PA3DNet: 3-D Vehicle Detection With Pseudo Shape Segmentation and Adaptive Camera-LiDAR Fusion

Meiling Wang, Lin Zhao, and Yufeng Yue, *Member, IEEE*

***Abstract*—3-D vehicle detection is a key perception technique in autonomous driving. In this article, a novel 3-D vehicle detection framework that fuses camera images and Light Detection and Ranging (LiDAR) point clouds is proposed, named PA3DNet. The key novelties of PA3DNet are the proposing of a pseudo shape segmentation (PSS) model and an adaptive camera-LiDAR fusion (ACLF) module. The PSS model leverages self-assembled vehicle prototypes to learn shape-aware vehicle features. In order to achieve the adaptive fusion between visual semantics and LiDAR point features, learnable weight parameters are developed in the ACLF module to formulate an implicit complementarity between the two modalities. Extensive experiments on the widely used autonomous driving KITTI dataset demonstrate that PA3DNet achieves competitive accuracy when compared to advanced methods. It achieves 5.37% higher average precision (AP) on easy difficulty of 30–50 m and 9.67% higher AP on moderate difficulty of >50 m.**

***Index Terms*—3-D object detection, autonomous driving, multimodal fusion.**

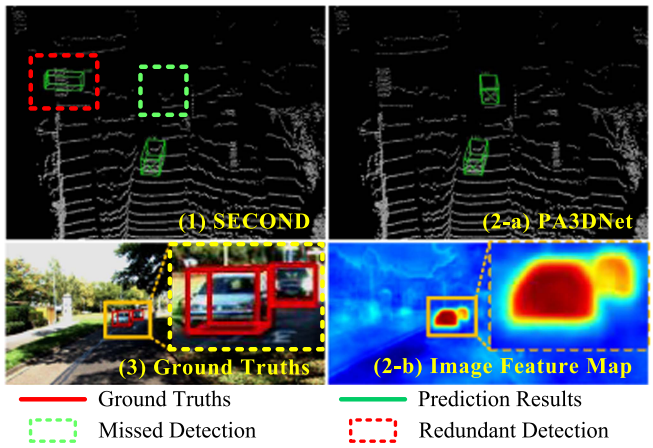

Fig. 1. Example of 3-D vehicle detection in an autonomous driving scene. First row: LiDAR point clouds with 3-D detection boxes. Second row: (3) RGB image with 3-D ground truths, as well as (2-b) image feature map from the proposed PSS model.

I. Introduction

With the growing development of the transportation industry, the safety of intelligent vehicles, especially self-driving cars, has become increasingly prominent [1]. In general, safe autonomous navigation requires robust and reliable 3-D perception technology. 3-D vehicle detection, which utilizes camera or Light Detection and Ranging (LiDAR) sensors to recognize surrounding vehicles in on-road environments, is considered a critical perception capability in the field of intelligent transportation systems. As intelligent vehicles are required to drive more safely and reduce traffic accidents in complex and dynamic environments, detecting surrounding 3-D vehicles can help them maintain a safe following distance, avoid potentially dangerous collisions, and provide essential environmental information for subsequent decision making and vehicle control.

Benefiting from the powerful feature extraction capability of deep convolutional neural networks, 3-D vehicle detection based on single-modal data from cameras or LiDAR sensors has been studied recently [3], [4]. Camera RGB images represent environments in a perspective view with rich colors and dense textures; however, existing camera-only 3-D vehicle detection methods [5], [6] suffer from performance degradation due to the lack of reliable long-range depth measurements, achieving only approximately 10% mean average precision (AP) on the official KITTI test dataset. On the other hand, precise ranging information from laser scans brings benefits to LiDAR-only methods. However, 3-D vehicle shape information from LiDAR point clouds tends to be severely incomplete in occluded or long-range scenes, inevitably leading to loss of geometric context and occasional erroneous detection results. As shown in Fig. 1-(1), the LiDAR-only method [8] suffers from missed detection (false negative) and redundant detection (false positive), while the proposed multimodal approach [in Fig. 1-(2-a,b)] fuses visual shape features to achieve better performance. Recently, several image-based algorithms have explored the fine-grained contextual information in images. Abdel-Basset et al. [2] utilized

Manuscript received 6 September 2022; revised 13 December 2022; accepted 15 January 2023. Date of publication 1 February 2023; date of current version 19 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62003039 and Grant 62233002 and in part by the CAST Program under Grant YESS20200126. Paper no. TII-22-3804. *(Corresponding author: Yufeng Yue.)*

The authors are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: wangml@bit.edu.cn; zhaolin_edu@qq.com; yueyufeng@bit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3241585.

Digital Object Identifier 10.1109/TII.2023.3241585

a metaheuristic balancing approach to find the optimal threshold for grayscale images. Haris and Glowacz [1] studied the feature prediction capability of the decoder in the lane segmentation network. Moreover, the authors of [7] further examined that the combination of camera RGB information and LiDAR depth information has better performance than using only a single modality in end-to-end intelligent driving. Therefore, fusing sensor data from two complementary modalities can fully exploit dense visual information and accurate LiDAR ranging measurements, thereby improving the robustness of 3-D vehicle detection in challenging occlusion and long-range cases. To this end, this article proposes the multimodal fusion algorithm between RGB images and point clouds as a more desirable solution for 3-D vehicle detection.

As summarized above, multimodal 3-D vehicle detection can complement the strengths of both modalities. Existing multimodal methods [9], [10], [11] utilize image cues obtained from two-stream extractors to compensate point features, but are limited to low-level visual feature fusion at the early stage. To take full advantage of the rich visual information, Pointpainting [12], PI-RCNN [13], and Sem-Aug [14] directly extracted high-level image features from pretrained semantic segmentation models or supervised by 2-D segmentation autolabeling. While effective, these methods are constrained by the generalization ability of pretrained models or rely on external 3-D priors to provide autolabeling references. In the absence of image annotation supervision or external 3-D priors, visual semantic segmentation networks may fail to learn reliable image features, resulting in degraded detection performance. Therefore, the first challenge for multimodal 3-D vehicle detection is to formulate a practical visual model that bridges the gap between high-quality image semantic learning and limited data supervision.

Compared with single-modal methods that learn from viewpoint-consistent sensor data, the cross-view association between LiDAR points and camera images poses another challenge. To localize 3-D objects in large-scale point cloud scenes, [16] generates 3-D LiDAR frustums from image-based region proposals to reduce the 3-D search space. In view of the correspondence between cross-view feature learning, MV3D [17], AVOD [18] and MVX-Net [19] aggregate region-based image features through pooling layers and then append them to point features to jointly learn cross-view information. The features of the two modalities are only fused at concatenation level in these methods, whereas the fusion layers cannot determine which image features are complementary for the corresponding LiDAR points. As a result, the second challenge is to carefully design a camera-LiDAR fusion layer that adaptively modulates the complementary weights between the two modalities.

The above challenges motivates this article to propose a novel multimodal 3-D vehicle detection framework, named PA3DNet, which contains a pseudo shape segmentation (PSS) model and an adaptive camera-LiDAR fusion (ACLF) module. Specifically, the PSS model self-assembles vehicle point cloud prototypes formed from full 3-D shapes without external 3-D prior models, and projects them as pseudo vehicle shapes on the image plane. Since the generated shapes approximately represent 2-D full contours, we exploit them to supervise a shape semantic segmentation network to learn visual shape features of vehicles. In the ACLF module, bilinear interpolation sampling is first adopted to fetch point-wise visual features at continuous pixel coordinates from a high-level image feature map. A camera-LiDAR fusion layer with learnable weight parameters is then incorporated, which formulates the contributions of the obtained image semantic information and exploits them to adaptively enhance LiDAR point features. The main contributions of PA3DNet can be summarized in threefold.

1) A PSS model utilizing self-assembled vehicle prototypes to generate 2-D pseudo shapes is proposed, which can learn visual shape features of vehicles.
2) The proposed ACLF module formulates the concatenation between image semantics and LiDAR point features via learnable weight variables, enabling adaptive camera-LiDAR feature fusion.
3) Extensive quantitative and qualitative experiments on a widely used autonomous driving KITTI dataset demonstrate that PA3DNet achieves competitive performance compared to advanced baseline methods.

The rest of this article is organized as follows. Section II discusses recent related work. Section III demonstrates the pipeline of PA3DNet. Section IV shows the qualitative and quantitative experiments on the on-road driving scenarios. Finally, Section V concludes this article.

II. Related Work

The existing 3-D vehicle detection methods are divided into three main modes: camera-only, LiDAR-only, and multimodal-based. This section mainly revisits LiDAR-only and multimodal methods.

A. LiDAR-Only 3-D Vehicle Detection

Recent LiDAR-only methods are mainly divided into two categories: point-based and voxel-based. Point-based methods [15], [20] learn local region features by aggregating key point features or employing a transformer-based backbone. 3DSSD [21] incorporates a refined box prediction subnetwork into a point-based 3-D object detector. These methods, however, suffer from costly computation of point feature encoding and have difficulty extending to large-scale tasks. Instead, voxel-based methods [4], [22] convert sparse LiDAR point clouds to regular voxels, and leverage 3-D sparse convolutional layers [8] to encode 3-D voxel features. SegVoxelNet [23] employs semantic segmentation masks from bird's-eye-view (BEV) to provide contextual information and actively guide 3-D vehicle detection. In PV-RCNN [24], a voxel-keypoint 3-D feature encoding module is proposed to enrich representative point cloud features. He et al. [30] and Ning et al. [31] exploited multiview representation and attention mechanism to extract more neighborhood and contextual information respectively. Although voxel-based feature extractors are effective, these methods are constrained by the inherent low-occupancy nature of LiDAR point clouds, making them suboptimal for complex 3-D scenes.

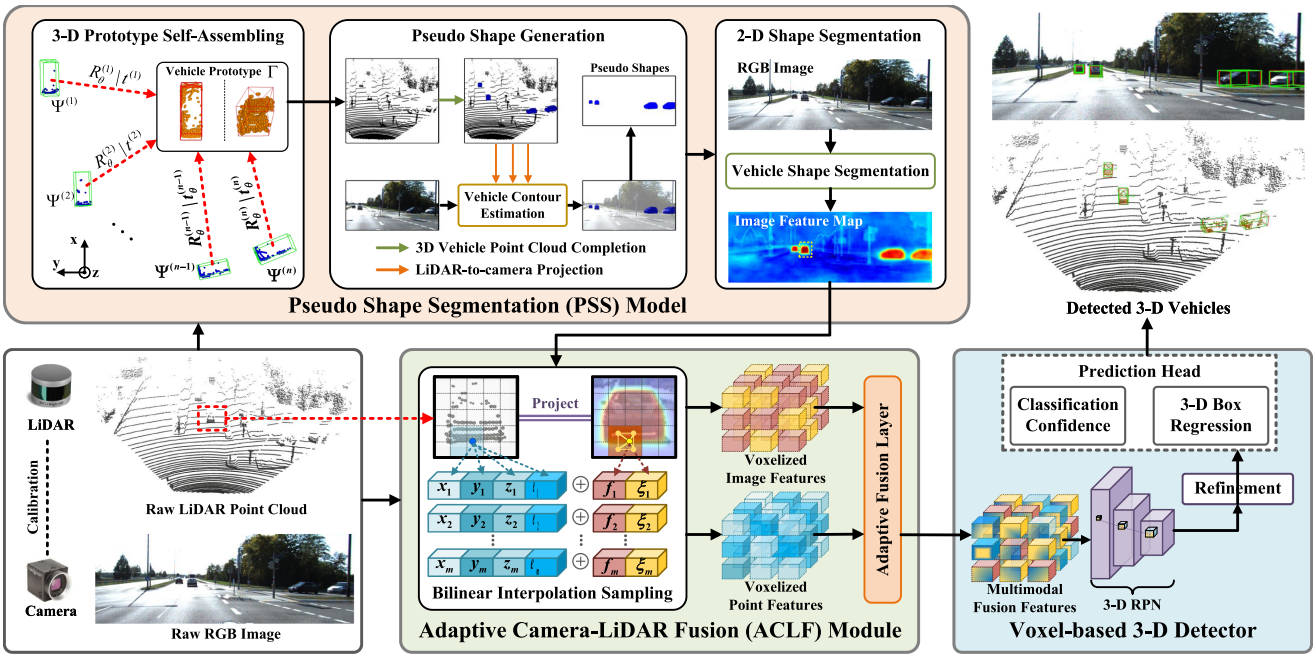

Fig. 2. Overview of PA3DNet, where the predicted 3-D detection results are shown in both image and LiDAR point cloud (⊕: concatenation operations).

B. Multimodal 3-D Vehicle Detection

Early multimodal approaches [16], [17], [18], [19] incorporate region-based fusion strategies into 3-D vehicle detection networks. Qi et al. [16] utilized region proposals from image-based object detectors to generate 3-D bounding frustums. In AVOD [18], a multiview region proposal network (RPN) is combined with a region-based fusion module to produce high-quality 3-D region proposals. Two-branch extractors with shared camera-LiDAR features were proposed in MV3D [17], followed by fusion-based RPN strategies to enhance the small-object prediction. However, these region-based late fusion methods tend to fuse limited visual information through heavy network structures. The recent SFD [25] utilizes attention mechanisms to fuse sparse LiDAR points and dense RGB point clouds. Despite several improvements, it relies on robust depth completion and is difficult to generalize in complex scenes.

To further explore the point-level correspondence between cameras and LiDAR sensors, Yoo et al. [10] and Wen and Jo [11] developed a cross-view feature fusion module to interpolate image semantics into dense BEV representations. MVXNet [19] extends the LiDAR-only detector with an additional camera branch, where the image features are fetched from a pretrained visual-based network. EPNet [9] and Three-Attn [11] incorporate the two-stream feature extractor with multimodal fusion attention modules, but only fuse low-level visual semantic features. By extracting image features from pretrained semantic segmentation networks, PointPainting [12] and PI-RCNN [13] directly decorate raw LiDAR point clouds with point-level image semantics, while Sem-Aug [14] leverages 2-D segmentation autolabeling to provide supervision for their segmentation subnetwork. However, these methods require external data assistance, or simply augment the raw LiDAR point clouds. More recently, Transfusion [34] and BEVFusion [35] investigate camera-LiDAR fusion in BEV representation space, while their frameworks are complicated and require 360° field of view (FOV) from multiple cameras.

III. Proposed Approach

The proposed PA3DNet is modularly designed and consists of three parts: a PSS model, an ACLF module, and a voxel-based 3-D detector, as shown in Fig. 2. To start, the PSS model assembles vehicle prototypes with full 3-D shapes and projects them onto the image plane to generate 2-D pseudo shapes. These pseudo shapes describe the approximate 2-D contours of vehicles. Consequently, the PSS model utilizes pseudo shapes as image labels to learn vehicle segmentation features without manual labeling. Then, the ACLF module extracts point-wise image semantics from high-level segmentation feature maps and incorporates them with LiDAR points in adaptive fusion layers. Finally, the fused multimodal features are fed into the voxel-based 3-D detector to predict 3-D vehicles.

A. Multisensor Configurations

1) Multimodal Inputs: The LiDAR sensor employs scanning laser beams to generate a 3-D representation of the environment, which is typically described as an unordered set of 3-D points. Denote $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ as a LiDAR point cloud data, where N is the number of 3-D points and $\mathbf{p} = [x, y, z]^T \in \mathbb{R}^3$ is the 3-D coordinate. In contrast, the camera image describes a perspective view of the environment and is represented by compact and ordered 2-D grids (also known as pixels). The input RGB image is denoted as $\mathcal{G} \in \mathbb{R}^{H \times W \times C}$ and the 2-D grid coordinates of each pixel on the image plane is denoted

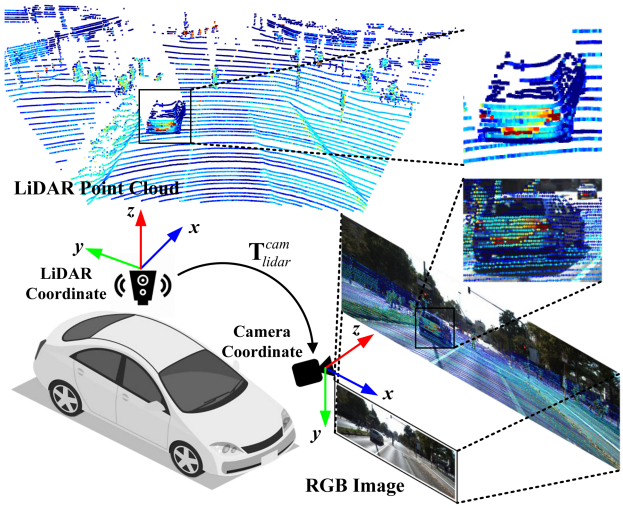

Fig. 3. Measurements, coordinate systems and coordinate transformation of LiDAR and camera sensors.

as $\mathbf{g} = [u, v]^T \in \mathbb{R}^2$, where H, W, and $C = 3$ are the height, width, and number of channels of the RGB image, respectively.

2) Camera-LiDAR Projection: Fig. 3 shows the measurements of the LiDAR and camera sensors along with the coordinate transformation between the two modalities. Denote $\mathbf{T}_{\text{lidar}}^{\text{cam}} \in \mathbb{R}^{4\times4}$ as the calibrated coordinate transformation matrix of the LiDAR coordinate system (LCS) with respect to the camera coordinate system, $\mathbf{K} \in \mathbb{R}^{3\times3}$ as the camera intrinsic matrix. The mapping for projecting a 3-D LiDAR point $\mathbf{p}$ onto the image plane is represented as

$$z_q \mathbf{r} = \mathbf{K}\mathbf{T}_{\text{lidar}}^{\text{cam}}\mathbf{p} \tag{1}$$

where z_q is the depth of the LiDAR point in the camera coordinate system, $\mathbf{r} = [r_x, r_y]^T$ denotes the 2-D coordinate of the projected point, and (1) implicitly transforms between homogeneous coordinates and nonhomogeneous coordinates.

B. PSS

Image semantics from well-trained semantic segmentation networks can provide fine-grained compensation for camera-LiDAR fusion. Compared to labeling-friendly 3-D bounding boxes, semantic segmentation requires collecting pixel-level classifications, which is costly and time-consuming. Therefore, we propose PSS, which utilizes pseudo shapes transformed from 3-D vehicle prototypes to learn fine-grained image semantic features. The proposed PSS consists of three components: (a) 3-D prototype self-assembling, (b) pseudo shape generation, and (c) 2-D shape segmentation. The main idea of (a) and (b) is to automatically generate image segmentation labels from annotations of 3-D LiDAR point clouds.

1) 3-D Prototype Self-Assembling: Camera-LiDAR projection makes it possible to map dense vehicle point cloud as a 2-D vehicle shape on the image plane. However, since LiDAR point clouds become scattered and sparse with increasing distance, it is inevitable that distant vehicles have low 3-D point occupancy, covering only part of the 3-D vehicle's shape. To perform point cloud completion for 3-D vehicles, we adopt a strategy of inserting 3-D vehicle prototypes within each 3-D bounding box. The prototype is obtained by collecting sparse 3-D vehicle points on the specified train dataset and assembling them into a dense point cloud model with complete 3-D vehicle shape. Denote a 3-D vehicle bounding box label as $\mathbf{b} = [c_x, c_y, c_z, l, w, h, \theta]^T$, where $[c_x, c_y, c_z]^T$ is the 3-D position of the center point, θ denotes the vehicle's orientation angle around z-axis, l, w, and h denote the length, width, and height of the 3-D bounding box, respectively. To query the 3-D LiDAR points belonging to the vehicle label $\mathbf{b}$, we first transform raw LiDAR point cloud from the LCS to the local vehicle coordinate system (VCS), whose origin is the center point of $\mathbf{b}$:

$$\mathcal{P}^V = \{\mathbf{q}_i \mid \mathbf{q}_i = \mathbf{R}_\theta(\mathbf{p}_i - \mathbf{t}), \mathbf{p}_i \in \mathcal{P}\} \tag{2}$$

where $\mathbf{R}_\theta \in \mathbb{R}^{3\times3}$ is the rotation matrix converted by the orientation angle θ, $\mathbf{t} = [c_x, c_y, c_z]^T \in \mathbb{R}^3$ denotes the translation vector between LCS and VCS, and $\mathcal{P}^V$ represents the LiDAR point cloud in VCS. Then, the 3-D vehicle points within $\mathbf{b}$ are denoted as $\Psi = \{\mathbf{q}_i \mid x_i \in [-\frac{l}{2}, \frac{l}{2}], y_i \in [-\frac{w}{2}, \frac{w}{2}], z_i \in [-\frac{h}{2}, \frac{h}{2}], \mathbf{q}_i \in \mathcal{P}^V\}$. The 3-D coordinates of these vehicle points are divided by the dimensions on each axis to normalize to unit length:

$$\hat{\Psi} = \left\{\hat{\mathbf{q}}_i | \hat{x}_i = \frac{x_i}{l}, \hat{y}_i = \frac{y_i}{w}, \hat{z}_i = \frac{z_i}{h}, \mathbf{q}_i \in \Psi\right\}. \tag{3}$$

Thus, each normalized $\hat{\Psi}$ is aligned within the same 3-D bounding box dimensions and can be assembled into a 3-D vehicle prototype with full shape. Also, we adopt voxel filtering[1] to downsample the dense vehicle point cloud model and remove outliers.

2) Pseudo Shape Generation: The assembled vehicle prototype is denoted as $\Gamma = \{\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_n\}$, where n is the number of 3-D points. Generally, Γ is a statistical point cloud model of full vehicle shape in the specified dataset. Given a sparse LiDAR point cloud $\mathcal{P}_{\mathcal{B}=\{\mathbf{b}_1,\ldots,\mathbf{b}_K\}}$ with K 3-D vehicle labels, the dimensions of each 3-D vehicle prototype are scaled to fit the size of the corresponding 3-D bounding box label before vehicle point completion:

$$\mathcal{O}_k = \{\mathbf{q}_i | x_i = \hat{x}_i l_k, y_i = \hat{y}_i w_k, z_i = \hat{z}_i h_k, \hat{\mathbf{q}}_i \in \Gamma\} \tag{4}$$

where $[l_k, w_k, h_k]$ is the box size of the kth 3-D vehicle label $\mathbf{b}_k$. With the proposed vehicle point completion strategy, we are able to reconstruct 3-D vehicle shapes in sparse LiDAR point cloud.

Enforcing the LiDAR-to-camera projection allows for establishing correspondences between 3-D LiDAR points and 2-D image pixels. When camera and LiDAR sensors are accurately calibrated, a 3-D to 2-D shape constraint exists to convert the complete 3-D vehicle model into the 2-D shape. Since the assembled vehicle prototype contains dense and complete 3-D vehicle points, we directly project it onto the image plane and generate a dense vehicle pixel set. Convex hull[2] is then exploited to estimate the contour curve of dense vehicle pixels. In general,

[1][Online]. Available: https://github.com/strawlab/python-pcl
[2][Online]. Available: https://github.com/opencv/opencv

a convex hull defines the set of all convex combinations of points in the given subset of the Euclidean space. Therefore, the estimated contour curve can form a pseudo vehicle shape that surrounds all the dense vehicle pixels from the prototype projection. We project the assembled 3-D vehicle prototypes onto the 2-D image plane in order of distance from far to near and generate pseudo shapes for all 3-D vehicle labels.

3) 2-D Shape Segmentation: The inner regions enclosed by the projected pseudo shapes are considered as approximate representations of complete 2-D vehicle shapes. PA3DNet utilizes pseudo shapes to provide supervision for learning high-quality image semantics without extra image annotations. To obtain robust image semantics from RGB images of on-road driving environments, HMANet [27], a multiscale semantic segmentation network, is exploited to learn pseudo vehicle shape segmentation. Specifically, we extract the last prediction feature map from the final stage of HMANet, just before being fed into the softmax layer. The prediction feature map is of the same resolution as the input RGB image and represents the highest-level shape features in the semantic segmentation network. In summary, the proposed PSS model leverages annotations of 3-D LiDAR point clouds to generate pseudo image segmentation labels, which are then utilized to supervise the learning of a shape segmentation network. It reduces the dependence of our multimodal fusion approach on different sensor annotations.

C. ACLF

1) Bilinear Interpolation Sampling: Denote the prediction feature map as $\mathcal{I} \in \mathbb{R}^{C\times H\times W}$, where $C = 2$ is the number of pixel classifications (i.e., vehicle and background). Given a 3-D LiDAR point $\mathbf{p}$, its 2-D projected point $\mathbf{r} = [r_x, r_y]^T$ falls within a grid coordinate of $\mathcal{I}$. Denote $\mathbf{g}^r = [\lfloor r_x \rceil, \lfloor r_y \rceil]^T$ as the grid coordinate corresponding to $\mathbf{r}$, where $\lfloor\cdot\rceil$ denotes the rounding function. To alleviate the floating-point deviation between the 2-D projected point coordinate $\mathbf{r}$ and the correspondence grid coordinate $\mathbf{g}^r$, PA3DNet utilizes bilinear interpolation sampling to extract point-wise semantic features at continuous image coordinates:

$$\mathbf{f}^p = \sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{V}} \underbrace{(1-|r_x-i|)}_{\text{weight in } x \text{ axis}} \underbrace{(1-|r_y-j|)}_{\text{weight in } y \text{ axis}} \mathcal{I}_{[i,j]} \tag{5}$$
$$\text{where } \mathcal{U} = \{\lfloor r_x \rfloor, \lceil r_x \rceil\}, \mathcal{V} = \{\lfloor r_y \rfloor, \lceil r_y \rceil\}$$

where $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ represent the floor and ceiling functions, respectively, i and j denote the neighboring pixel coordinates of the sampling position $\mathbf{r}$, and $\mathbf{f}^p = [f, \xi]^T$ is the binary segmentation feature appended to the 3-D LiDAR point $\mathbf{p}$. When the sampling position exceeds the image grid resolution, its corresponding segmentation features are filled with zeros. Through bilinear interpolation sampling, the raw LiDAR point cloud $\mathcal{P}$ is projected onto the full-size prediction feature map to obtain a semantic feature set, denoted as $\mathcal{F} = \{\mathbf{f}^{\mathbf{p}_1}, \mathbf{f}^{\mathbf{p}_2}, \ldots, \mathbf{f}^{\mathbf{p}_N}\}$.

2) Adaptive Fusion Layer: Denote the input LiDAR point cloud as $\mathcal{P}^g = \{\mathbf{p}_i^g = [x_i, y_i, z_i, e_i]^T \in \mathbb{R}^4\}_{i=1,2,\ldots,N}$, where $\mathbf{p}_i^g$ contains the 3-D geometric coordinates of the ith LiDAR point, and e_i denotes its reflection intensity. Let $\mathcal{P}^s = \{\mathbf{p}_i^s = [x_i, y_i, z_i, f_i, \xi_i]^T \in \mathbb{R}^5\}_{i=1,2,\ldots,N}$ denote the input semantic point cloud, where $\mathbf{p}_i^s$ contains the 3-D position of the ith semantic point and its appended image semantics. First, we convert both the input LiDAR point cloud $\mathcal{P}^g$ and the semantic point cloud data $\mathcal{P}^s$ to voxel representations $\mathcal{V}^g$ and $\mathcal{V}^s$, respectively, and utilize two 3-D sparse convolution layers [8], one to extract LiDAR geometric features from $\mathcal{V}^g$ and the other to obtain image semantic features from $\mathcal{V}^s$. The voxelized LiDAR features and camera features are then fed into an adaptive fusion layer, to fuse the rich geometric-semantic dependencies in 3-D space and refine voxel features.

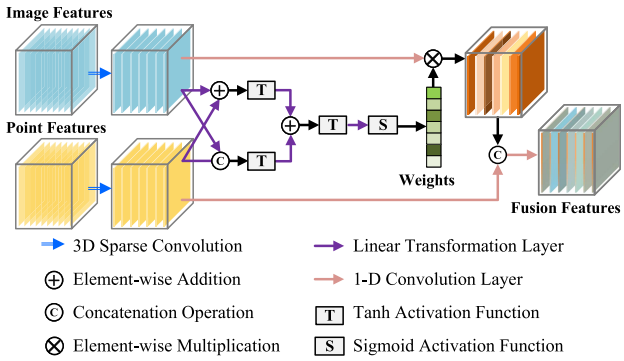

Fig. 4. Details of the ACLF module. Adaptive weight parameters are learned to fuse features from two modalities.

As shown in Fig. 4. the adaptive fusion layer first utilizes 3-D sparse convolutions with the same output channels to process camera features and LiDAR features separately, making the feature channels consistent for both modalities. Through this way, voxelized camera and LiDAR features located in 3-D space share the same sparse voxel coordinates and number of voxels, allowing features from both modalities to be fused in a voxel-wise manner. However, directly concatenating the features of the two modalities cannot achieve flexible associations between useful image semantics and the corresponding LiDAR point geometries. To this end, a learnable sparse weight parameter is embedded in the camera-LiDAR layer to improve the adaptive fusion between the two modalities.

Let $\mathbf{F}_g \in \mathbb{R}^{M\times D}$ and $\mathbf{F}_s \in \mathbb{R}^{M\times D}$ denote the geometric and semantic features of the input nonempty sparse voxels, respectively, where M is the number of nonempty voxels, and D denotes the channel of nonempty voxel features. First, $\mathbf{F}^g$ and $\mathbf{F}^s$ are fed to linear transformation layers, respectively:

$$\mathbf{F}_{g^*} = \mathbf{F}_g\mathbf{w}_g^T + \mathbf{b}_g \tag{6}$$

$$\mathbf{F}_{s^*} = \mathbf{F}_s\mathbf{w}_s^T + \mathbf{b}_s \tag{7}$$

where $\mathbf{w}_{s(g)} \in \mathbb{R}^{D\times D}$ and $\mathbf{b}_{s(g)} \in \mathbb{R}^{D\times 1}$ denote the weights and biases of the linear transformation layers, respectively. The transformed features $\mathbf{F}_{g^*}$ and $\mathbf{F}_{s^*}$ are merged via the element-wise addition operation to form multimodal fusion features:

$$\mathbf{F}_{sg1} = \mathrm{LN}_{\mathbf{w}_1,\mathbf{b}_1}(\tanh(\mathbf{F}_{g^*} + \mathbf{F}_{s^*})) \tag{8}$$

where $\tanh(\cdot)$ denotes the element-wise hyperbolic tangent activation function, and $\mathrm{LN}_{\mathbf{w}_1,\mathbf{b}_1} : \mathbb{R}^{M\times D} \to \mathbb{R}^{M\times D}$ represents a linear transformation layer with learnable parameters $(\mathbf{w}_1, \mathbf{b}_1)$.

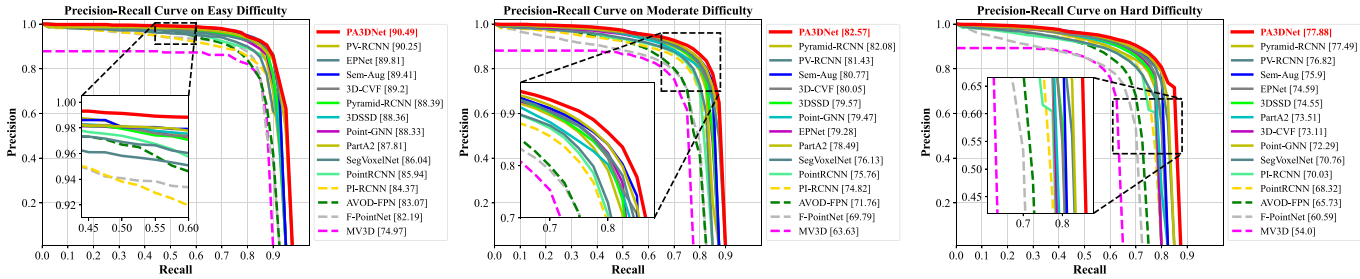

Fig. 5. Precision recall curves of 3-D vehicle detection methods in the official KITTI object detection evaluation test dataset. The performance of PA3DNet compared to state-of-the-art methods at three detection difficulty levels is reported separately.

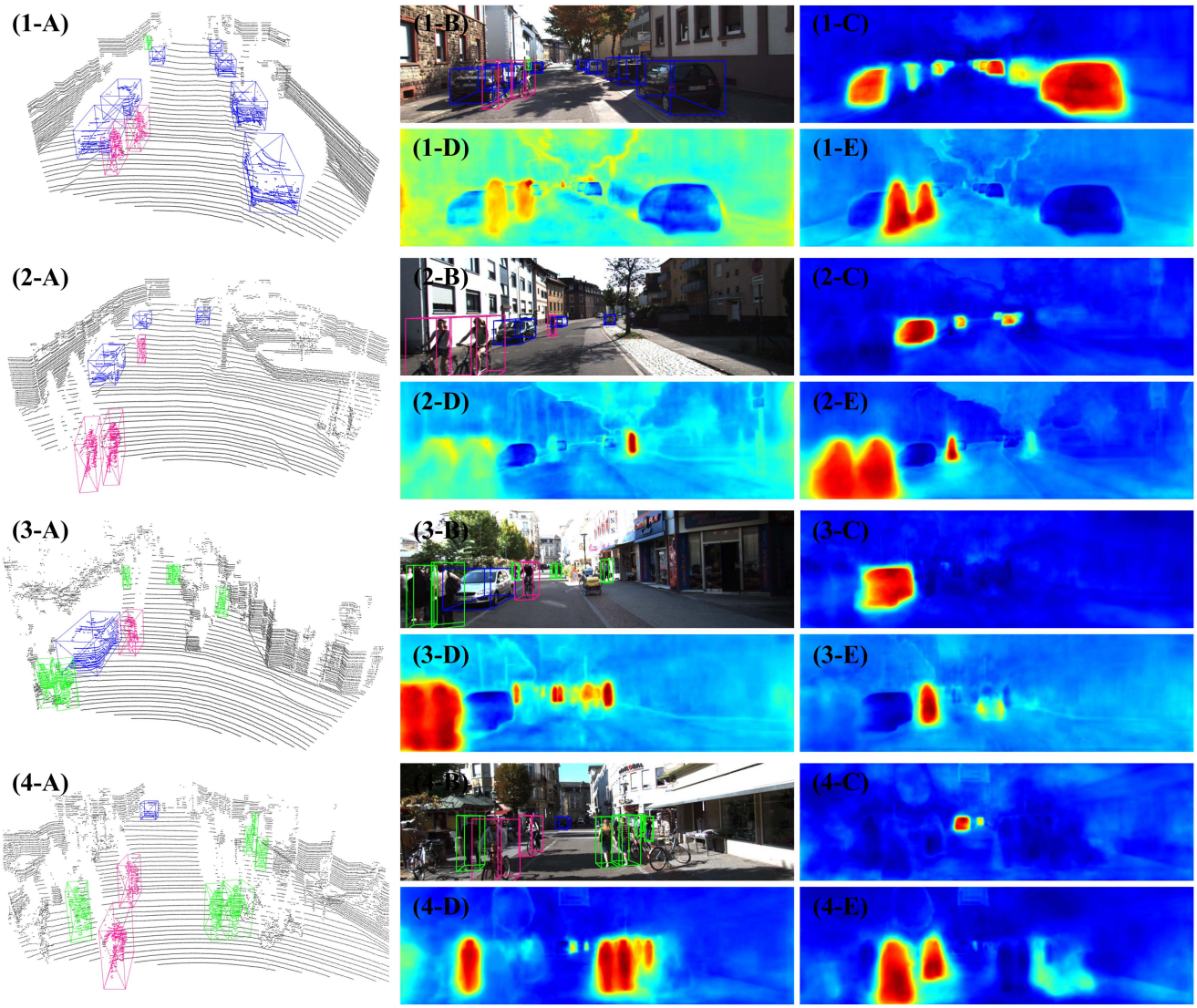

Fig. 6. Visualization of multiclass 3-D object detection results on the KITTI dataset. (1-A to 4-A): LiDAR point cloud with 3-D detection boxes, where *Car*, *Pedestrian*, and *Cyclist* classes are shown in blue, green, and pink, respectively. (1-B to 4-B): RGB images with 3-D bounding boxes. (1-C, 1-D, 1-E to 4-C, 4-D, 4E): semantic image feature maps of *Car*, *Pedestrian*, and *Cyclist* classes, respectively.

Considering that each sparse voxel contains both multidimensional geometric and semantic features, an element-wise concatenation operation is introduced to extend the feature channels of sparse voxels:

$$\mathbf{F}_{sg2} = \mathrm{LN}_{\mathbf{w}_2,\mathbf{b}_2}(\tanh(\mathbf{F}_{g^*} \oplus \mathbf{F}_{s^*})) \tag{9}$$

where $\oplus$ represents the feature channel concatenation operation and $\mathrm{LN}_{\mathbf{w}_2,\mathbf{b}_2} : \mathbb{R}^{M\times 2D} \rightarrow \mathbb{R}^{M\times D}$ denotes another linear transformation layer that outputs the same feature channel as $\mathbf{F}_{sg1}$. The merged $\mathbf{F}_{sg1}$ emphasizes the spatial position information of sparse voxels, while the concatenated $\mathbf{F}_{sg2}$ focuses on the channel information of each sparse voxel feature. We exploit

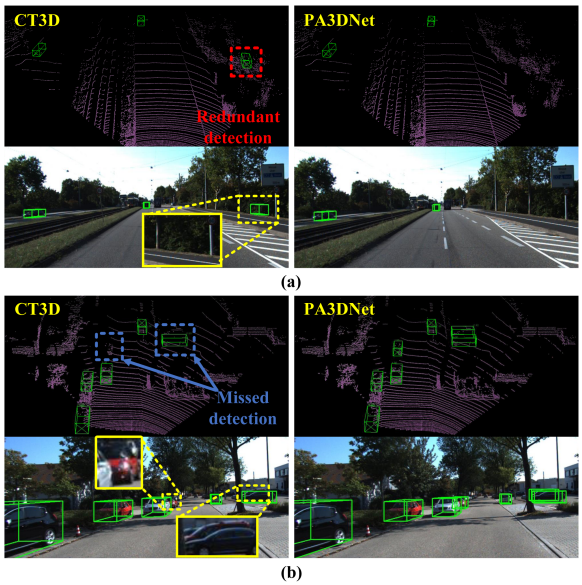

Fig. 7. Qualitative comparison results of 3-D vehicle detection. 3-D detection boxes (in green solid box) are visualized in both LiDAR point clouds and RGB images. Left column: There are missed 3-D detections (in red dotted box) and redundant 3-D detections (in blue dotted box) for the LiDAR-only approach [22]. Right column: better prediction results from PA3DNet. (a) KITTI road scene. (b) KITTI city scene.

an element-wise addition operation to combine the spatial and channel information at each nonempty voxel, and then utilize sigmoid activation function to generate a voxel-wise learnable weight vector:

$$\mathbf{E} = \sigma(\mathrm{LN}_{\mathbf{w}_3,\mathbf{b}_3}(\tanh(\mathbf{F}_{sg1} + \mathbf{F}_{sg2}))) \tag{10}$$

where the linear transformation layer $\mathrm{LN}_{\mathbf{w}_3,\mathbf{b}_3} : \mathbb{R}^{M\times D} \rightarrow \mathbb{R}^{M\times 1}$ is employed to map the multidimensional features of each sparse voxel into one channel, σ denotes the sigmoid activation function, and $E \in \mathbb{R}^M$ represents the learnable weight parameter.

The learnable weight vector $\mathbf{E}$ is derived from the dual fusion of camera features and LiDAR features, capable of representing both image semantics and geometric information. Since LiDAR points provide accurate geometric measurements for 3-D bounding box estimation, a key idea of PA3DNet is to preserve more reliable geometric information of LiDAR points in camera-LiDAR fusion, while enhancing the contextual information of 3-D points with high-level image semantics. To this end, we utilize the weight vector $\mathbf{E}$ to learn the compensation coefficients of multimodal features:

$$\mathbf{F}_{\mathrm{fuse}} = \mathcal{C}(\mathrm{LN}_{\mathbf{w}_4,\mathbf{b}_4}(\mathbf{F}_g) \oplus \mathbf{E}(\mathrm{LN}_{\mathbf{w}_5,\mathbf{b}_5}(\mathbf{F}_s))) \tag{11}$$

where $\mathcal{C}$ is a 1-D convolutional layer with output channel of D and $\mathbf{F}^{\mathrm{fuse}} \in \mathbb{R}^{M\times D}$ denotes the multimodal sparse voxel feature from the ACLF module.

D. Multimodal 3-D Vehicle Detection

1) Voxel-Based 3-D Detector: Starting from the input voxel feature $\mathbf{F}_{\mathrm{fuse}}$ with D-dimensional multimodal features, the voxel-based 3-D detector consists of a 3-D RPN [8], and a proposal refinement network [22]. First, the 3-D RPN predicts coarse 3-D vehicle bounding boxes, also known as 3-D proposals, from the input multimodal fusion feature $\mathbf{F}_{\mathrm{fuse}}$. Following the principle of utilizing precise geometric information of raw LiDAR points to refine 3-D proposals, the proposed refinement network then adopts an encoder–decoder architecture to learn the geometric features of each proposal. Specifically, the encoder performs feature encoding and feature extraction on the raw LiDAR points sampled in each 3-D proposal. The encoded point feature of each proposal is then decoded into a global representation vector, which is employed to predict the final classification confidence and the 3-D bounding box residuals, respectively.

2) Loss Functions: In the first stage of the voxel-based 3-D detector, we follow [8] to configure the loss function of 3-D RPN, which contains proposal classification loss, 3-D proposal box regression loss, and orientation angle loss:

$$\mathcal{L}_{\mathrm{RPN}} = w_1\mathcal{L}_{\mathrm{cls}} + w_2\mathcal{L}_{\mathrm{dir}} + w_3\mathcal{L}_{\mathrm{box}} \tag{12}$$

where $\mathcal{L}_{\mathrm{RPN}}$ denotes the loss of RPN network with parameters $w_1 = 1, w_2 = 2$, and $w_3 = 0.2$. The classification confidence refinement loss $\mathcal{L}_{\Delta\mathrm{cls}}$ and 3-D box refinement loss $\mathcal{L}_{\Delta\mathrm{box}}$ in the proposal refinement network are derived from [22]:

$$\mathcal{L} = w_4\mathcal{L}_{\mathrm{RPN}} + w_5\mathcal{L}_{\Delta\mathrm{cls}} + w_6\mathcal{L}_{\Delta\mathrm{box}} \tag{13}$$

where $\mathcal{L}$ denotes the total loss in the training phase of the 3-D voxel-based detector, with parameters $w_4 = w_5 = w_6 = 1$.

IV. Experimental Result and Analysis

A. Evaluation Methodology

In this section, the KITTI 3-D object detection dataset [33], which is widely available for autonomous driving, is adopted to evaluate PA3DNet. The KITTI 3-D dataset consists of 7518 test scenes, 3712 train scenes, and 3769 validation scenes, where each scene sample provides a 64-beam LiDAR point cloud, an RGB camera image, and the corresponding camera-LiDAR calibration data. The train scenes contain 14 357 3-D bounding box labels for vehicle category, where each 3-D vehicle label is classified into three detection difficulty levels: easy (E), moderate (M), and hard (H). We utilize the 3-D vehicle labels of the easy level in the train scenes to assemble the 3-D vehicle prototype. The predicted results of 3-D vehicles are evaluated by the AP metric with intersection over union (IoU) threshold of 0.7, and the 3-D detection performance for each detection difficulty level is evaluated separately.

Besides the KITTI dataset, we extend ablation experiments on the large-scale nuScenes dataset [28] to further validate the universality and robustness of the proposed ACLF method. The nuScenes dataset is a large-scale autonomous driving dataset for 3-D object detection, which contains a train set of 28 130 samples and a validation set of 6019 samples. Each sample includes a point cloud acquired by a top 32-beam LiDAR sensor, and RGB images collected by six cameras covering a 360° FOV. The mean average precision (mAP) and nuScenes detection score (NDS) metrics are adopted to evaluate 3-D detection.

TABLE I
PERFORMANCE COMPARISON OF 3-D VEHICLE DETECTION ON THE KITTI 3-D TEST SET. THE **RED**, **GREEN** AND **BLUE** COLORS SHOW THE TOP THREE BEST PERFORMANCE RESULTS

Method	Modality	3-D AP_{40} (%)			
		E	M	H	mAP
SECOND [8]	LiDAR	83.13	73.66	66.20	74.33
PointRCNN [20]	LiDAR	85.94	75.76	68.32	76.67
3DSSD [21]	LiDAR	88.36	79.57	74.55	80.83
SegVoxelNet [23]	LiDAR	86.04	76.13	70.76	77.64
Point-GNN [29]	LiDAR	88.33	79.47	72.29	80.03
PV-RCNN [24]	LiDAR	**90.25**	81.43	76.82	**82.83**
CT3D [22]	LiDAR	87.83	**81.77**	77.16	82.25
PartA^2 [4]	LiDAR	87.81	78.49	73.51	79.94
Pyramid R-CNN [32]	LiDAR	88.39	**82.08**	**77.49**	82.65
H^{2}3D R-CNN [3]	LiDAR	**90.43**	81.55	**77.22**	**83.06**
Pointformer [15]	LiDAR	87.13	77.06	69.25	77.81
PVB-SSD [31]	LiDAR	89.99	80.68	76.23	82.30
DVFENet [30]	LiDAR	86.20	79.18	74.58	79.99
MV3D [17]	LiDAR+Camera	74.97	63.63	54.00	64.20
F-PointNet [16]	LiDAR+Camera	82.19	69.79	60.59	70.86
AVOD-FPN [18]	LiDAR+Camera	83.07	71.76	65.73	73.52
MXV-Net [19]	LiDAR+Camera	83.20	72.70	65.20	73.70
PI-RCNN [13]	LiDAR+Camera	84.37	74.82	70.03	76.41
EPNet [9]	LiDAR+Camera	89.81	79.28	74.59	81.23
3D-CVF [10]	LiDAR+Camera	89.20	80.05	73.11	80.79
Sem-Aug [14]	LiDAR+Camera	89.41	80.77	75.90	82.03
PA3DNet (Ours)	LiDAR+Camera	**90.49**	**82.57**	**77.88**	**83.64**

B. Training and Inference

The training of the PSS model follows the configuration of [27]. We then employ the Adam optimizer with 100 epochs, six batch sizes, and a learning rate of 0.001 to train the voxel-based multimodal 3-D detector. Data augmentation operations for camera images include random horizontal flipping, random scaling (scale factor: 0.5–2.0x), and Gaussian blur. All training programs are conducted on two NVIDIA GTX 1080Ti GPUs. For the performance evaluation on the KITTI validation set, only the train set is utilized for training, while the performance evaluation on the KITTI test set is trained using all sample data from validation and train sets. To suppress the overfitting problem caused by the limited number of samples, data augmentation operations of random global rotation, scaling, and flipping are performed on the raw LiDAR point clouds and all labeled 3-D vehicle boxes during training. The noise of the global scaling obeys a uniform distribution of [0.95,1.05] and the noise of the global rotation is set to $[-\frac{\pi}{4}, \frac{\pi}{4}]$. Besides, a ground truth sampling strategy [8] is employed to augment the number of 3-D vehicle labels and accelerate the convergence of the 3-D vehicle detection network.

C. 3-D Vehicle Detection on the KITTI Dataset

1) Results on the Test Set: To evaluate the 3-D vehicle detection performance of PA3DNet, we follow the evaluation criteria of the official KITTI 3-D object detection dataset and calculate interpolated AP metric over 40 recall sample locations $\left[\frac{1}{40}, \frac{2}{40}, \frac{3}{40}, \ldots, \frac{39}{40}, 1\right]$. Table I reports the AP performance comparison results of PA3DNet and state-of-the-art 3-D detection methods on the KITTI 3-D object detection test set. It can be seen that existing LiDAR-only methods tend to perform better than camera-LiDAR fusion methods, while our PA3DNet instead achieves more competitive 3-D vehicle detection performance. Specifically, the 3-D vehicle detection results of each detection difficulty level are evaluated separately. PA3DNet improves the AP by at least 0.49% (versus Pyramid R-CNN [32]) in the moderate detection difficulty level, and at least 0.39% (versus Pyramid R-CNN [32]) in the hard detection difficulty level. The mAP of 3-D vehicle detection on three difficulty levels is also evaluated. Compared with the latest advanced LiDAR-only methods H^{2}3D R-CNN [3] and PV-RCNN [24], our PA3DNet reports a 058% mAP improvement (versus H^{2}3D R-CNN) and a 0.81% mAP improvement (versus PV-RCNN). The above performance evaluation results verify the effectiveness of the two proposed methods. 1) The PSS model can extract more contextual semantic information from camera images. 2) The adaptive fusion of image semantic features and point cloud features can improve the performance of multimodal methods since it bridges the gap between sparse voxel representation and abstract semantic information.

Furthermore, precision-recall curves are calculated to evaluate the precision performance at each recall locations, as shown in Fig. 5. The proposed PA3DNet outperforms both existing LiDAR-only and camera-LiDAR fusion methods, demonstrating the effectiveness of our camera-LiDAR fusion network. As a result, by compensating geometric information of LiDAR points with high-quality visual shape features, the proposed multimodal approach enriches the semantic understanding of 3-D vehicle objects, further improving the robustness and accuracy of 3-D vehicle detection.

2) Results on the Validation Set: The 3-D vehicle detection performance of PA3DNet compared to advanced methods on the KITTI validation set is further reported, as shown in Table II. For a fair comparison with previously published experimental results, we employ the same 11 recall sample points $[0, 0.1, 0.2, \ldots, 1]$ as MV3D [17] to evaluate the AP. PA3DNet achieves impressive 3-D vehicle detection performance on three difficult levels. Compared with the LiDAR-only method Pyramid R-CNN [32] and multimodal method Three-Attn [11], PA3DNet improves mAP for 3-D vehicle detection by 0.86% and 6.46%, respectively.

Considering that the top-view representation provides explicit localization information and is more applicable to intelligent decision making in autonomous driving, the performance evaluation of 3-D vehicle localization (i.e., BEV detection) is also reported in Table II. Specifically, the predicted 3-D vehicle bounding boxes are projected onto the top-view plane, and the resulting rotated 2-D bounding boxes are then evaluated with an IoU threshold of 0.7 for BEV detection. It should be noted that LiDAR-only methods generally show better detection performance than multimodal methods. This reveals that the complementary association between sparse LiDAR point cloud and image semantic features has not been well exploited in previous camera-LiDAR fusion methods. In contrast, compared to top two LiDAR-only methods, PartA^2 [4] and Point-GNN [29], PA3DNet obtains 0.38% and 0.73% mAP improvements in BEV detection, which indicates the potential of our proposed camera-LiDAR fusion method for autonomous driving perception.

TABLE II

PERFORMANCE COMPARISON OF 3-D VEHICLE DETECTION AND BEV DETECTION ON THE KITTI 3-D VALIDATION SET. THE **RED**, **GREEN** AND **BLUE** COLORS SHOW THE TOP THREE BEST PERFORMANCE RESULTS. THE SYMBOL "-" INDICATES THAT NO PUBLISHED RESULTS ARE PROVIDED BY THE METHODS

Method	Modality	3-D AP_{11} (%)				BEV AP_{11} (%)			
		Easy	Moderate	Hard	mAP	Easy	Moderate	Hard	mAP
SECOND [8]	LiDAR	87.43	76.48	69.10	77.67	89.96	87.07	79.66	85.56
PointRCNN [20]	LiDAR	89.19	78.85	77.91	81.98	**90.21**	87.89	85.51	87.87
3DSSD [21]	LiDAR	**89.71**	79.45	**78.67**	**82.61**	-	-	-	-
SegVoxelNet [23]	LiDAR	89.35	79.05	77.41	81.94	-	-	-	-
Point-GNN [29]	LiDAR	87.89	78.34	77.38	81.20	89.82	**88.31**	**87.16**	**88.43**
PV-RCNN [24]	LiDAR	-	**83.90**	-	-	-	-	-	-
$PartA^2$ [4]	LiDAR	**89.47**	79.47	78.54	82.49	**90.42**	**88.61**	**87.31**	**88.78**
Pyramid R-CNN [32]	LiDAR	89.37	**84.38**	**78.84**	**84.20**	-	-	-	-
MV3D [17]	LiDAR+Camera	71.29	62.68	56.56	63.51	86.55	78.10	76.67	80.44
F-PointNet [16]	LiDAR+Camera	83.76	70.92	63.65	72.78	88.16	84.02	76.44	82.87
AVOD-FPN [18]	LiDAR+Camera	84.41	74.44	68.65	75.83	-	-	-	-
MVX-Net [19]	LiDAR+Camera	85.50	73.30	67.40	75.40	89.50	84.90	79.00	83.47
PI-RCNN [13]	LiDAR+Camera	88.27	78.53	77.75	-	-	-	-	-
Three-Attn [11]	LiDAR+Camera	85.12	76.23	74.46	78.60	89.64	86.23	85.60	87.16
PA3DNet (Ours)	LiDAR+Camera	**89.87**	**86.31**	**79.43**	**85.20**	**90.56**	**88.64**	**88.29**	**89.16**

TABLE III

PERFORMANCE COMPARISON OF 3-D MULTICLASS OBJECT DETECTION ON THE KITTI VALIDATION SET, WHERE * INDICATES REPRODUCED RESULTS USING THE OPEN SOURCE CODE. THE BEV DETECTION AND 3-D OBJECT DETECTION ARE EVALUATED BY MEAN AP WITH 40 RECALL POSITIONS

Methods	Car (3-D Detection)			Car (BEV Detection)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
baseline*	92.20	84.93	82.94	95.68	90.17	87.79
PA3DNet-3CLS	93.18	86.02	83.74	96.59	91.11	89.16
Improvement	**+0.98**	**+1.09**	**+0.8**	**+0.91**	**+0.94**	**+1.37**
Methods	Pedestrian (3-D detection)			Pedestrian (BEV detection)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
baseline*	59.65	54.30	50.12	62.63	57.55	53.82
PA3DNet-3CLS	70.99	63.79	58.35	73.89	67.41	62.30
Improvement	**+11.34**	**+9.49**	**+8.23**	**+11.26**	**+9.86**	**+8.48**
Methods	Cyclist (3-D detection)			Cyclist (BEV detection)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
baseline*	89.02	71.80	67.53	89.71	75.18	70.94
PA3DNet-3CLS	89.98	73.25	69.00	90.80	75.62	71.36
Improvement	**+0.96**	**+1.45**	**+1.47**	**+1.09**	**+0.44**	**+0.42**

TABLE IV

PERFORMANCE COMPARISON OF 3-D VEHICLE DETECTION AT DIFFERENT DISTANCES ON THE KITTI VALIDATION SET

Method	Modality	Difficulty	3-D AP_{11} (%)		
			0-30m	30-50m	>50m
CT3D	LiDAR	Easy	89.42	53.88	-
CT3D + ACLF	LiDAR+RGB		**89.78**	**59.25**	**-**
Improvement	-	-	*0.36*	*5.37*	-
CT3D	LiDAR	Mod.	89.79	62.63	10.61
CT3D + ACLF	LiDAR+RGB		**90.04**	**63.49**	**20.28**
Improvement	-	-	*0.25*	*0.86*	*9.67*
CT3D	LiDAR	Hard	89.23	62.58	13.04
CT3D + ACLF	LiDAR+RGB		**89.44**	**63.10**	**15.70**
Improvement	-	-	*0.21*	*0.52*	*2.66*

The bold values are the highest 3-D detection accuracy in comparative experiments under the same distance conditions.

D. Robustness Against Nonrigid Road User Classes

In order to promote the application of our method on 3-D object detection of more road users, we extend a multiclass 3-D object detection experiment for PA3DNet, evaluating on the KITTI validation set with *Car*, *Pedestrian*, and *Cyclist* annotations, as shown in Fig 6. For *Pedestrian* and *Cyclist* classes, the rotated IoU threshold of mAP calculation is set to 0.5 according to the general configuration [4]. The KITTI train set and validation set provide 2207/2280 *Pedestrian* annotations and 734/893 *Cyclist* annotations, respectively. For the multiclass 3-D object detection experiment, we leverage the open source code[3] provided by CT3D [22] to reproduce the LiDAR-only baseline results. The multiclass 3-D detection version of PA3DNet (PA3DNet-3CLS) is trained for 100 epochs with a batch size of 24 and a learning rate of 0.001 on the KITTI train set. As shown in Table III, the results indicate that the proposed approach can improve the performance of BEV detection and 3-D detection at each difficulty level, especially for *Pedestrian* class. We argue that the information loss of LiDAR point clouds becomes worse for *Pedestrians* and *Cyclists* due to their smaller shape sizes, while the fusion of semantic image features in the proposed approach brings more useful contextual information. As a result, the performance gains of 3-D pedestrian detection and 3-D cyclist detection are more significant.

[3][Online]. Available: https://github.com/hlsheng1/CT3D

E. Effects When Point Cloud Distance Changes

The distance between the 3-D object and the LiDAR sensor is positively correlated with the sparsity of raw LiDAR point clouds. We investigate the benefits of the proposed ACLF module at different distance ranges, as shown in Table IV , where the LiDAR-only baseline method comes from CT3D [22] and only processes the input LiDAR point cloud. The 3-D vehicle objects in the range of $[0, 30\,\mathrm{m})$ have relatively dense point clouds. By integrating ACLF module, the AP of the CT3D+ACLF method for near-range 3-D vehicles is improved (Easy: 0.36%, Moderate: 0.25%, Hard: 0.21%). Furthermore, distant 3-D vehicle objects in the range of > 50 m have rather sparse point clouds with highly incomplete geometric information. Due to the adaptive compensation of abstract visual shape features, the CT3D+ACLF method reports significant AP improvements (Moderate: 9.67%, Hard: 2.66%), which demonstrates that the proposed ACLF module can provide more fine-grained contextual information for sparse point clouds. The above results also validate the robustness of our proposed approach when the LiDAR density varies.

TABLE V
PERFORMANCE COMPARISON OF 3-D OBJECT DETECTION ON NUSCENES VALIDATION SET. (*: OUR RE-IMPLEMENTATION)

Method	Modality	NDS	mAP
TransFusion-L*	LiDAR	68.63	64.60
BEVFusion*	LiDAR+Camera	71.15	67.71
BEV-ACLF	LiDAR+Camera	**71.34**	**68.29**

The bold values are the highest 3-D detection results.

TABLE VI
RUNTIME PER FRAME ON THE KITTI VALIDATION DATASET. THE TOTAL RUNTIMES OF SINGLE THREAD AND MULTITHREADING ARE DENOTED AS *S-* AND *M-*, RESPECTIVELY. ALL RUNTIMES ARE TESTED ON AN NVIDIA 1080TI GPU

Modules	Segmentation	ACLF	3-D Detector	Total time	
				s-	*m-*
Time (s)	0.13	0.01	0.07	0.21	0.14

F. Generalization to LiDAR Resolution Variations

Different from the KITTI dataset with 3-D point clouds from a 64-beam LiDAR scan, the 3-D point cloud samples on the nuScenes dataset are acquired from a 32-beam LiDAR sensor. To analysis the generalization and robustness of the proposed ACLF with LiDAR resolution variations, we conduct an ablation study on the nuScenes validation dataset. Inspired by the BEV representation of BEVFusion [35], we employ the same BEV architecture to create a fusion variant, named BEV-ACLF, while switching the fully convolutional fusion layer in BEVFusion with the proposed ACLF module. The LiDAR branch of BEV-ACLF is the same as TransFusion-L [34]. Table V reports the comparison of BEV-ACLF with TransFusion-L and BEVFusion on the nuScenes validation dataset. We train BEV-ACLF on the nuScenes detection dataset following the same training configuration settings as in [35]. Compared with TransFusion-L and BEVFusion, BEV-ACLF achieves 3.69% and 0.58% mAP improvement, respectively. This illustrates that the proposed ACLF module can also benefit the cross-fusion of LiDAR point clouds and camera features in BEV representation space. The results also validate the generalization of the proposed method in case of LiDAR resolution variation.

G. Qualitative Results

Fig. 7 shows the visualization results of 3-D vehicle detection from PA3DNet and the LiDAR-only baseline method CT3D [22] on the KITTI 3-D dataset, respectively. The predicted 3-D vehicle results are visualized in both RGB images and LiDAR point clouds. It can be seen that PA3DNet overcomes the problem of false detection caused by sparse and irregular LiDAR points, and also improves the missed detection cases caused by occlusion and long distance, validating that the proposed multimodal fusion method is beneficial for sparse LiDAR point cloud, as shown in Fig. 6.

H. Runtime Analysis

The runtime of PA3DNet per frame is reported in Table VI. The PSS model and the voxel-based 3-D detector can run in parallel threads to reduce the overall runtime. We evaluate the total runtime of PA3DNet in both single thread (i.e., each module runs sequentially) and multithreading settings. Note that the proposed ACLF module takes only 10 ms to fuse LiDAR point features and semantic image features. Therefore, the latency of the ACLF module combined with other detectors is almost negligible, which further demonstrates the efficiency of the proposed approach. It is worth mentioning that PA3DNet is modular in design, which means that the 2-D shape segmentation model can be easily replaced by other lightweight semantic segmentation networks, further reducing the total running time.

V. CONCLUSION

In this article, we propose a camera-LiDAR fusion-based 3-D vehicle detection approach for accurate 3-D perception in intelligent transportation systems. To address the high-cost problem of image segmentation annotation in multimodal methods, a PSS model is designed to learn vehicle shape features without prior manual image annotation. By adopting the bilinear interpolation sampling algorithm, image semantic features are aligned with sparse 3-D LiDAR points in a point-wise manner. The proposed ACLF module formulates cross-modal attention between the two modalities. As a result, the semantic information of vehicles in the images can be reasonably encoded without additional manual image annotations, further improving the 3-D vehicle detection performance in challenging cases. Qualitative and quantitative experimental results on autonomous driving scenarios demonstrate that the proposed method is able to perform efficient multimodal 3-D vehicle detection in challenging long-range environments. In summary, PA3DNet provides a new perspective of 3-D perception for autonomous vehicles, which is also complementary to camera-LiDAR fusion. In future work, the resolution gap between sparse LiDAR and dense image semantic features will be studied to preserve more image contextual information. The multimodal fusion framework will be integrated into the safe navigation module of intelligent vehicles. It will be helpful for intelligent vehicles in maintaining a safe driving distance on urban roads and avoiding potential collisions caused by 3-D obstacles.

REFERENCES

[1] M. Haris and A. Glowacz, "Lane line detection based on object feature distillation," *Electronics*, vol. 10, no. 9, 2021, Art. no. 1102.
[2] M. Abdel-Basset, V. Chang, and R. Mohamed, "A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems," *Neural Comput. Appl.*, vol. 33, no. 17, pp. 10685–10718, 2021.
[3] J. Deng et al., "From multi-view to hollow-3D: Hallucinated hollow-3D R-CNN for 3D object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4722–4734, Dec. 2021.
[4] S. Shi et al., "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, 1 Aug. 2021.
[5] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A general framework for monocular 3D object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5170–5184, Sep. 2022, doi: 10.1109/TPAMI.2021.3074363.
[6] W. Bao, B. Xu, and Z. Chen, "MonoFENet: Monocular 3D object detection with feature enhancement networks," *IEEE Trans. Image Process.*, vol. 29, pp. 2753–2765, Nov. 2019, doi: 10.1109/TIP.2019.2952201.
[7] M. Haris and A. Glowacz, "Navigating an automated driving vehicle via the early fusion of multi-modality," *Sensors*, vol. 22, no. 4, 2022, Art. no. 1425.

[8] Y. Yan et al., "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018, Art. no. 3337.
[9] T. Huang et al., "EPNet: Enhancing point features with image semantics for 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 35–52.
[10] J. H. Yoo et al., "3D-CVF: Generating joint camera and LiDAR features using cross-view spatial feature fusion for 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 720–736.
[11] L.-H. Wen and K.-H. Jo, "Three-attention mechanisms for one-stage 3-D object detection based on LiDAR and camera," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 6655–6663, Oct. 2021.
[12] S. Vora et al., "PointPainting: Sequential fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4604–4612.
[13] L. Xie et al., "PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 12460–12467, 2020.
[14] L. Zhao, M. Wang, and Y. Yue, "Sem-Aug: Improving camera-LiDAR feature fusion with semantic augmentation for 3D vehicle detection," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9358–9365, Oct. 2022.
[15] X. Pan et al., "3D object detection with pointformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7463–7472.
[16] C. R. Qi et al., "Frustum pointnets for 3d object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.
[17] X. Chen et al., "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.
[18] J. Ku et al., "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.
[19] V. A. Sindagi, Y. Zhou, and O. Tuzel, "MVX-Net: Multimodal VoxelNet for 3d object detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 7276–7282.
[20] S. Shi et al., "Pointrcnn: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770—779.
[21] Z. Yang et al., "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11040–11048.
[22] H. Sheng et al., "Improving 3D object detection with channel-wise transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2743–2752.
[23] H. Yi et al., "SegVoxelNet: Exploring semantic context and depth-aware features for 3D vehicle detection from point cloud," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 2274–2280.
[24] S. Shi et al., "PV-RCNN: Point-Voxel feature set abstraction for 3d object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10529–10538.
[25] X. Wu et al., "Sparse fuse dense: Towards high quality 3D detection with depth completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5418–5427.
[26] M. Liang et al., "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 641–656.
[27] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," 2020, *arXiv:2005.10821*.
[28] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.
[29] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1711–1719.
[30] Y. He et al., "DVFENet: Dual-branch Voxel feature extraction network for 3D object detection," *Neurocomputing*, vol. 459, pp. 201–211, Oct. 2021.
[31] K. Ning, Y. Liu, Y. Su, and K. Jiang, "Point-Voxel and bird-eye-view representation aggregation network for single stage 3D object detection," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–13, Dec. 2022, doi: 10.1109/TITS.2022.3225880.
[32] J. Mao et al., "Pyramid R-CNN: Towards better performance and adaptability for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2723–2732.
[33] A. Geiger et al., "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[34] X. Bai et al., "Transfusion: Robust lidar-camera fusion for 3D object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022,, pp. 1090–1099.
[35] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," 2022, *arXiv:2205.13542*.

Meiling Wang received the B.S. and M.S. degrees in automation and the Ph.D. degree in navigation, guidance, and control from the Beijing Institute of Technology, Beijing, China, in 1992, 1995, and 2007, respectively.

She was with the University of California San Diego as a Visiting Scholar in 2004. Since 1995, she has been with the Beijing Institute of Technology, where she is currently a Professor and the Director of Integrated Navigation and Intelligent Navigation Laboratory. Her research interests include advanced technology of sensing and detecting and vehicle intelligent navigation.

Lin Zhao received the B.E. degree in automation in 2018 from the School of Automation, Beijing Institute of Technology, Beijing, China, where he is currently working toward the Ph.D. degree in control science and engineering with the School of Automation.

His research interests include computer vision and autonomous driving.

Yufeng Yue (Member, IEEE) received the B.Eng. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2014, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2019.

He is currently a Professor with the School of Automation, Beijing Institute of Technology. He has authored a book in Springer, and authored or coauthored more than 40 journal/conference papers, including IEEE TRANSACTIONS ON MULTIMEDIA, IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and conferences like ICRA/IROS. His research interests include perception, mapping, and navigation for autonomous robotics.

He is an Associate Editor for 2020–2023 IEEE IROS. He was the recipient of the 2020 IEEE ICARCV Best Paper Award and the 2021 IEEE ICUS Best Paper Award.