mid \|v_i - v_j\|_2 < r\} \quad (4-1)$$

这样，我们可以将整个三维空间细分为  $N$  个 3D 球形体素  $B = \{b_1, b_2, \dots, b_N\}$ 。

### 4.1.2 局部点特征表示

如图 4-1 所示，每个球形体素  $b_i = \{p_i = [x_i, y_i, z_i, s_i]^T\}_{j=1,2,\dots,t}$  包含  $t$  个点（ $t$  随不同体素球而可能发生变化），内部所有点的坐标信息构成输入向量，通过多层感知机（MLP）提取每个体素的局部逐点特征：

$$f(b_i) = MLP(p_j)_{j=1,2,\dots,t} \quad (4-2)$$

其中  $p_j$  代表体素  $b_i$  中的点，同一个体素内的点所用的 MLP 参数共享。之后，我们可以得到每个体素球的特征  $F = \{f_i, i=1, \dots, t\}$ （其中  $f_i$  代表该体素内  $p_i$  点提取出的特征），再由后续的网络层提取更深的特征。

### 4.1.3 局部点注意力层

获取局部逐点特征后，局部点注意力层通过一系列信息聚合输出细化后的特征  $F' = \{f'_i, i=1, \dots, t\}$ 。如图 4-2 所示，我们为每个局部节点集构建了一个完全图，为所有球形体素构建了一个 KNN 图。我们根据局部和全局注意力分数汇总每个节点的信息。

一个局部点注意力层将输入的每一个点的特征进行细化的过程表示为：

$$f'_j = \beta_i \cdot f_j + \sum_{k \in \cup(p_j)} \alpha_{j,k} \cdot f_{j,k} \quad (4-3)$$

其中  $f_j$  表示输入局部点注意力层的  $p_j$  的特征， $f'_j$  表示对  $f_j$  进行再提取获得的特征， $k \in \cup(p_j)$  表示同一体素内除  $p_j$  其他点的索引， $\alpha_{j,k}$  代表点  $p_j$  与同一体素内其他节点之间的局部注意得分， $\beta_i$  代表  $p_j$  所在的索引为  $i$  的体素的全局注意力得分， $f_{j,k}$  表示与  $p_j$  同一体素内第  $k$  个节点的特征。

其中，在进行细化前，通过将体素中每一个点的特征作为节点，为每一个体素构造一个完全图，可以获得局部注意得分：

$$\alpha_{j,k} = softmax_j(f_j, f_{j,k}) = \frac{\exp(f_j^T \cdot f_{j,k})}{\sum_{k \in \cup(p_j)} \exp(f_j^T \cdot f_{j,k})} \quad (4-4)$$

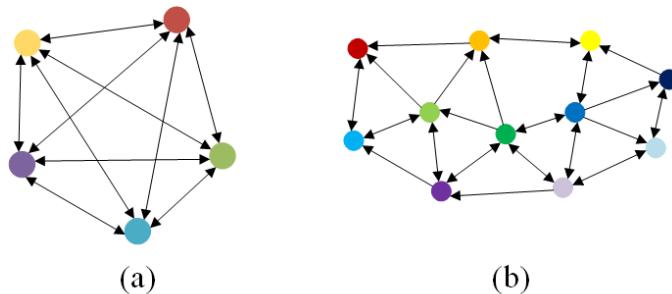


图 3-4 图的构造. 每个具有不同颜色的节点表示聚合的特征，箭头方向表示具有独立注意力计算得分的信息传播方向。（a）局部完整图：对于每个节点，我们根据注意力得分汇总同一球形体素内所有节点的信息。（b）全局 3-NN 图：我们根据注意力得分汇总每个节点周围三个最近邻居的信息。

### 4.1.4 全局注意力层

通过构造局部完整图，聚集的特征只能描述局部特征，而不能整合全局信息。因此，我们设计全局注意力层来学习每个球形体素的全局特征，并提供与每个节点对齐的特征因子。

对于  $N$  个 3D 球形体素  $B = \{b_1, b_2, \dots, b_N\}$  中的每个  $b_i = \{p_i = [x_i, y_i, z_i, s_i]^T\}_{j=1,2,\dots,t}$  中的所有的点云，计算出每一个球形体素的物理中心，然后对所有  $N$  个体素的物理中心  $\{c_i\}_{i=1,\dots,N}$ ，每个中心通过一个参数共享的 MLP 进行特征提取，获得初始全局特征  $F_g = \{f_{g,1}, f_{g,2}, \dots, f_{g,N}\}$ 。如图 4-2 (b) 所示，根据提取出的特征建立  $N$  节点的全局 KNN 图，通过这样得到：

$$\beta_i = \frac{f_{g,i}^T \cdot f_{g,i,l}}{\sum_{l \in \mathcal{U}(f_{g,i})} f_{g,i}^T \cdot f_{g,i,l}} \quad (4-5)$$

其中  $\mathcal{U}(f_{g,i})$  表示节点  $f_{g,i}$  的  $K$  个邻近点索引。

#### 4.1.5 体素图特征表示

通过将局部点注意力层细化后输出的每一个点的特征  $f'_j$  送入一个带非线性激活的 2 层 MLP 进行更新。同一体素内的  $f'_j$  所用的 MLP 参数共享。通过叠加“局部点注意力层+体素图特征表示”，学习局部聚集特征和全局点注意力特征。

完成特征提取后，对每一个点体素的点的特征进行聚合，对聚合后的特征使用最大值池化，得到一组共  $N$  个体素的特征，每个特征与体素的空间位置相对应，这组特征将作为稀疏到稠密回归模块的输入。

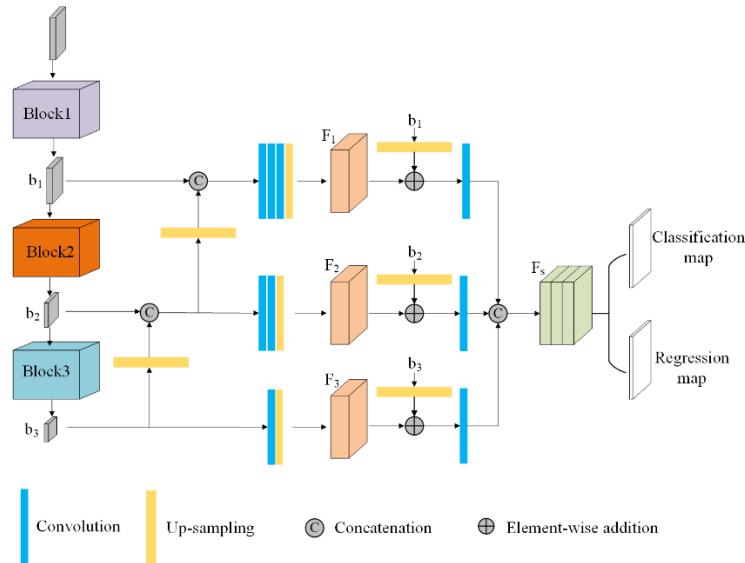


图 4-3 基于多尺度特征融合的 3D 目标检测方法

#### 4.2 稀疏到稠密的回归

对于三维空间中的每个三维边界框，预测的框信息表示为  $(x, y, z, l, w, h, \theta)$ ，其中  $(x, y, z)$  是边界框的中心坐标， $(l, w, h)$  分别是沿着长度、宽度和高度的大小信息，

$\theta$  是目标航向角。利用区域建议回归模块对体素图形网络的特征图进行处理。本章设计的稀疏到密集回归(SDR)模块的体系结构如图 4-3 所示。

SDR 模块首先应用与[39]类似的三个块, 以自上而下产生空间分辨率减小的特征。每个块由一系列 2 维卷积  $\text{Conv2D}(f_{in}, f_{out}, k, s, p)$  层组成, 后面是 Batch-Norm 和 ReLU 激活, 其中  $f_{in}$  和  $f_{out}$  是输入和输出通道的数目,  $k, s, p$  分别表示内核大小、步长和填充大小。对于每个块的第一层, 步长被设置为 2, 从而将特征映射下采样为原来的一半, 然后是与步长为 1 的卷积序列。三个块的输出分别表示为  $b_1$ 、 $b_2$ 、 $b_3$ 。

为了将高分辨率特征与大感受野和低分辨率特征与小感受野结合起来, 将第二和第三模块  $b_2$ 、 $b_3$  的输出与第一和第二模块  $b_1$ 、 $b_2$  的输出进行了级联。这样, 低层的密集特征和高层的稀疏特征可以很好地结合起来。然后, 在三个尺度通道上并行进行一系列包括上采样层的卷积运算, 以生成三个具有相同尺度的特征映射, 这些特征映射被表示为  $F_1$ 、 $F_2$ 、 $F_3$ 。

为了将原始稀疏特征映射和一系列处理过的稠密特征映射结合起来, 将上采样后的原始输出  $b_1$ 、 $b_2$ 、 $b_3$  和  $F_1$ 、 $F_2$ 、 $F_3$  分别逐元素相加后再通过一个核为  $3 \times 3$  的卷积层, 将 3 个融合后的特征进行级联, 得到最终的输出  $F_s$ 。并以  $F_s$  作为输入进行分类和三维边界框回归。

### 4.3 损失函数

我们使用多任务损失来训练我们的网络。每个前锚和地面真值边界框分别参数化为  $(x_a, y_a, z_a, l_a, w_a, h_a, \theta_a)$  和  $(x_{gt}, y_{gt}, z_{gt}, l_{gt}, w_{gt}, h_{gt}, \theta_{gt})$ 。锚与真值之间的回归残差计算如下:

$$\begin{aligned}\Delta x &= \frac{x_{gt} - x_a}{d_a}, \Delta y = \frac{y_{gt} - y_a}{d_a}, \Delta z = \frac{z_{gt} - z_a}{h_a} \\ \Delta w &= \log\left(\frac{w_{gt}}{w_a}\right), \Delta l = \log\left(\frac{l_{gt}}{l_a}\right), \Delta h = \log\left(\frac{h_{gt}}{h_a}\right) \\ \Delta \theta &= \sin(\theta_{gt} - \theta_a)\end{aligned}\quad (4-6)$$

其中  $d_a = \sqrt{(w_a)^2 + (l_a)^2}$ 。采用 Smooth L1 损失作为三维边界框的回归损失

$L_{reg}$ 。

采用分类二叉熵损失作为目标分类损失  $L_{cls}$ :

$$L_{cls} = \gamma_1 \frac{1}{N_{pos}} \sum_i L_{cls}(p_i^{pos}, 1) + \gamma_2 \frac{1}{N_{neg}} \sum_i L_{cls}(p_i^{neg}, 0) \quad (4-7)$$

其中，其中  $N_{pos}$  和  $N_{neg}$  是正锚和负锚（其概念在实施测试部分会详细说明）的数量。 $p_i^{pos}$  和  $p_i^{neg}$  分别是正负锚的 softmax 输出。 $\gamma_1$  和  $\gamma_2$  是用于平衡不同锚正常数。

总损失由分类损失  $L_{cls}$  和边界框回归损失  $L_{reg}$  两部分组成，表示为：

$$L_{total} = \alpha L_{cls} + \beta \frac{1}{N_{pos}} \sum_{t \in \{x, y, z, l, w, h, \theta\}} L_{reg}(\Delta t^*, \Delta t) \quad (4-8)$$

其中  $\Delta t^*$  和  $\Delta t$  分别为预测残差和回归目标。参数  $\alpha$  和  $\beta$  用于平衡两类损失。

## 4.4 实验及分析

### 4.4.1 实验细节

本章的网络架构如图 4-1 所示，在局部点特征层和全局注意层中，点集首先由三层 MLP 处理，其大小均为（64、128、128）。在局部点注意力层，叠加 3 个局部点注意力图来聚合特征，每个特征后面跟着一个 2 层 MLP。三个 MLP 的大小分别为（128，128），（128，256）和（512，1024）。我们训练了两个网络，一个是检测汽车网络，另一个是检测行人和自行车网络。

对于汽车，我们取  $N=1024$  作为初始点集。为了构造局部完全图，我们选择  $r=1.8m$ 。对于锚，如果某一锚框与真值框的 IoU 最高或其 IoU 值超过 0.6，则该锚框被认为正。如果某一锚框与所有真值框的 IoU 都小于 0.45，则锚框被视为负锚。为了减少冗余，我们对非极大抑制（NMS）采用 0.7 的 IoU 阈值。

对于自行车和行人，初始点集的数目为  $n=512$ 。设置  $r=0.8$  来构造局部图。如果一个 anchor 的最高 IoU 分数超过 0.5，则 anchor 被认为是正的。如果 anchor 的 IoU 得分低于 0.35，则 anchor 被认为是负的。NMS 的 IoU 阈值设置为 0.6。

在损失函数上， $\gamma_1$  和  $\gamma_2$  分别取 1.5 和 1； $\alpha$  和  $\beta$  分别取 1 和 2。

网络以端到端的方式进行训练，训练机器是 GTX 1080 GPU。使用的优化器是 ADAM，其初始学习率为 0.001（前 140 轮），每 20 个轮衰减 10 次。我们总共训练 200 轮，都是在 4 张 GPU 上完成的，Batch Size 被设为 16。此外，本章采取[39]中相同的数据增强来防止过拟合。

#### 4.4.2 数据集以及评价指标

本章提出的算法在广泛使用的 KITTI 3D 目标检测基准进行了评估。所用的 KITTI 数据集的分布与处理方式及本章对于训练集、验证集和测试集的处理方法同第三章。在评价中，平均精度（AP）指标是比较不同方法，汽车、自行车和行人的 3D IoU 阈值分别为 0.7、0.5 和 0.5。

表 4-1 汽车、行人和骑行者 KITTI 3D 目标检测性能比较。“R”表示 RGB 图像输入，“L”表示激光雷达点云输入。

方法	模式	AP <sub>car</sub> (%)			AP <sub>pedestrian</sub> (%)			AP <sub>cyclist</sub> (%)		
		简单	中等	困难	简单	中等	困难	简单	中等	困难
MV3D <sup>[52]</sup>	R+L	71.09	62.35	55.12	-	-	-	-	-	-
F-Pointnet <sup>[20]</sup>	R+L	81.20	70.39	62.19	51.21	44.89	40.23	71.96	56.77	50.39
AVOD-fpn <sup>[60]</sup>	R+L	81.94	71.88	66.38	50.80	42.81	40.88	64.00	52.18	46.61
F-ConvNet <sup>[79]</sup>	R+L	85.88	76.51	68.08	52.37	45.61	41.49	79.58	64.68	57.03
MMF <sup>[80]</sup>	R+L	86.81	76.75	68.41	-	-	-	-	-	-
Voxelnet <sup>[39]</sup>	L	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
SECOND <sup>[46]</sup>	L	83.13	73.66	66.2	51.07	42.56	37.29	70.51	53.85	46.9
PointPillars <sup>[47]</sup>	L	79.05	74.99	68.3	52.08	43.43	41.49	75.78	59.07	52.92
Pointrcnn <sup>[10]</sup>	L	85.94	75.76	68.32	49.43	41.78	38.63	73.93	59.6	53.59
STD <sup>[81]</sup>	L	86.61	77.63	76.06	53.08	44.24	41.97	78.89	62.53	55.77
3DSSD <sup>[40]</sup>	L	88.36	79.57	74.55	-	-	-	-	-	-
SA-SSD <sup>[84]</sup>	L	88.75	79.79	74.16	-	-	-	-	-	-
PV-RCNN <sup>[43]</sup>	L	90.25	81.43	76.82	-	-	-	78.60	63.71	57.65
Point-GNN <sup>[82]</sup>	L	88.33	79.47	72.29	51.92	43.77	40.14	78.60	63.48	57.08
<b>SVGA-Net</b>	L	87.40	80.82	76.23	47.59	39.88	37.57	75.45	61.86	54.68

#### 4.4.3 实验结果及分析

实验在 KITTI 测试集的 3D 检测基准和鸟瞰检测基准上评估了本章的算法。结果如表 4-1 和表 4-2 所示，“R”表示 RGB 图像输入，“L”表示激光雷达点云输入。将本章结果与最新的 RGB+Lidar 和仅用 Lidar 的 3D 对象检测和鸟瞰检测任务的方法进行了比较。本章提出的方法比最有效的 RGB+Lidar 方法 MMF<sup>[80]</sup> 在 3D 检测和 BEV 检测的三个难度级别上分别有较大的提高。

与基于 Lidar 的方法相比，本章的 SVGA 网络在这三类问题上仍然表现出良好的性能。特别是，在三个类别的检测中使用相同的图表示方法远远优于 Point-GNN<sup>[82]</sup>。这可能得益于构建的局部和全局图，以更好地捕捉点云的特征信息。

表 4-2 汽车、行人和骑自行车者 KITTI 鸟瞰检测性能比较

方法	模式	$AP_{car}(\%)$			$AP_{pedestrian}(\%)$			$AP_{cyclist}(\%)$		
		简单	中等	困难	简单	中等	困难	简单	中等	困难
MV3D <sup>[52]</sup>	R+L	86.02	76.9	68.49	-	-	-	-	-	-
F-Pointnet <sup>[20]</sup>	R+L	88.70	84.00	75.33	58.09	50.22	47.2	75.38	61.96	54.68
AVOD-fpn <sup>[60]</sup>	R+L	88.53	83.79	77.9	58.75	51.05	47.54	68.09	57.48	50.77
F-ConvNet <sup>[79]</sup>	R+L	89.69	83.08	74.56	58.9	50.48	46.72	82.59	68.62	60.62
MMF <sup>[80]</sup>	R+L	89.49	87.47	79.10	-	-	-	-	-	-
Voxelnet <sup>[39]</sup>	L	89.35	79.26	77.39	46.13	40.74	38.11	66.7	54.76	50.55
SECOND <sup>[46]</sup>	L	88.07	79.37	77.95	55.1	46.27	44.76	73.67	56.04	48.78
PointPillars <sup>[47]</sup>	L	88.35	86.1	79.83	58.66	50.23	47.19	79.14	62.25	56
Pointrcnn <sup>[10]</sup>	L	89.47	85.58	79.10	-	-	-	81.52	66.77	60.78
STD <sup>[81]</sup>	L	89.66	87.76	86.89	60.99	51.39	45.89	81.04	65.32	57.85
SA-SSD <sup>[84]</sup>	L	95.03	91.03	85.96	-	-	-	-	-	-
PV-RCNN <sup>[43]</sup>	L	94.98	90.65	86.14	-	-	-	82.49	68.89	62.41
Point-GNN <sup>[82]</sup>	L	93.11	89.17	83.90	55.36	47.07	44.61	81.17	67.28	59.67
<b>SVGA-Net</b>	L	91.98	88.21	85.46	51.45	44.57	42.45	78.93	66.66	59.55

与第三章中 SRDL 的对比与分析：本章的 SVGA-Net 算法在 KITTI 测试集和验证集上的检测结果总体来说要优于第三章中 SRDL 的结果，在 KITTI 官方网站上的排名也高于 SRDL，原因可能为：（1）使用 RGB 与点云融合的算法结构中，这种不同数据仍存在异源数据配准的问题，所设计的两阶段算法仍然不能很好的融合信息，（2）SVGA-Net 中所设计的从局部和全局两个维度学习特征约束参数的方式可以很好的对点云特征的更新进行监督，所学习的特征更加聚合于目标，（3）对于存在遮挡的目标，RGB 图像数据不能像点云一样给出区分。

定性结果：图 4-4 展示了在 KITTI 数据集的测试集上 SVGA-Net 的一些定性预测边界结果。检测到的对象显示为绿色的三维边界框和相关标签。每幅图像的上一行是投影到 RGB 图像上的三维目标检测结果，下一行是对应点云的结果。从图中可以看出，本章提出的网络能够在不同场景中准确地估计出三维边界框，即使在恶劣的光照条件和严重的遮挡下，仍然可以生成精确的三维边界框。

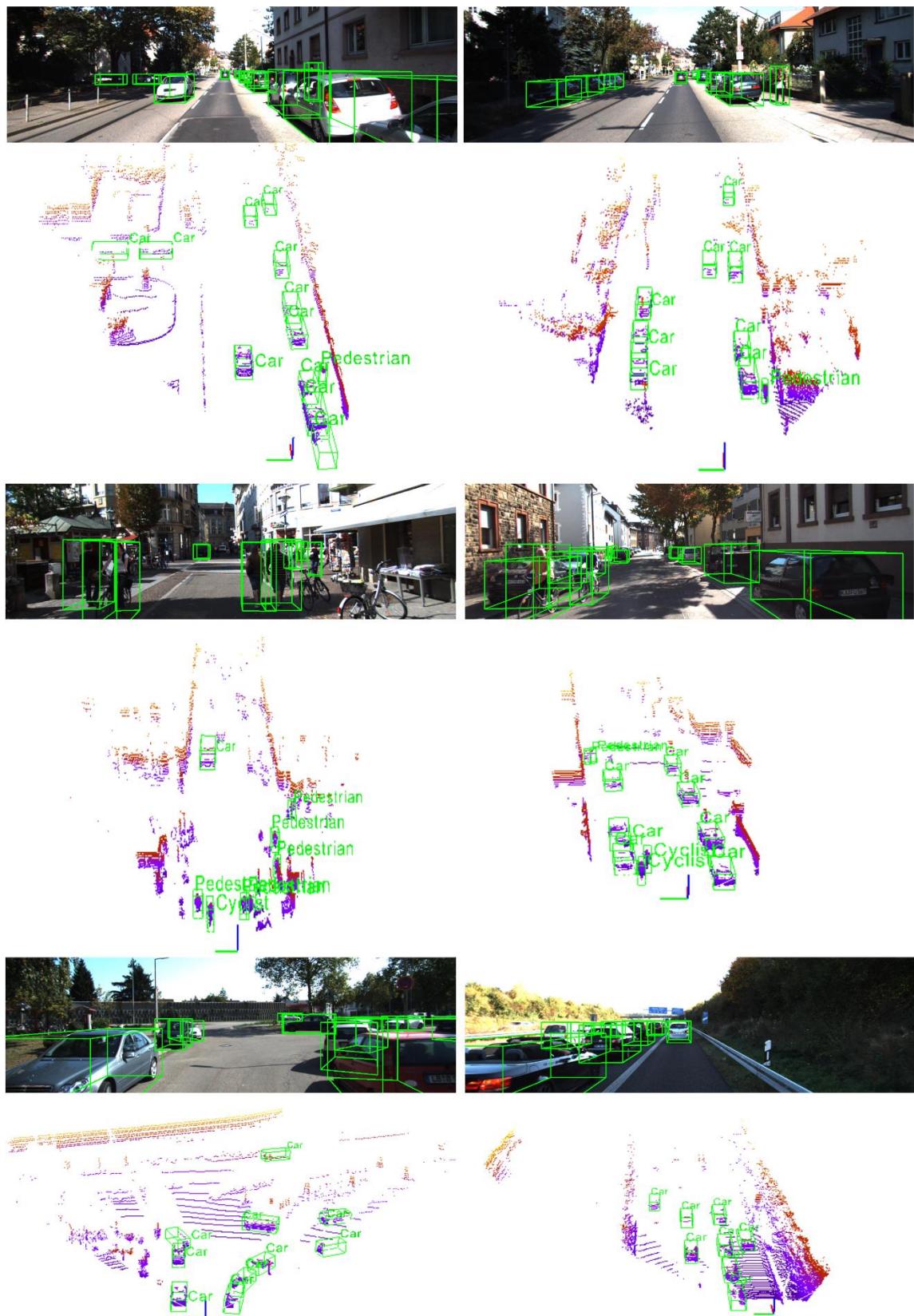


图 4-4 SVGA-Net 在 KITTI 测试集上的定性 3D 检测结果

表 4-3 KITTI 汽车三维目标检测验证集性能比较

方法	模式	$AP_{car}(\%)$		
		简单	中等	困难
MV3D <sup>[52]</sup>	R+L	71.29	62.68	56.56
F-Pointnet <sup>[20]</sup>	R+L	83.76	70.92	63.65
AVOD-FPN <sup>[60]</sup>	R+L	84.41	74.44	68.65
Cont-Fuse <sup>[83]</sup>	R+L	86.32	73.25	67.81
F-ConvNet <sup>[79]</sup>	R+L	89.02	78.8	77.09
Voxelnet <sup>[39]</sup>	L	81.97	65.46	62.85
SECOND <sup>[46]</sup>	L	87.43	76.48	69.1
PointRCNN <sup>[10]</sup>	L	88.88	78.63	77.38
Fast PointRCNN <sup>[77]</sup>	L	89.12	79	77.48
STD <sup>[81]</sup>	L	89.7	79.8	79.3
SA-SSD <sup>[84]</sup>	L	90.15	79.91	78.78
3DSSD <sup>[40]</sup>	L	89.71	79.45	78.67
Point-GNN <sup>[82]</sup>	L	87.89	78.34	77.38
<b>SVGA-Net</b>	<b>L</b>	<b>90.59</b>	<b>80.23</b>	<b>79.15</b>

除此之外,从多角度展示本章所提出算法的性能,还分别绘制了 car、pedestrian 及 cyclist 在 3D 目标检测和鸟瞰图检测任务上的 PR 曲线,如图 4-5 所示,其中每一个 PR 曲线下的面积即为表中 AP 值的大小。

#### 4.4.4 消融实验

通过基于 KITTI 的验证集进行了一系列广泛的消融研究,进而说明每个模块在改进最终结果和参数选择方面的作用。所有的消融研究都是在包含最多训练实例的汽车类上进行的。评估指标是验证集的平均精度 (AP%)。

在 KITTI 验证集上的性能如表 4-3 和表 4-4 所示。对于汽车类,本章提出的方法在三个难度水平上都取得了比目前最先进的方法更好或可比的结果,说明了该方法的优越性。

在局部点注意力中,叠加几个局部点注意力层来提取聚集特征。注意力层数目 n 从 1 到 4 的变化来训练网络。如表 4-5 所示,当在第 1 层到第 3 层上发送局部特征信息时,由于特征被连续聚合到目标本身,因此检测精度不断提高。当 n 增加到 4 时,检测精度略有下降,该网络存在过拟合倾向。

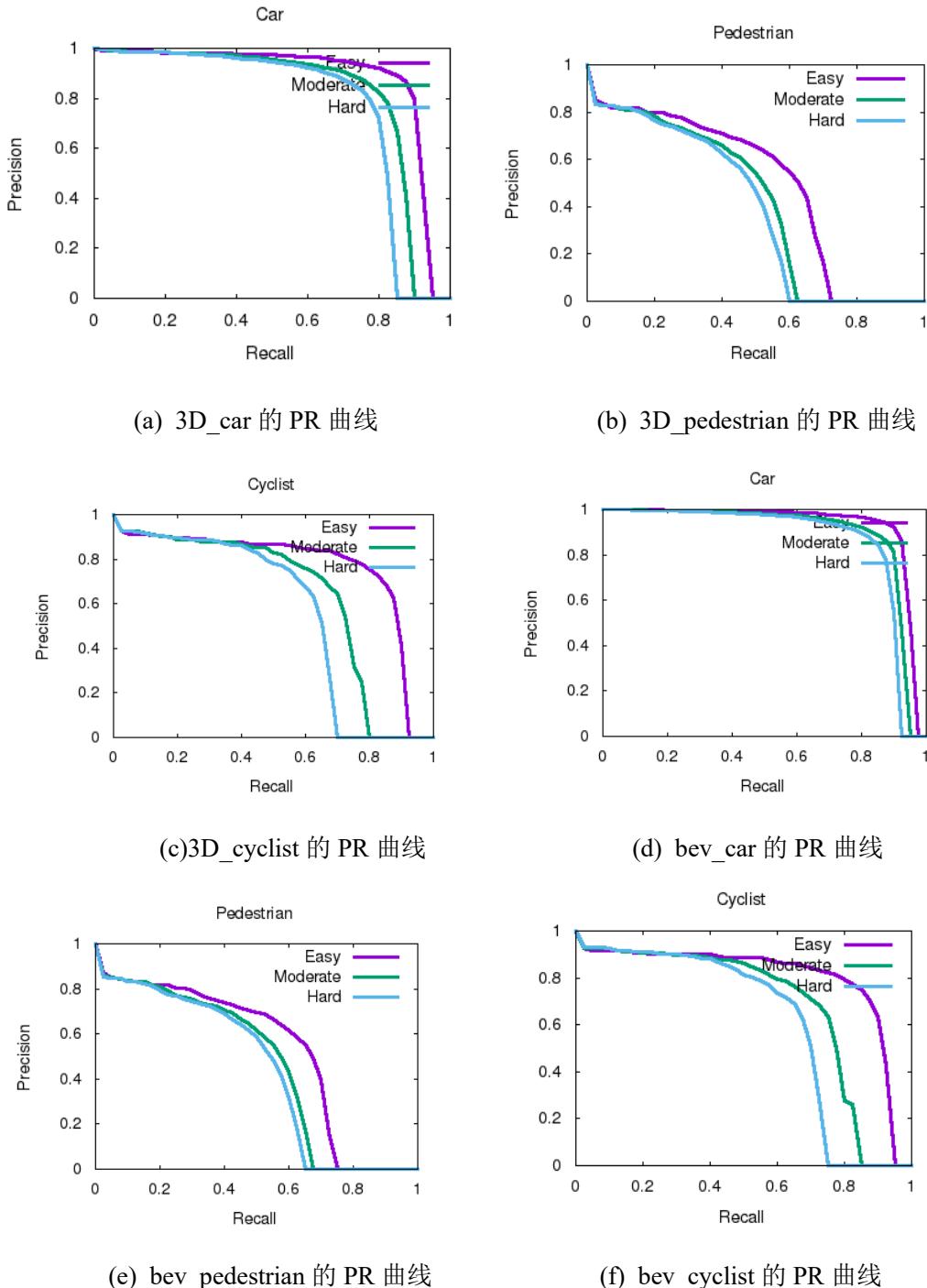


图 4-5 SVGA-Net 在 3D 检测和鸟瞰图 (bev) 检测任务上的 PR 曲线

此外，还研究了全局注意层对提高检测精度的重要性。如表 4-5 所示，当从网络中移除此模块时，两个检测任务上的 AP 值都大大降低，这证明了此设计在为每个点提供全局特征信息方面的重要性。

表 4-4 KITTI 汽车鸟瞰检测验证集性能比较

方法	模式	$AP_{car}(\%)$		
		简单	中等	困难
MV3D <sup>[52]</sup>	R+L	86.55	78.1	76.67
F-Pointnet <sup>[20]</sup>	R+L	88.16	84.02	76.44
F-ConvNet <sup>[79]</sup>	R+L	90.23	88.79	86.84
Voxelnet <sup>[39]</sup>	L	89.6	84.81	78.57
SECOND <sup>[46]</sup>	L	89.96	87.07	79.66
Fast Pointrcnn <sup>[77]</sup>	L	90.12	88.1	86.24
STD <sup>[81]</sup>	L	90.5	88.5	88.1
Point-GNN <sup>[82]</sup>	L	89.82	88.31	87.16
<b>SVGA-Net</b>	<b>L</b>	<b>90.27</b>	<b>89.16</b>	<b>88.11</b>

表 4-5 不同设计选择下的性能比较。n 是点-注意层力的数量。“w/o.”表示是否保留全局注意层。SDR 表示稀疏到稠密的回归。

		3DAP <sub>car</sub> (%)			BEV AP <sub>car</sub> (%)		
		简单	中等	困难	简单	中等	困难
n	1	86.77	75.37	74.19	87.54	86.11	83.72
	2	88.86	78.81	78.03	89.04	88.44	87.05
	3	90.59	80.23	79.15	90.27	89.16	88.11
	4	89.62	79.26	77.58	89.72	88.51	87.17
w/o.	o.	88.42	78.11	76.54	89.71	87.45	84.33
	w.	90.59	80.23	79.15	90.27	89.16	88.11
k	1	76.37	69.15	68.47	82.11	80.27	79.58
	2	84.53	75.61	71.92	86.23	85.65	83.66
	3	90.59	80.23	79.15	90.27	89.16	88.11
	4	88.91	79.22	77.86	80.07	87.88	87.08
	5	86.58	76.82	75.43	85.29	84.38	83.47
SDR		87.53	77.81	76.22	86.95	86.62	85.04
DR		88.39	78.44	76.56	87.91	86.82	86.73
<b>SDR</b>		<b>90.59</b>	<b>0.23</b>	<b>79.15</b>	<b>90.27</b>	<b>89.16</b>	<b>88.11</b>

构造 KNN 图时，在对验证集进行一系列实验后，在实验中选择数字“3”，如表 4-5 的中间五行所示。当 K 从 1 增加到 3 时，AP 值显着增加，但是当它继续增加时，AP 值确实会减少。

在表 4-5 的最后三行中，探讨了稀疏到密集回归模块中不同设计的效果。如图 4-3 所示，SR 是去除  $b_1$ 、 $b_2$  与上采样  $b_2$ 、 $b_3$  的连接，DR 是去除  $b$  与  $F$  的相加。结果表明，只有稀疏到稠密回归的设计才能在提高检测精度取得好的效果。

运行时间：实际测试网络使用 Python 编写，并用 Pytorch 实现 GPU 计算。在运行时间上，一个样本的平均推断时间为 62ms，其中 14.5% (9ms) 用于数据读取和预处理，66.1% (41ms) 用于局部和全局特征聚合，19.4% (12ms) 用于最终边界框检测。

## 4.5 本章小结

本章提出了一种新颖的稀疏体素图注意力网络（SVGA-Net），用于从原始点云进行 3D 目标检测。本章提出的算法将图形表示引入来处理点云，通过在划分的球面体素空间中构造局部完整图，可以获得点特征的更好的局部表示，并且可以融合点及其邻域之间的信息。通过构建全局图，我们可以更好地监督和学习点的特征。此外，稀疏到密度回归模块还可以融合不同比例的特征图。在 KITTI 数据集上的实验结果也证明了本章提出的算法的可行性，而且系列消融实验也证明了本章网络中各种设计选择的有效性。

## 第五章 总结与展望

### 5.1 全文总结

对于许多现实世界的应用，例如自动驾驶和增强现实（AR），准确而鲁棒的 3D 目标检测至关重要且必不可少。最新的方法可以实现 2D 目标检测的较高的平均精度（AP），并且在测试公共数据集（例如 KITTI 和 COCO）上达到很高的水平。然而，由于点云的稀疏性和不规则性，无法直接将 2D 检测网络应用到 3D 检测任务。因此，本文在深入探究了当下在 3D 目标检测领域中的经典算法、探究了所利用的 RGB 图像数据及 LIDAR 点云数据的数据特性的基础之上，针对已有的工作中存在的信息丢失及不同数据融合等问题展开了深入研究，本文的主要工作内容如下：

（1）本文针对所能利用的 RGB 图像数据和 LIDAR 点云数据，从利用的数据源的不同的角度出发，重点研究了基于 RGB 图像数据的 3D 目标检测算法、基于 LIDAR 点云数据的 3D 目标检测算法及基于 RGB 与 LIDAR 点云融合的 3D 目标检测算法。对于这三类算法，对每一类算法中都跳出了具有代表性的网络进行了深入研究。对于基于 RGB 图像数据的 3D 目标检测，从不同处理 RGB 图像数据的角度出发，重点研究了利用不同视角的图像数据所设计的网络的不同之处；对于基于 LIDAR 点云的 3D 目标检测算法，重点研究了针对如何对于点云的手段所设计的网络；对于基于 RGB 与 LIDAR 点云融合的 3D 目标检测算法，重点研究了不同数据的融合方式。

（2）在针对非同源数据的融合上，本文提出了一种双目 RGB 与 LIDAR 点云融合的网络构架。由于双目 RGB 可以从不同视角针对同一目标提供不同的感受野，所提供的融合后的预选框也比单目视角更贴合目标所在的位置，因此首先提出了一种双目 RGB 的融合方案；在针对于点云数据的处理上，在所要使用的点云切割网络中加入边缘卷积及残差注意力模块来融合更多的点云局部邻域信息及生成更紧凑的点云表示。除此之外，在进行边框回归时，提出一种新的边框编码方案，通过坐标变换的方式减少目标框中点的使用。每一项设计均证明了对最终目标的预测起正向作用。

（3）在对点云数据的处理上，本文提出了一种用图这种新的点云表示方式来处理不规则的点云数据。通过将点云划分为球形体素空间，可以更贴合目标的将点云进行切分，在所切分的球形体素中，从局部到全局两个维度建立点与点之间的连接关系，从局部和全局的注意力机制中分布学得一个监督因子对点的特征向量的

更新做出约束，最终形成具有局部和全局邻域信息的特征图。在对于所学得的紧凑的点云的特征图，设计了一种融合了不同尺度特征图信息的回归网络，从稀疏到稠密对目标进行预测。

## 5.2 后续工作展望

在 3D 目标检测中本文提出的方法在对不同数据处理上做出了相应的设计，在对车辆的 3D 检测方面取得了一定的结果，但是由于数据融合的复杂性、点云数据的无序性等问题还有很多地方可以做更深入的研究：

1、在数据集的层面上，本文重点是在 KITTI 这一数据集挑战下进行的研究，这一数据集也是发布了近十年的公认的权威数据集，但是正如在第一章中所述，近年来随着自动驾驶的发展，众多公司于研究机构又发布了包括 Waymo、nuScenes 等新的数据集，这些数据集中包含的场景更加丰富复杂，每一个场景中所包含的角度、目标个数及点云数量也更多，对于这种更加复杂的场景下的 3D 目标检测又将面临新的挑战，因此在这种大型场景下的 3D 目标检测值得进一步的研究。

2、在所利用的点云数据上，包括本文在内的众多工作所利用的都是 3D 点云，即从 LIDAR 中直接扫描的数据，但是近年来已经有少量的工作将开始关注 Rang Image 表示的点云数据，所谓的 Rang Image 即为点云的柱面投影结果，这种柱面视角下的表示其实更加符合人眼对于点云目标的直接观察表示，因此可以将 Rang Image 这种表示下的点云数据替换为当下主流算法中的点云，再做出相应的网络模型的改进以提高检测精度。

3、对于点云数据的处理上，目前针对点云数据的处理包含四种方式，分别为体素化处理方法、投影法、图表示方法和直接点云数据处理，但是这四种方式都不是可逆的点云处理方法，那么如何将点云进行可逆的编码，将 3D 的点云表示转换为紧凑的 2D 表示，而这一转换过程是无损且可逆的，将是深入研究的方向。

4、在网络结构设计上，目前针对于点云数据所设计的网络，不管是可以作为 backbone 的网络还是独立应用的网络都不能很好的做到检测精度与速度兼顾，因此如何像 2D 检测中既能加速网络、减少复杂度又能达到较高的检测准确率是值得深入探究的方向。

## 致 谢

转眼间，三年的硕士研究生生涯就要结束了，会想起拿着录取通知书刚入学的时候就仿佛昨天一样，而现在却已经要说再见了。三年的时间里，感谢母校电子科技大学给我提供的优良的学习与成长的环境，从一名懵懂的本科生成长为一名对科研与生活满怀热情的研究生，在这段生涯中收获满满，感谢所有陪我成长的老师与同学。

首先，感谢我的研究生导师，王正宁老师。感谢王老师对我学术生涯的指点迷津，在我最迷茫与无助的时候总能从王老师身上看到未来努力的方向，一次次的让我这艘迷失方向的航船找到灯塔的方向。感谢王老师对我在科研道路上的悉心教导，您对于科研孜孜不倦、认真严谨的精神让我受益匪浅，这种对于科研的高标准追求的精神将使我受益终生。感谢王老师在我科研道路上提供的各种帮助与各种资源，是您提供的资源让我在科研道路上加速成长，让我这一个星星之火逐渐成长为燎原之势，衷心的感谢您无私的奉献与帮助。

其次，我要感谢我家人在我读研期间对我的支持与鼓励，感谢您们为我的生活提供的保障，感谢为我付出的操劳，您们对我的支持永远是我坚持下去最有力的支柱。

感谢教研室中赵德明，万思琦同学在科研与日常生活对我的帮助，让我平淡的生活增添了许多色彩。感谢曾浩师兄，曾仪师妹，刘怡君师妹对我科研上的鼎力支持，特别是不能回校期间对我写论文、发论文、跑实验的帮助，让我能够如鱼得水。感谢曾凡伟师兄、吕侠师兄、周阳师兄、冯龙飞师兄在生活和科研上对我的指点，让我在学习的路上少走了弯路，让我生活更加乐观。

最后，衷心感谢评审我的论文的各位专家与老师，您们的<sup>67</sup>意见将是我宝贵的财富！

## 参考文献

- [1] World Health Organization. Global status report on road safety 2018: Summary[R]. World Health Organization, 2018
- [2] Peden M M, Puvanachandra P. Looking back on 10 years of global road safety[J]. International health, 2019, 11(5): 327-330
- [3] Liu P, Yang R, Xu Z. How safe is safe enough for self - driving vehicles?[J]. Risk analysis, 2019, 39(2): 315-325
- [4] Chen S, Kuhn M, Prettner K, et al. The global macroeconomic burden of road injuries: estimates and projections for 166 countries[J]. The Lancet Planetary Health, 2019, 3(9): e390-e398
- [5] Litman T. Autonomous vehicle implementation predictions: Implications for transport planning[J]. 2020
- [6] Shuttleworth J. SAE Standards News: J3016 automated-driving graphic update[J]. SAE International, 2019
- [7] Gruyer D, Magnier V, Hamdi K, et al. Perception, information processing and modeling: Critical stages for autonomous driving applications[J]. Annual Reviews in Control, 2017, 44: 323-341
- [8] Van Brummelen J, O'Brien M, Gruyer D, et al. Autonomous vehicle perception: The technology of today and tomorrow[J]. Transportation research part C: emerging technologies, 2018, 89: 384-406
- [9] Pendleton S D, Andersen H, Du X, et al. Perception, planning, control, and coordination for autonomous vehicles[J]. Machines, 2017, 5(1): 6
- [10] Shi S, Wang X, Li H. Pointrcnn: 3d object proposal generation and detection from point cloud[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 770-779
- [11] Girshick R. Fast r-cnn[C]. Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448
- [12] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]. Advances in neural information processing systems. 2015: 91-99
- [13] Arnold E, Al-Jarrah O Y, Dianati M, et al. A survey on 3d object detection methods for autonomous driving applications[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(10): 3782-3795

- [14] Elkhalili O, Schrey O M, Ulfig W, et al. A  $64 \times 8$  pixel 3-D CMOS time of flight image sensor for car safety applications[C]. 2006 Proceedings of the 32nd European Solid-State Circuits Conference. IEEE, 2006: 568-571
- [15] Cadena C, Carlone L, Carrillo H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J]. IEEE Transactions on robotics, 2016, 32(6): 1309-1332
- [16] 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. IEEE, 2012: 3354-3361
- [17] Caesar H, Bankiti V, Lang A H, et al. nuscenes: A multimodal dataset for autonomous driving[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11621-11631
- [18] Sun P, Kretzschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: Waymo open dataset[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 2446-2454
- [19] Geyer J, Kassahun Y, Mahmudi M, et al. A2d2: Audi autonomous driving dataset[J]. arXiv preprint arXiv:2004.06320, 2020
- [20] Qi C R, Liu W, Wu C, et al. Frustum pointnets for 3d object detection from rgb-d data[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 918-927
- [21] He Q, Wang Z, Zeng H, et al. Stereo RGB and Deeper LIDAR Based Network for 3D Object Detection[J]. arXiv preprint arXiv:2006.05187, 2020
- [22] D. G. Lowe. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110
- [23] X. Ren, D. Ramanan. Histograms of sparse codes for object detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013, 3246-3253
- [24] C. C. Chang, C. J. Lin. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): 27
- [25] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904-1916
- [26] J. Redmon, S. Divvala, R. Girshick, et al. You only look once: Unified, real-time object detection[C]. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2016, 779-788

- [27] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. computer vision and pattern recognition, 2017: 6517-6525
- [28] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv: Computer Vision and Pattern Recognition, 2018
- [29] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot Multi Box Detector[J]. European conference on computer vision, 2016: 21-37
- [30] Duan K, Bai S, Xie L, et al. Centernet: Keypoint triplets for object detection[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 6569-6578
- [31] Law H, Deng J. CornerNet: Detecting objects as paired keypoints[C]. Proceedings of the European Conference on Computer Vision (ECCV). 2018: 734-750
- [32] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660
- [33] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[C]. Advances in neural information processing systems. 2017: 5099-5108
- [34] Uijlings J R R, Van De Sande K E A, Gevers T, et al. Selective search for object recognition[J]. International journal of computer vision, 2013, 104(2): 154-171
- [35] Ma X, Wang Z, Li H, et al. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 6851-6860
- [36] Li P, Chen X, Shen S. Stereo r-cnn based 3d object detection for autonomous driving[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 7644-7652
- [37] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]. Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969
- [38] Sun J, Chen L, Xie Y, et al. Disp R-CNN: Stereo 3D object detection via shape prior guided instance disparity estimation. In 2020 IEEE[C]. CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020: 10545-10554
- [39] Zhou Y, Tuzel O. Voxelnets: End-to-end learning for point cloud based 3d object detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4490-4499
- [40] Yang Z, Sun Y, Liu S, et al. 3dssd: Point-based 3d single stage object detector[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11040-11048

- [41] Tian Z, Shen C, Chen H, et al. Fcos: Fully convolutional one-stage object detection[C]. Proceedings of the IEEE international conference on computer vision. 2019: 9627-9636
- [42] Yang Z, Liu S, Hu H, et al. Reppoints: Point set representation for object detection[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 9657-9666
- [43] Shi S, Guo C, Jiang L, et al. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 10529-10538
- [44] Liu Z, Hu H, Cao Y, et al. A closer look at local aggregation operators in point cloud analysis[C]. European Conference on Computer Vision. Springer, Cham, 2020: 326-342
- [45] Shi S, Wang Z, Shi J, et al. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020
- [46] Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337
- [47] Lang A H, Vora S, Caesar H, et al. Pointpillars: Fast encoders for object detection from point clouds[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 12697-12705
- [48] Wang Y, Chao W L, Garg D, et al. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 8445-8453
- [49] You Y, Wang Y, Chao W L, et al. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving[J]. arXiv preprint arXiv:1906.06310, 2019
- [50] Engelmann F, Stückler J, Leibe B. Joint object pose estimation and shape reconstruction in urban street scenes using 3D shape priors[C]. German Conference on Pattern Recognition. Springer, Cham, 2016: 219-230
- [51] Leventon M E, Grimson W E L, Faugeras O. Statistical shape influence in geodesic active contours[C]. 5th IEEE EMBS International Summer School on Biomedical Imaging, 2002. IEEE, 2002: 8 pp
- [52] Chen X, Ma H, Wan J, et al. Multi-view 3d object detection network for autonomous driving[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 1907-1915
- [53] Qi X, Liao R, Jia J, et al. 3d graph neural networks for rgbd semantic segmentation[C]. Proceedings of the IEEE International Conference on Computer Vision. 2017: 5199-5208

- [54] Bi Y, Chadha A, Abbas A, et al. Graph-based object classification for neuromorphic vision sensing[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 491-501
- [55] Thakur S, Peethambaran J. Dynamic Edge Weights in Graph Neural Networks for 3D Object Detection[J]. arXiv preprint arXiv:2009.08253, 2020
- [56] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. Acm Transactions On Graphics (tog), 2019, 38(5): 1-12
- [57] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017
- [58] Sindagi V A, Zhou Y, Tuzel O. MVX-Net: Multimodal voxelnet for 3D object detection[C]. 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 7276-7282
- [59] Mai N A M, Duthon P, Khoudour L, et al. Sparse LiDAR and Stereo Fusion (SLS-Fusion) for Depth Estimationand 3D Object Detection[J]. arXiv preprint arXiv:2103.03977, 2021
- [60] 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). IEEE, 2018: 1-8
- [61] Wang T H, Hu H N, Lin C H, et al. 3d lidar and stereo fusion using stereo matching network with conditional cost volume normalization[J]. arXiv preprint arXiv:1904.02917, 2019
- [62] Qiu J, Cui Z, Zhang Y, et al. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 3313-3322
- [63] Chang J R, Chen Y S. Pyramid stereo matching network[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 5410-5418
- [64] Fang J, Zhou D, Song X, et al. MapFusion: A General Framework for 3D Object Detection with HDMaps[J]. arXiv preprint arXiv:2103.05929, 2021
- [65] Laina I, Rupprecht C, Belagiannis V, et al. Deeper depth prediction with fully convolutional residual networks[C]. 2016 Fourth international conference on 3D vision (3DV). IEEE, 2016: 239-248
- [66] Engelcke M, Rao D, Wang D Z, et al. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks[C]. 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 1355-1361
- [67] Song S, Lichtenberg S P, Xiao J. Sun rgb-d: A rgb-d scene understanding benchmark suite[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 567-576

- [68] Jaderberg M, Simonyan K, Zisserman A. Spatial transformer networks[J]. Advances in neural information processing systems, 2015, 28: 2017-2025
- [69] Kaul C, Pears N, Manandhar S. Sawnet: A spatially aware deep neural network for 3d point cloud processing[J]. arXiv preprint arXiv:1905.07650, 2019
- [70] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778
- [71] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks[C]. European conference on computer vision. Springer, Cham, 2016: 630-645
- [72] Song S, Xiao J. Deep sliding shapes for amodal 3d object detection in rgb-d images[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 808-816
- [73] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]. Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988
- [74] Chen X, Kundu K, Zhu Y, et al. 3d object proposals using stereo imagery for accurate object class detection[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(5): 1259-1272
- [75] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014
- [76] Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks[J]. Advances in neural information processing systems, 2017, 30: 971-980
- [77] Chen Y, Liu S, Shen X, et al. Fast point r-cnn[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 9775-9784
- [78] Brazil G, Liu X. M3d-rpn: Monocular 3d region proposal network for object detection[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 9287-9296
- [79] Wang Z, Jia K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal. In 2019 IEEE[C]. RSJ International Conference on Intelligent Robots and Systems (IROS). 1742-1749
- [80] Liang M, Yang B, Chen Y, et al. Multi-task multi-sensor fusion for 3d object detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 7345-7353
- [81] Yang Z, Sun Y, Liu S, et al. Std: Sparse-to-dense 3d object detector for point cloud[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 1951-1960

- [82] Shi W, Rajkumar R. Point-gnn: Graph neural network for 3d object detection in a point cloud[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1711-1719
- [83] Liang M, Yang B, Wang S, et al. Deep continuous fusion for multi-sensor 3d object detection[C]. Proceedings of the European Conference on Computer Vision (ECCV). 2018: 641-656
- [84] He C, Zeng H, Huang J, et al. Structure Aware Single-stage 3D Object Detection from Point Cloud[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11873-11882
- [85] Hu Q, Yang B, Xie L, et al. Randla-net: Efficient semantic segmentation of large-scale point clouds[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11108-11117
- [86] Deng J, Shi S, Li P, et al. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection[J]. arXiv preprint arXiv:2012.15712, 2020

## 攻读硕士学位期间取得的成果

### 论文:

- [1] He Q, Wang Z, Zeng H, et al. Stereo RGB and Deeper LIDAR Based Network for 3D Object Detection[J]. submitted to IEEE Transactions on Intelligent Transportation Systems.
- [2] He Q, Wang Z, Zeng H, et al. SVGA-Net: Sparse Voxel-Graph Attention Network for 3D Object Detection from Point Clouds[J]. (under review)

### 专利:

- [1] 王正宁, 何庆东, 赵德明, 等. 一种基于联合热力图的人脸 3D 关键点检测方法[P]. 中国, 发明专利, 201910818457.6, 2019 年 8 月 30 日
- [2] 王正宁, 何庆东, 赵德明, 等. 一种基于分布式热力图的人脸 3D 关键点检测方法[P]. 中国, 发明专利, 201910818437.9, 2019 年 8 月 30 日
- [3] 王正宁, 赵德明, 何庆东, 等. 一种轻量化人脸 3D 关键点检测方法[P]. 中国, 发明专利, 201910818437.9, 2019 年 8 月 30 日
- [4] 王正宁, 吕侠, 何庆东, 等. 一种 3D 目标检测方法[P]. 中国, 发明专利, CN201911354155.4, 2019 年 12 月 25 日
- [5] 王正宁, 赵德明, 何庆东, 等. 一种面向航拍影像的目标跟踪方法[P]. 中国, 发明专利, 201911043274.8, 2019 年 10 月 30 日
- [6] 王正宁, 赵德明, 何庆东, 等. 一种基于时空约束的长时目标跟踪方法[P]. 中国, 发明专利, 201911057813.3, 2019 年 11 月 1 日
- [7] 王正宁, 吕侠, 赵德明, 何庆东, 等. 一种基于数据融合的 3D 目标检测方法[P]. 中国, 发明专利, CN201911354164.3, 2019 年 12 月 25 日
- [8] 王正宁, 曾浩, 潘力立, 何庆东, 等. 一种基于多层次特征混合与注意力机制的目标跟踪方法[P]. 中国, 发明专利, 202010518472.1, 2020 年 6 月 9 日