Multi-modal fusion methods for 3D object detection : A survey

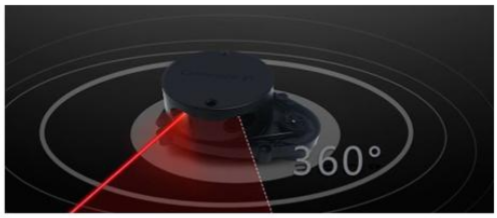
Motivation

丰富数据特征（RGB具有颜色纹理，Point cloud具有深度等结构信息）

Challenges

难点一：传感器视角问题

camera获取到的信息是“小孔成像”原理，是从一个视锥出发获取到的信息，而lidar是在真实的3D世界中获取到的信息。这使得在对同一个object的表征上存在很大的不同。



难点二：数据表征不一样

这个难点也是所有多模态融合都会遇到的问题，对于image信息是dense和规则的，但是对于点云的信息则是稀疏的、无序的。所以在特征层或者输入层做特征融合会由于domain的不同而导致融合定位困难。

难点三：信息融合的难度

从理论上讲，图像信息是dense和规则的，包含了丰富的色彩信息和纹理信息，但是缺点就是由于为二维信息。存在因为远近而存在的sacle问题。相对图像而言，点云的表达为稀疏的，不规则的这也就使得采用传统的CNN感知在点云上直接处理是不可行的。但是点云包含了三维的几何结构和深度信息，这是对3D目标检测更有利的，因此二者信息是存在理论上的互补的。如何融合两者的信息是个挑战。

难点四：时间同步要求较高

Methods

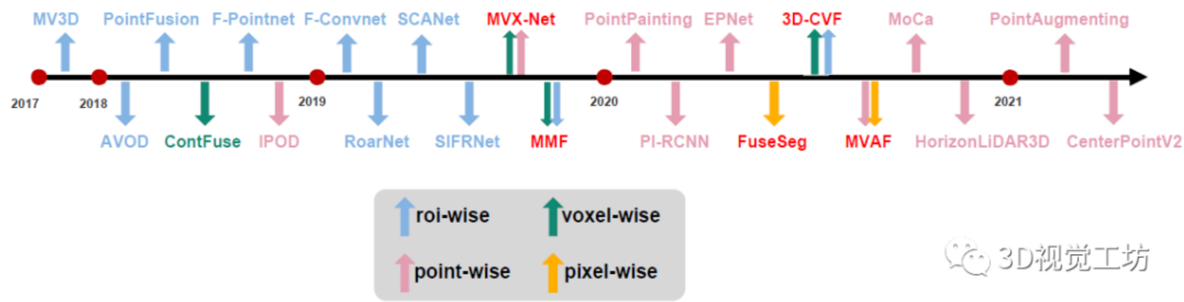


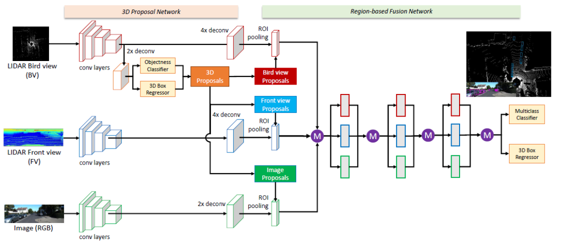
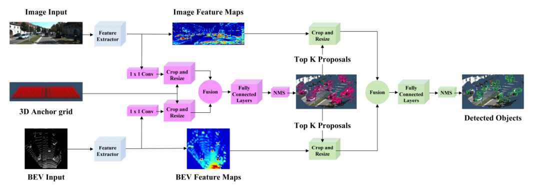
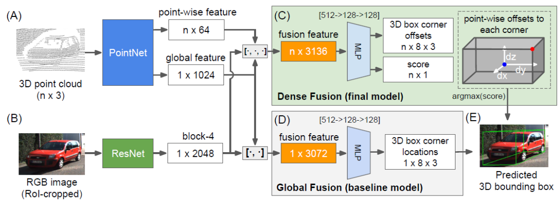
Table 3 Summary of multi-modal 3D detection methods: loc (fusion location), gran (fusion granularity), PCR (point cloud representation), IR (image representation), lat (latency), DS (dataset used for evaluation)

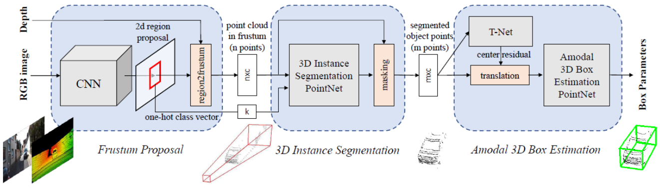
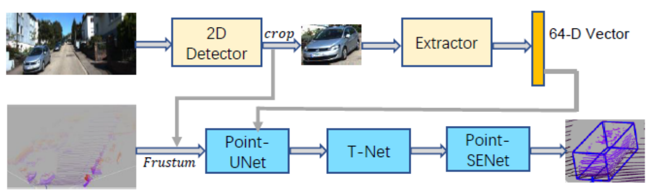
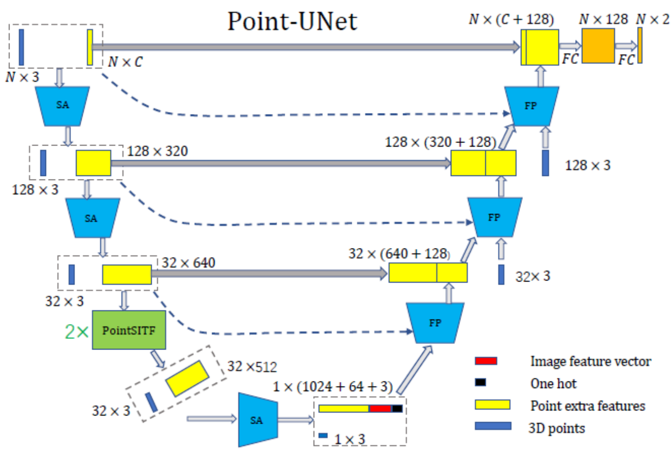
	loc	gran	PCR	IR	Hardware	lat	DS	mAP
MV3D (Chen et al., 2017)	Feature	ROI-wise	View	Feature map	Titan X	0.36s	KITTI	63.63%
AVOD (Ku et al., 2018)	Feature		View	Feature map	Titan XP	0.08s	KITTI	71.76%
PointFusion (Xu et al., 2018)	Feature		Point	Feature map	GTX1080	1.3s	KITTI	63.00%
F-Pointnet (Qi et al., 2018)	Feature		Point	Feature map	GTX1080	0.17s	KITTI	69.79%
F-ConvNet (Wang and Jia, 2019b)	Feature		Point	Feature map	-	0.1s	KITTI	75.50%
RoarNet (Shin et al., 2019)	Feature		Point	Feature map	Titan X	-	KITTI	73.04%
SCANet (Lu et al., 2019)	Feature		View	Feature map	GTX1080	0.09s	KITTI	66.30%
SIFRNet (Zhao et al., 2019)	Feature		Point	Feature map	-	-	KITTI	-
Confuse (Liang et al., 2018)	Feature	Voxel-wise	Voxel	Feature map	GTX1080	0.06s	KITTI	68.78%
IPOD (Yang et al., 2018b)	Feature	Point-wise	Point	mask	-	0.1s	KITTI	72.57%
PointPainting (Vora et al., 2020)	Feature		Point	Mask	GTX1080	0.4s	KITTI	71.70%
PI-RCNN (Xie et al., 2020)	Feature		Point	Feature map	-	0.06s	KITTI	71.70%
EPNet (Huang et al., 2020)	Feature		Point	Feature map	Titan XP	0.1s	KITTI	81.23%
Moca (Zhang et al., 2020a)	Feature		Voxel	Feature map	-	-	nuScenes	66.60%
HorizonLiDAR3D (Ding et al., 2020)	Feature		Voxel	Mask	-	-	Waymo	78.49%
PointAugmenting (Wang et al., 2021)	Feature		Voxel	Feature map	-	-	nuScenes	66.80%
CenterPointV2 (Yin et al., 2021)	Feature		Voxel	Mask	-	-	nuScenes	67.10%
FuseSeg (Sun et al., 2020c)	Feature	Pixel-wise	View	Feature map	-	-	KITTI	-
MMF (Liang et al., 2019)	Feature	Multiple	Voxel & View	Feature map	GTX1080	0.08s	KITTI	77.43%
MVX-Net (Sindagi et al., 2019)	Feature		Voxel	Feature map	-	-	KITTI	72.70%
3D-CVF (Yoo et al., 2020b)	Feature		Voxel	Feature map	-	0.06s	KITTI	80.45%
MVAF (Wang et al., 2020)	Feature		Voxel & View	Pseudo-LiDAR	-	[illegible]	KITTI	[illegible]
CLOCs (Pang et al., 2020)	Decision	-	-	-	-	0.1s	KITTI	82.25%

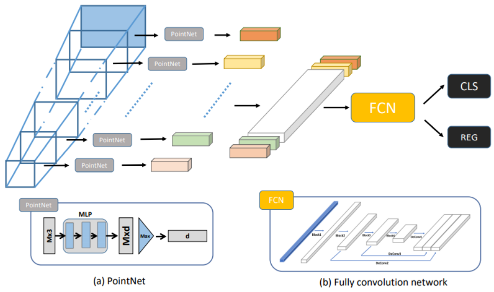
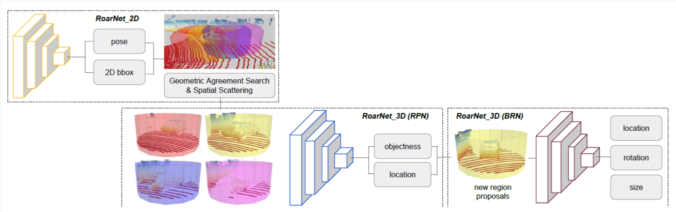
ROI-wise

2D proposal, 3D proposal, frustum

NI(Network input), Src code (source code)

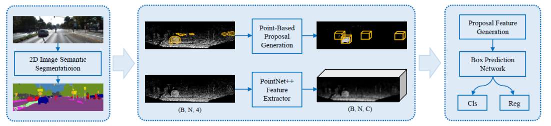
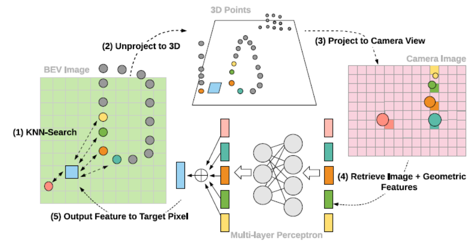
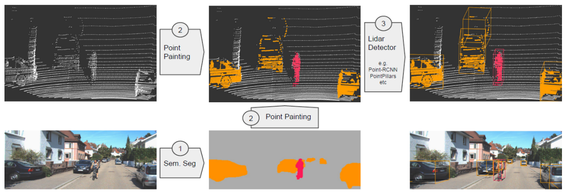
Methods	NI	fusion methods	Src Code
MV3D (CVPR2017)[1]	BV+FV+RGB	two-stage：在BEV上生成3D proposal然后投影到不同views，得到相应模态的特征，进行融合。  Figure 1: **Multi-View 3D object detection network (MV3D):** The network takes the bird's eye view and front view of LIDAR point cloud as well as an image as input. It first generates 3D object proposals from bird's eye view map and project them to three views. A deep fusion network is used to combine region-wise features obtained via ROI pooling for each view. The fused features are used to jointly predict object class and do oriented 3D box regression.	Y
AVOD[2] (IROS2018)	BEV+RGB	(在MV3D方法上进行改进，proposal的生成) two-stage： 1）RGB image和BEV的**全局特征融合** 2）利用融合后的特征进行proposal生成 3）proposal投影到不同模态特征上融合进行detection  Fig. 2: The proposed method's architectural diagram. The feature extractors are shown in **blue**, the region proposal network in **pink**, and the second stage detection network in **green**.	Y
PointFusion[3] (CVPR2018)	Point cloud+RGB	对MV3D方法的改进（投影到FV，BV视角上会导致信息丢失） two-stage（融合的是ROI crop）：直接利用raw point作为输入，且对每个point都融合point-wise feature，global feature，image feature，最后对每个点都预测box和score（无监督）  Figure 2. An overview of the dense PointFusion architecture. PointFusion has two feature extractors: a PointNet variant that processes raw point cloud data (A), and a CNN that extracts visual features from an input image (B). We present two fusion network formulations: a vanilla *global* architecture that directly regresses the box corner locations (D), and a novel *dense* architecture that predicts the spatial offset of each of the 8 corners relative to an input point, as illustrated in (C): for each input point, the network predicts the spatial offset (white arrows) from a corner (red dot) to the input point (blue), and selects the prediction with the highest score as the final prediction (E). 最终效果是和MV3D差不多，没有AVOD好（说明proposal生成也挺重要的)	Y

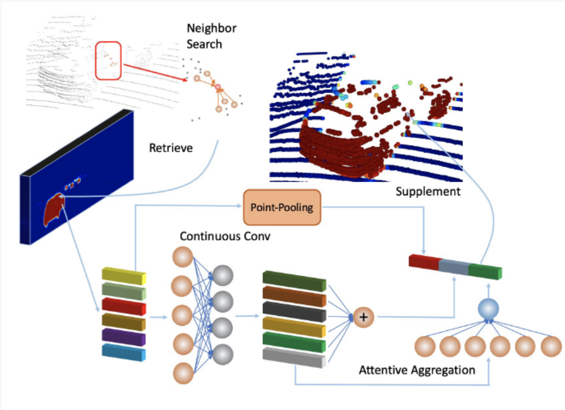
F-PointNet [4] (CVPR2018)	Point+ RGB-D	**首次以2D detection driven 3D**，使用Frustum 缩小搜索空间 two-stage：利用2D检测器得到2D proposal，然后投影到3D空间形成frustum proposal，提取该区域的所有点云送进PointNet++进行3D instance segmentation（进一步缩小proposal的三维空间），最后，利用T-Net对坐标归一，并再次使用PointNet++，回归出物体3D Bouding Box的相关参数。  大大提升了小目标物体检测	Y
SIFRNet [5] (AAAI2019)	Point+RGB	**对F-PointNet的改进**（F-PointNet仅利用了2D proposal来缩小搜索空间，并没有利用到RGB图像的纹理，颜色等信息） two-stage（利用2D detector生成2D proposal）：1）**Point-UNet** (用于3D实例分割，区分出前景点还是背景点) 输入为3D points in the frustum（N*3为点的x,y,z坐标，NC*为对应点的RGB信息和点云的反射强度） 2）**T-Net**（坐标变换） 3）**Point-SENet** （预测3D box，网络输出为（3+4*NS+2*NH）维向量，NS表示size模板数量，NH表示方向角模板数量）  Figure 1: The pipeline of SIFRNet for 3D object detection 融合方法： 1）利用2D detector的结果投射到3D Frustum中，取相应的点云送进后续网络（并使用RGB对点云进行信息增强） 2）对2D detector 检测到的ROI提取特征，得到64-D的vector，融合进网络中，并且还提供2D detection的class information（one-hot编码）  Figure 3: The network architecture of Point-UNet.	N

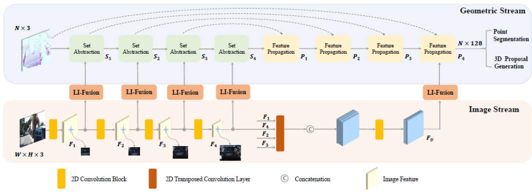
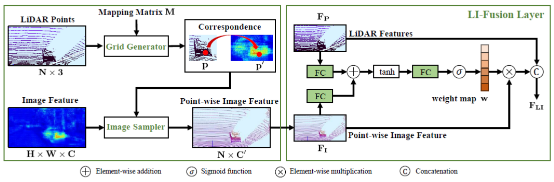
F-ConvNet [6] (IROS2019)	Point+RGB	对F-PointNet的改进(AP_{3D}提升比较大）（改进了点的采样方式，将pixel-wise feature转换成frustum-level feature，提升计算效率。对结果优化避免2D检测器失效）1）利用2D detector得到三维空间中的Frustum，设定Frustum步长s，对Frustum进行空间上的划分 2）使用PointNet对每个Frustum区域进行特征提取，得到Frustum-level特征，然后将这些特征re-formed成2D feature map，送入FCN进行分类和回归 3）优化结果（因为初始的2D region proposa 并不精准），首先对于预测的框乘上1.2扩展系数，然后通过平移旋转来normalize这些3D框内的点，再次送入F-ConvNet进行优化  Fig. 2: The whole framework of our F-ConvNet. We group points and extract features by PointNet from a sequence of frustums, and for 3D box estimation, frustum-level features are re-formed as a 2D feature map for use of our fully convolutional network (FCN) and detection header (CLS and REG). (a) The architecture of PointNet. (b) The architecture of FCN used in Frustum ConvNet for KITTI dataset. Each convolutional layer is followed by Batch Normalization and ReLU nonlinearity. Blue-colored bar in (b) represents the 2D feature map of arrayed frustum-level feature vectors.	Y
RoarNet [7] (IEEE IV2019)	Point+RGB	类似视锥的思想，用了**几何约束的方法** [8] ，且计算2D和3D box的IOU解决同步问题 1) **RoarNet_2D**：输入为RGB image，输出为3D region proposal a. 根据2D box和3D box投影得到的约束关系减少自由度求解，并计算box_{2D}和$box_{3Dproject2D}$的IOU作为configuration score，最终选**得分最高**的作为3D box。 b. 对3D box尺寸进行**扩展**，得到smal box和large box，连接两个box中心，设定固定步长得到一些等间距点，以点为中心设置半径生成圆柱形proposal。 2) **RoarNet_3D**：预测3D bounding box  Fig. 2: Architecture of RoarNet	Y / N

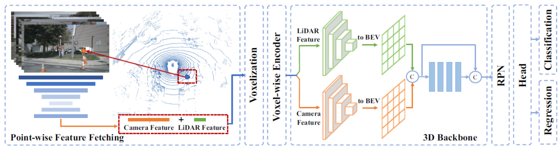
Point-wise

1. **use (2D segmentation score/ segmentation feature/detection feature/raw RGB) to decorate point cloud**
2. **dense and discrete feature convert**
3. **different stream using attention to fuse**

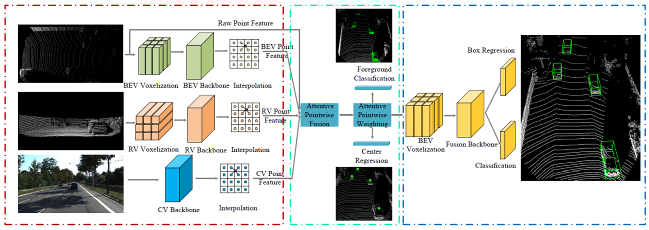
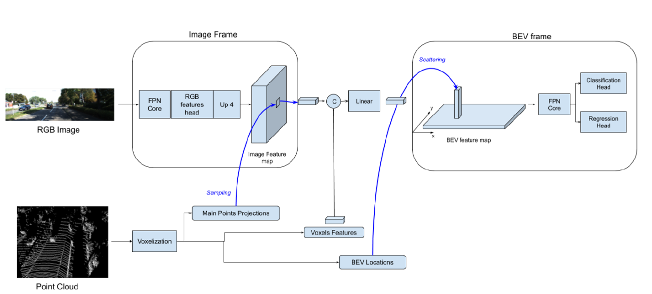
Methods	NI	fusion method	Src Code
IPOD [9] (2018)	Point+RGB	用2D分割，然后投影分割结果到点云来区分每个点是positive还是negative，对**每个positive点**都生成multiple scales, angles and shift的**proposal**，再NMS）  Figure 1. Illustration of our framework. It consists of three different parts. The first is a subsampling network to filter out most background points. The second part is for point-based proposal generation. The third component is the network architecture, which is composed of backbone network, proposal feature generation module and a box prediction network. It classifies and regresses generated proposals.	N
Contfuse [10] (CVPR2018)	BEV+RGB	 Fig. 2: Continuous fusion layer: given a target pixel on BEV image, we first extract K nearest LIDAR points (Step 1); we then project the 3D points onto the camera image plane (Step 2-Step 3); this helps retrieve corresponding image features (Step 4); finally we feed the image feature + continuous geometry offset into a MLP to generate feature for the target pixel (Step 5). BEV和RGB的融合（对每一个BEV location取图片特征，将discrete RGB feature投影到dense BEV上）	N
PointPainting [11] (CVPR2020)	Point+RGB	将激光雷达点云投影到图像平面，得到对应的pixel-wise segmentation scores，得到painted point cloud  	Y

Methods	NI	fusion method	Src Code
PI-RCNN [12] (AAAI2020)	Point+RGB	针对ContFuse做的改进工作（认为之前的融合不够精准，增加了point-pooling和attentive aggregation） 融合部分的**输入**为：3D proposals和2D segmentation 的mask **feature**（使用分割的原因是可以得到全分辨率的feature map） **融合模块point-based attentive contfuse moudle（PACF）**： 1)对*每个3D point*，选择K-nearest个邻域点，并投影到2D feature map上。 2）依次将k+1个点的2D feature map的语义特征和3D几何特征（邻域点到该point的offset）concat。 3）使用attentive continuous convolution融合语义和几何特征。 4）对步骤2）的output进行point-pooling，并和3）的output进行concat。  	N

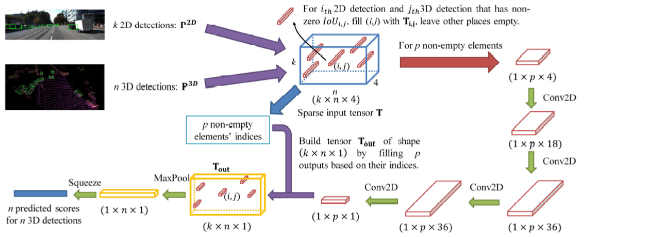
Methods	NI	fusion method	Src Code
EP-Net[13] (ECCV2020)	point+RGB	**motivation：** (1)传感器级联（不同阶段使用不同传感器）导致结果受限于每个传感器，没有使用到不同传感器之间的互补性，有一些融合方法需要生成BEV的会造成信息丢失，或者只是建立了一些粗糙的融合关系。----->本文提出LI-Fusion来解决上面两个问题（Point-wise的方式并且自适应的估计**图像语义特征**的重要性）(2) localiztion 和 classification confidence的不连续性------>本文提出consistency enforing loss  **Fig. 2.** Illustration of the architecture of the two-stream RPN which is composed of a geometric stream and an image stream. We employ several LI-Fusion modules to enhance the LiDAR point features with corresponding semantic image features in multiple scales. N represents the number of LiDAR points. H and W denote the height and width of the input camera image, respectively. 包含geometric stream和image stream两部分，使用多个LI-Fusion模块来增强point feature（使用不同scales的图像语义特征)  **Fig. 3.** Illustration of the LI-Fusion module, which consists of a grid generator, an image sampler, and a LI-Fusion layer. **Step1：将点云投影到图像平面上** $p' = M \times p$, 其中$p(x, y, z)$为点云坐标， $P'(x', y')$为该点在RGB image中对应坐标。 **Step2：得到point-wise image semantic feature** $V^{(p)} = K(F^{N(P')})$, 其中$V^{(p)}$为点p对应的图像特征，$(F^{N(P')}$为采样点P'领域的image feature，K为双线性插值操作。 **Step3：利用LiDAR feature来自适应地估计point-wise image semantic feature的重要性** $W = \sigma(wtanh(uF_P + vF_I))$, 其中，$w, u, v$表示可学习的权重矩阵，$\sigma$表示sigmoid激活函数。首先将LiDAR feature和point-wise feature送入全连接层，使得它们具有相同通道数并相加。然后通过tanh和全连接层，sigmoid激活函数进行压缩，得到权重W map。 对于point-wise image semantic feature乘以权重矩阵然和和LiDAR feature进行concatenation。	Y
CenterPoint V2[14] (CVPR2021)	point+RGB	LossCenterpoint+pointpainting	Y

Methods	NI	fusion method	Src Code
Point-Augmenting[15] (CVPR2021)	Point+RGB	1）提出对于RGB图片而言，颜色纹理特征比2D segmentation score对点云互补作用更大，使用2D object detection网络提取到的CNN feature作为image representation和点云进行融合。并使用**3D sparse convolution**分别对LiDAR 和Camera feature进行卷积操作。 2）在GT-Paste[16] 基础上提**出了数据增强方法。**  Figure 3. PointAugmenting overview. The architecture consists of two stages. (1) Point-wise feature fetching: LiDAR points are projected onto image plane and then appended by the fetched point-wise CNN features. (2) 3D detection: we extend CenterPoint with an additional 3D sparse convolution stream for camera features and fuse different modalities via a simple skip and concatenation approach in BEV maps. 精度要比centerpointV2稍微低一个点这样，初步猜想是因为处理成BEV导致部分信息丢失。	N

Multiple

Methods	NI	fusion method	Src Code
MVAF[17] (2020)	RV+RGB +BEV+ Point	**one-stage**(PI-RCNN, EPNet虽然也使用了注意力机制，但是由于two-stage，计算量很大)  Fig. 1: Overall architecture of MVAF-Net. The overall MVAF-Net consists of three parts: 1) single view feature extraction (SVFE), 2) multi-view feature fusion (MVFF) and 3) fusion feature detection (FFD). In the SVFE part, the raw RGB images and point clouds are processed by a three-stream CNN backbone (CV, BEV and RV backbone) to generate multi-view feature maps, where the point clouds are voxelized in both BEV and RV. In the MVFF part, the multi-view features are adaptively fused with the proposed attentive pointwise fusion module in a pointwise manner. The fused point features are further processed with the proposed attentive pointwise weighting module to reweight the point features and learn structure information. In the FFD part, the fused and reweighted point features are voxelized again and used as input to the fusion backbone for final 3D detection.	N
		1）**SVFE**：分别在BEV，RV，CV上提取特征，生成multi-view features。对raw point clouds中的每一个点，都投影到不同view，得到相应的feature。将raw, BEV, FV, CV point feature concat得到multi-view feature。 2）**MVFF**：将multi-view feature送入APF中，利用注意力机制学习不同view channel 的importance，得到fused feature。再将fused feature送入APW，利用foreground classification和center regression作为监督，reweight下fused point feature。 3）**FFD**：将reweight后的fused point feature送入网络进行检测。	
R-AGNO-RPN[18] (2020)	RGB+Point (Voxel & Point-wise)	 1)将图片送入FPN得到不同scale的features，使用RGB feature head对feature进行merge，并4倍上采样。 2)对点云进行体素化，对每一个voxel，记录voxels feature和BEV location，并将voxel中的点投影到image平面，获取image feature。 3) 将image feature和voxel feature concat，并通过BEV location得到BEV feature map，再次送入FPN进行物体检测	N

Result fusion(late)

Methods	NI	fusion method	Src Code
CLOCS[19] (IROS2020)	2D& 3D box	 Fig. 3: CLOCs Fusion network architecture. First, individual 2D and 3D detection candidates are converted into a set of consistent joint detection candidates (a sparse tensor, the blue box); Then a 2D CNN is used to process the non-empty elements in the sparse input tensor; Finally, this processed tensor is mapped to the desired learning targets, a probability score map, through maxpooling. $p_i^{2D} = \{x_{i1}, y_{i1}, x_{i2}, y_{i2}, s_i^{2D}\}$ $p_i^{3D} = \{h_i, w_i, l_i, x_i, y_i, z_i, \theta_i, s_i^{3D}\}$ $T_{i,j} = \{IoU_{i,j}, s_i^{2D}, s_i^{3D}, d_j\}$	Y

算法落地

Methods	KITTI（3D car)	Hardware	lat	Src Code
EPNet	118	Titan XP	0.1s	Y
CLOCs	135	/	0.1s	Y
F-ConvNet	162	/	0.1s	Y
AVOD	192	Titan XP	0.08s	Y
PointFusion	/	GTX1080	1.3s	unofficial

reference

1. X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017 ↩

2. J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, “Joint 3d proposal generation and object detection from view aggregation",in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ↩

3. D. Xu, D. Anguelov, and A. Jain, “PointFusion: Deep sensorfusion for 3D bounding box estimation,” in CVPR, 2018. ↩

4. C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum PointNets for 3D object detection from RGB-D data,” in CVPR,2018. ↩

5. X. Zhao, Z. Liu, R. Hu, and K. Huang, “3D object detection using scale invariant and feature reweighting networks,” in AAAI,2019. ↩

6. Z. Wang and K. Jia, “Frustum convNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection,” in IROS, 2019. ↩

7. K. Shin, Y. P. Kwon, and M. Tomizuka, “RoarNet: A robust 3D object detection based on region approximation refinement,” in IEEE IV, 2019. ↩

8. A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, “3D Bounding Box Estimation Using Deep Learning and Geometry,” in CVPR, 2017. ↩

9. Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, “IPOD: Intensive point-based object detector for point cloud,” arXiv preprint arXiv:1812.05276, 2018. ↩

10. M. Liang, B. Yang, S. Wang, and R. Urtasun, “Deep continuous fusion for multi-sensor 3D object detection,” in ECCV, 2018. ↩

11. V. Sourabh, L. Alex H., H. Bassam, and B. Oscar, “PointPainting: Sequential fusion for 3D object detection,” in CVPR, 2020. ↩

12. Xie, Liang et al. “PI-RCNN: An Efficient Multi-sensor 3D Object Detector with Point-based Attentive Cont-conv Fusion Module.” *ArXiv* abs/1911.06084 (2020). ↩

13. Huang T., Liu Z., Chen X., Bai X. (2020) EPNet: Enhancing Point Features with Image Semantics for 3D Object Detection. In ECCV 2020. ↩

14. **Center-based 3D Object Detection and Tracking**, Tianwei Yin, Xingyi Zhou, Philipp Krähenbühl, *arXiv technical report (arXiv 2006.11275)* ↩

15. PointAugmenting: Cross-Modal Augmentation for 3D Object Detection, CVPR2021 ↩

16. Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. Sensors, 18(10):3337, 2018. ↩

17. Multi-View Adaptive Fusion Network for 3D Object Detection. 2011.00652.pdf (arxiv.org) ↩

18. R-AGNO-RPN: A LIDAR-Camera Region Deep Network for Resolution-Agnostic Detection. arXiv:2012.05740v1 ↩

19. CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection, in IROS2020. ↩