


Article

Sparse Sensor Fusion for 3D Object Detection with Symmetry-Aware Colored Point Clouds

Lele Wang ¹, Peng Zhang ^{1,*}, Ming Li ² and Faming Zhang ¹

¹ School of Artificial Intelligence, Suzhou Chien-Shiung Institute of Technology, Suzhou 215400, China; 0072@csit.edu.cn (L.W.); 240302712137@csit.edu.cn (F.Z.)

² School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China; minglee2015@126.com

* Correspondence: 9076@csit.edu.cn

Abstract: Multimodal fusion-based object detection is the foundational sensing task in scene understanding. It capitalizes on LiDAR and camera data to boost the robust results. However, there are still great challenges in establishing an effective fusion mechanism and performing accurate and diverse feature interaction fusion. In particular, the relationship construction between the two modalities has not been comprehensively exploited, leading to sensor data utilization deficiencies and redundancies. In this paper, a novel 3D object-detection framework, namely a symmetry-aware sparse sensor fusion detection network (2SFNet), is proposed. This framework was designed to leverage point clouds and RGB images. The 2SFNet consists of three submodules, filtered colored point cloud generation, pseudo-image generation, and a dilated feature fusion network, to solve these problems. Firstly, filtered colored point cloud generation constructs non-ground colored point cloud (NCPC) data by employing an early fusion strategy and a ground-height-filtering module, selectively retaining only object-related information. Subsequently, 2D grid encoding is used on the reduced colored data. Finally, the processed colored data are fed into the improved PillarsNet architecture, which now has expanded receptive fields to enhance the fusion effect. This design optimizes the fusion process by ensuring a more balanced and effective data representation, aligning with the symmetry concept that underlies the model's functionality. Experiments and evaluations were conducted on the KITTI dataset to present the effectuality, particularly for categories characterized by sparse point clouds. The results indicate that the symmetry-aware design of the 2SFNet leads to an improved performance when compared to other multimodal fusion networks, and alleviates the phenomenon caused by highly obscured and crowded scenes.

Keywords: data fusion; point cloud; object detection; deep learning; symmetry



Citation: Wang, L.; Zhang, P.; Li, M.; Zhang, F. Sparse Sensor Fusion for 3D Object Detection with Symmetry-Aware Colored Point Clouds.

Symmetry **2024**, *16*, 1690. <https://doi.org/10.3390/sym16121690>

Academic Editors: Quanxin Zhu and Marek T. Malinowski

Received: 18 October 2024

Revised: 17 November 2024

Accepted: 5 December 2024

Published: 20 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the domains of AI-driven applications, robotics, autonomous driving, etc., object detection is the most critical and foundational prerequisite. Its purpose is to localize and classify objects with bounding boxes in 3D space for subsequent path planning.

Autonomously driving cars are typically equipped with a variety of sensors, including RGB cameras and 3D LiDAR sensors, which can supply information for perceiving the surrounding environment. RGB images acquired by cameras can offer plentiful color and dense texture details with pixels in the RGB color space, allowing for enhanced visual feature extraction. However, they lack in-depth information, which limits their performance. In contrast, point clouds scanned by LiDAR can seize 3D structures and precise depth data, presenting a clear outline of object shapes and spatial arrangements. Nevertheless, owing to their inherent characteristics, these point clouds usually become sparser and more precarious from the center of the digital scanner outwards. Moreover, the relationship between the two modalities is not sufficient to fetch geometric features. The sparsity of

data related to small, distant, and obscured objects may induce the performance collapse of LiDAR-based methods. By combining their advantages and fusing complementary information from these two modalities, the completeness of the description of the surrounding environment can be increased, and the performance of 3D object-detection tasks can be further boosted [1,2].

Owing to the inherent nature of point clouds in irregularity and sparsity, LiDAR-only-based approaches convert 3D data into projected 2D views [3–12] and voxel grids [13–19] using convolutional neural networks, or directly into pure point clouds by utilizing PointNet [20–22]. The common drawback of 2D view projection and voxelization is that they inevitably result in the loss of crucial 3D information. While PointNet is able to extract features from 3D data with fewer points, its magnitude for larger data remains uncertain. Designing a rational spatial convolution network for data processing is a persistent challenge, unlike the techniques used for 2D image detection. In addition, small objects like vehicles and pedestrians have no distinct geometric shapes and fewer points, making them unable to be recognized by LiDAR-only methods. Furthermore, the lack of semantic color information can result in false detection in the case of similar objects. Therefore, researchers have greatly focused on multimodal fusion methods that employ multimodal sensors to supplement the rich semantic surface, supplied by a camera with the precise position capabilities of LiDAR.

Depending on the fusion processing, different fusion strategies can be divided into three categories: early-level fusion [23–26], middle-level fusion [27–44], and late-level fusion [45,46]. As shown in Figure 1, the fusion strategies vary depending on the stage in which the RGB image participates.

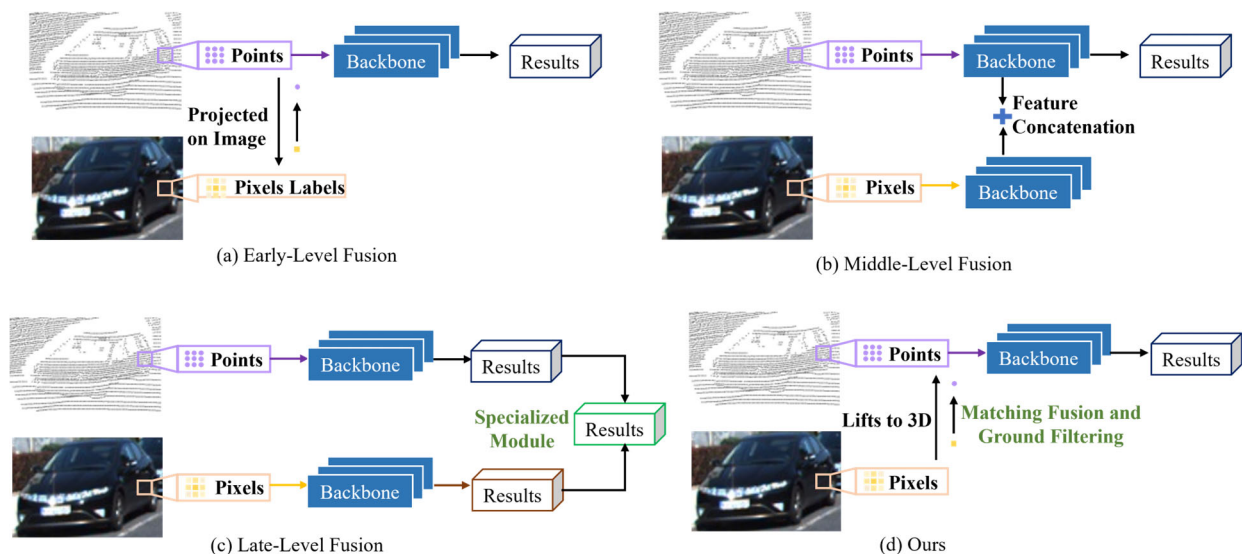


Figure 1. Illustration of early-level fusion, middle-level fusion, late-level fusion, and our fusion architectures.

The early-level fusion [23–26] strategy involves incorporating raw multimodal data into a unified coordinate system directly; then, the perception data are imported into the designed module, e.g., augmenting LiDAR data with appropriate semantic labels [23,24]. This strategy produces redundant information and demands a heavy level of computation during data storage and processing. Most scholars have primarily focused on the middle-level fusion [27–44] strategy, which entails extracting features from different 2D projected maps and RGB images, and then learning fusion representation using a deep intermediate network, offering a balance between sensitivity and flexibility. These methods often perform addition or concatenation on different projected views without considering the varying validity of both modality features. Although this strategy is very popular in ensuring information consistency, it lacks sensitivity to the point-wise correspondence

between sensor features, which is relatively coarse. The late-level fusion [45,46] strategy processes each sensor data point on its independent detection module and joins the 2D and 3D results based on interrelationships or special models. It ignores the potential information generated by the performance of each sensor, involving less interaction between different modalities. Both the middle- and late-level strategies must apply two independent networks to extract the features of each modality, making it hard to achieve the optimal point-wise correspondence. Their performance is limited by 2D mature detectors. As the early-level strategy records two modes into a consistent coordinate system, which can realize the fusion of limited perception data, it simplifies the design and improves the efficiency of convolutional coding. Nevertheless, early-level fusion methods have not been thoroughly explored yet, and we aimed to fill this gap and expand the current knowledge in this field.

Most multimodal fusion-based works [27–32,39–44] have generated projected maps or established connections between 3D points and 2D pixels by means of projecting the 3D points onto the image plane to locate the appropriate pixels. These methods are limited by spatial information in the encoding process; merely a small number of points will complete the matching fusion. In contrast to the aforementioned methods, this method lifts image pixels into a 3D space and chooses to encode the 3D sparse data with a colored texture.

In this paper, we proposed a symmetry-aware sparse sensor fusion network (2SFNet) detection method that addresses the problems mentioned above. This 2SFNet consists of three submodules: filtered colored point cloud generation, pseudo-image generation, and a dilated feature fusion network. In the filtered colored point-cloud-generation module, the early fusion strategy and the height-filtering module are utilized to construct a 7D colored sparse point cloud. In the former, the point clouds and RGB images are jointly calibrated and fused based on their transformation relationships. This constrains the detection range through joint calibration, whereby the RGB pixels are projected into 3D space to augment the point cloud (XYZr). The latter module was adopted to eliminate invalid ground data, accelerate the subsequent network encoding speed, further refine the point cloud, and align with the symmetry of the scene by selectively retaining only object-related information. Then, in the pseudo-image-generation module, 2D grid encoding is applied to reduce the generated colored data. Finally, in the dilated feature fusion network module, the processed colored data are fed into the improved PillarsNet architecture with expanded receptive fields to enhance the fusion effect. The main contributions can be categorized as follows:

- (1) In this paper, a novel 3D object-detection framework, namely a symmetry-aware sparse sensor fusion detection network (2SFNet), is proposed. It was designed to take advantage of point clouds and RGB images.
- (2) To the best of our knowledge, this is the first method that lifts pixels into 3D space, enhances the representation of 3D point data by incorporating image pixels point-to-point, and employs a height-filtered module to filter ground points, thereby constructing a new 7D colored point dataset.
- (3) This paper proposes an improved PillarsNet with an increased receptive field network to deeply encode the processed colored data for multiscale feature fusion learning, aligning with the symmetry of feature extraction and further reinforcing the symmetry in feature representation.

The rest of this paper is organized as follows: Section 2 briefly reviews the related works on LiDAR-based and multimodal data-fusion-based object detection. An overview of the proposed 2SFNet model and its methodology (filtered colored point clouds, pseudo-image generation, the dilated feature fusion network, and the loss function) are introduced in Section 3. Section 4 presents an evaluation of the 2SFNet on a public dataset, including the implementation details and experimental results. Section 5 summarizes the current model.

2. Related Work

LiDAR sensors provide highly accurate depth information, excel at distance measurements and structure recognition, and enable detailed 3D mapping of the environment.

Meanwhile, RGB data offer rich color and texture details. This paper integrates multimodal data fusion to enhance the object-detection performance by providing complementary information, particularly in complex scenes where occlusion and ambiguity may arise.

This section investigates recent approaches to the use of deep learning models for 3D point cloud object detection, especially focusing on LiDAR-only-based and multimodal data-fusion-based detection tasks.

2.1. LiDAR-Only-Based Detection

Conventional deep learning networks struggle to directly transfer to point clouds because the structure of point clouds is irregular. According to point cloud representations, LiDAR-only-based methods commonly involve partitioning the irregular point clouds into 2D views [3–12], voxel grids [13–19], and directly using pure point clouds as the input [20–22].

Projected 2D-view-based methods: To reduce the computational burden, some algorithms refer to the projection of 3D data into 2D images, such as front view formats like VeloFCN [3], LMNet [4], and FVNet [5] and bird’s-eye-view (BEV) formats like BirdNet [6], BirdNet+ [7], Complex-YOLO [8], RT3D [9], PIXOR [10], and HDNet [12]. After the conversion, a proven 2D convolution technique can be directly implemented. The front view is resemblant of the RGB image and comprises coordinates in space. The BEV format is typically utilized for autopilots, since objects do not superimpose on the height axis, making it easier to obtain the position and appearance of objects.

Voxel-grid-based approaches discretize 3D spaces into voxel structures with a uniform size and utilize a formal 3D CNN [14] to obtain a high-dimensional representation. Notably, PointPillars [11] proposes a pseudo-image manner that divides voxels only on the plane. PVRCNN [13] leverages the key point features with the output of 3D space convolution to improve proposal generation. The voxel feature extractor [15] expands the receptive field and enhances the context of the extracted features. Based on VoxelNet [15], SECOND [16] enhances the efficiency of 3D convolution by utilizing sparse convolution modules to erase null voxels. For the special case of occluded vehicles, SegVoxelNet [17] designs a depth-aware head with different kernel sizes and convolutional layer expansion rates. The recent works on PVRCNN++ [18] and Lidar RCNN [19] employed the attention mechanism to extract and detect features within voxels.

Pure point-cloud-based methods are dedicated to analyzing pure points directly. With the advent of PointNet [20], it is possible to perform convolution operations on 3D points. PointFormer [21] applies the transformer model to a 3D object-detection network. The advantages of PointNet in translation invariance, local connections, and shared parameters have spawned some specialized versions [21,22]. However, both the computation and memory consumption of computing 3D models increase cubically.

2.2. Multimodal Data-Fusion-Based Detection

Existing sensor fusion-based approaches are broadly categorized into three groups: early-level [23–26], middle-level [27–44], and late-level fusion [45,46].

Early-level fusion-based methods directly overlay the two types of modal sensing data at the data level, e.g., PointPainting [23] utilizes DeeplabV3+ to acquire per-pixel labels and then projects these labels back to the 3D space and constructs the augmented data by superposition. Dense sequential fusion [24] utilizes the foreground mask to selectively enhance the point clouds, focusing on relevant objects and ignoring background noise. This type of method does not merge the high-level features of different modalities. PointAugmenting [26] applies the deep features extracted from 2D images to augment LiDAR data. **Middle-level fusion-based methods** [27–44] extract and combine features into a single feature vector, e.g., MV3D [27] and AVOD [28] take different 2D projected perspectives as inputs for different pipelines and generate 3D proposals to predict bounding boxes.

BEVfusion [33,34] employs LSS [33] operations to project image features into BEV space, and then integrates the two modality features by concatenation.

MENet [35] introduces a mapping pyramid that leverages the semantic representation of image features at various stages and incorporates an attention-mechanism-based fusion module to refine point cloud features with auxiliary image features. To address the imbalance between foreground instances and background samples in BEV space, IS fusion [36] comprises a hierarchical scene fusion (HSF) module and an instance-guided fusion (IGF) module; the former captures the multimodal scene context at different granularities, while the latter aggregates the local multimodal context for each instance. MMAF-Net [37] combines data-level fusion with feature-level fusion to fully exploit the strengths of multimodal information, and it designs a region attention adaptive fusion module by utilizing an attention mechanism to guide the network.

Moreover, based on dense feature representations, some methods [38,39] utilize a view transformation module to construct 3D features of multi-view perspectives, which are then fused with point-level elements. However, the switching module incurs extra computing costs due to the sparsity of redundant spatial information. As the perceived distance rises, the computational load and memory requirements of the model increase dramatically, limiting its practical application. Late-level fusion-based methods [45,46] combine 2D and 3D modality results from different detectors, and then decide the final accurate 3D prediction. The frustrum-based methods, F-PointNet [45] and F-ConvNet [46], yield 2D proposals from the image first, which are reprojected into 3D space, and then they utilize PointNet as a basic feature extractor within the 3D space, viewing the frustrum from 2D region proposals to yield highly accurate trajectories.

The above methods need to transform 3D data into a specific perspective or voxel, which may involve the loss of information. Although multimodal fusion methods have been extensively studied, there is a lack of research on mapping relationship optimization.

In conclusion, compared with existing methods that utilize either a complex model to deal with different types of modal data or specific late fusion modules, this paper designed a simple, yet effective, fusion strategy to reduce the amount of data computations in the early stage and achieve interactions between modal features.

3. 2SFNet Method for LiDAR–Camera Fusion

To achieve satisfactory LiDAR–camera fusion, the proposed 2SFNet and its entire model for colored data detection are presented in Figure 2. This model consists of two modalities: RGB images captured by the camera and sparse point clouds by Velodyne 64E LiDAR from Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) [47]. The Klein’s Institute of Technology in the Information (KITTI) Vision Benchmark Suite is a widely used dataset in the field of autonomous driving research. It encompasses a rich collection of driving scenarios captured in urban, rural, and highway settings, including synchronized stereo vision and LiDAR data. This dataset enables comprehensive evaluations of various algorithms for tasks such as 3D object detection, visual odometry, and semantic segmentation.

This model restricts the search space through joint calibration, adds color and texture information to corresponding point clouds through a fusion module in 3D space, and removes a lot of inefficient data through a ground-height-filtering module. Subsequently, based on high-quality corresponding features between neighborhood points, the improved PillarsNet with increased receptive fields was introduced.

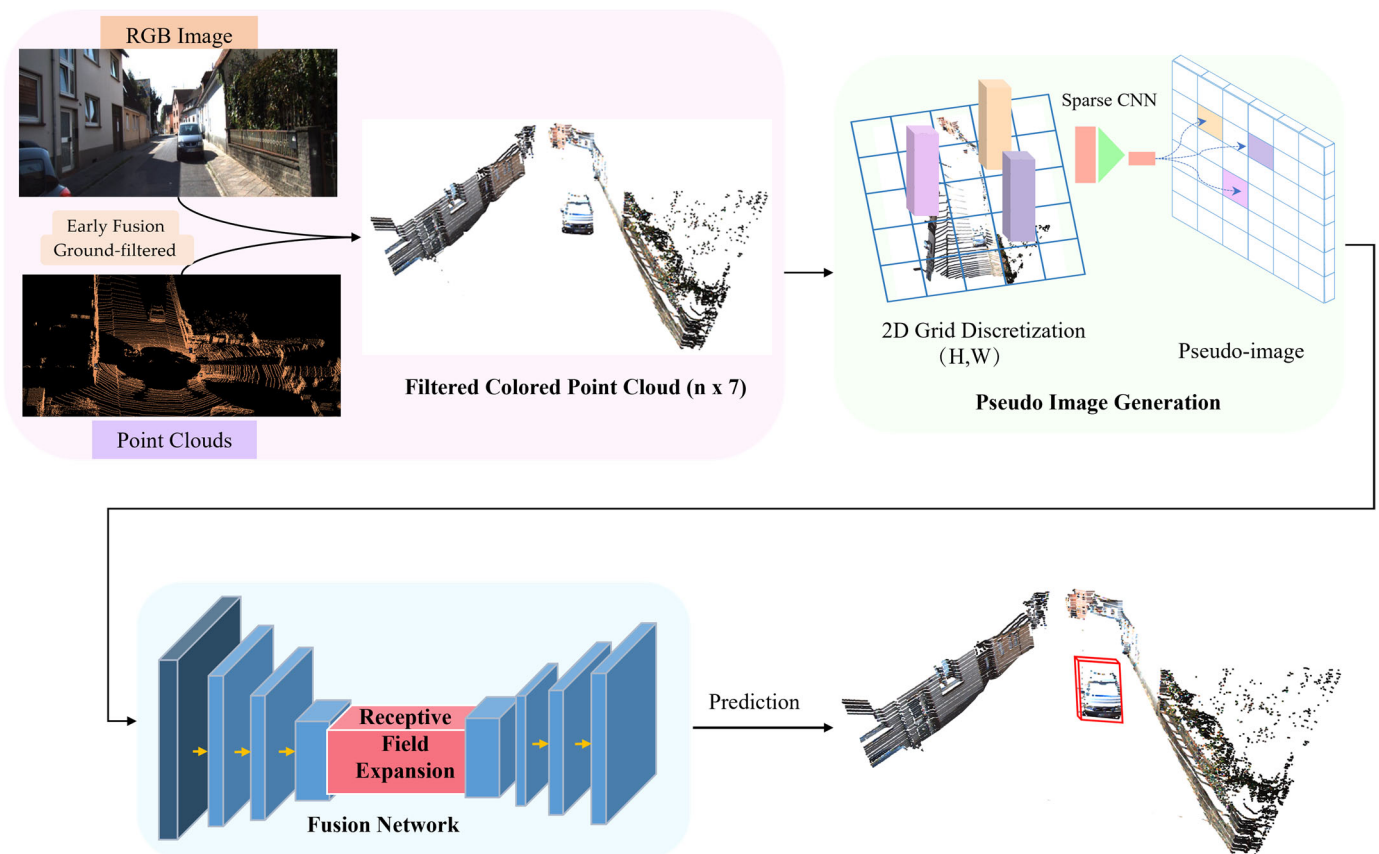


Figure 2. The proposed 2SFNet architecture.

3.1. Filtered Colored Point Cloud Generation

It is commonly evident that the amount of information for pixels is considerably larger than the points in all categories, and a single instance of LiDAR data contains more than one million points. This model recognizes the disparity in the information density between pixels and points and aims to maintain symmetry in the processing of both modes. For real-time detection, this model constrains the effective relevant search regions and reduces the amount of data processing.

The filtered colored point cloud model consists of an early fusion strategy and height-filtering module. In the former, informed by the symmetry concept, the detection range is constrained through joint calibration, whereby the image pixels (RGB) are cast into 3D space so as to augment the point cloud (XYZr). The latter module is utilized to filter out invalid ground data, and then the new colored data, referred to as the non-ground colored point cloud (NCPC), are generated. The primary advantage of a 7D colored point cloud over the traditional 3D point cloud lies in the enhanced information. The additional dimensions (color and intensity) provide significant contextual clues and detailed appearance features, which improve the accuracy and robustness of object detection, especially in complex environments. The multiplication process is exemplified in Figure 3.

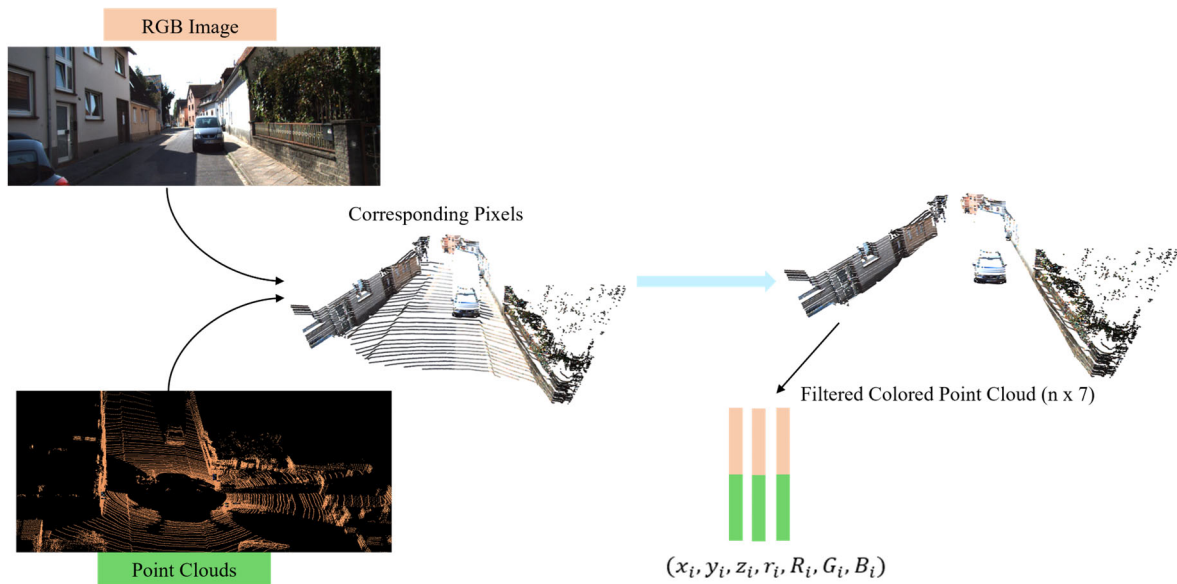


Figure 3. Schematic of filtered 7D colored point cloud generation. By means of a calibration matrix (calib.txt), the RGB pixels are cast onto corresponding points. R and T are the rotation matrix.

3.1.1. Early Fusion Strategy

An early-level fusion strategy plays a crucial role in our module by integrating LiDAR and camera data into a unified coordinate system before the model is processed. This enables a simplified designed model to leverage both modalities' strengths simultaneously, allowing for a more comprehensive understanding of the scene and improving the model's ability to perform convolutional coding.

Despite the sparsity of point clouds, an early fusion strategy can achieve fusion with limited image pixels. These finite pixels can still provide rough texture information, especially for nearby objects, forming rich texture features. Therefore, an early fusion strategy is designed to assign color texture information to the point cloud. Two types of modal data are calibrated by synchronizing and calibrating the parameters. The transformation equations are as follows:

$$P_{cam} = R_{rect}^0 \cdot T_{velo}^{cam} \cdot P_{lidar} \quad (1)$$

$$p_{cam} = T_{proj} \cdot P_{cam} \quad (2)$$

$$T_{velo}^{cam} = \begin{bmatrix} R_{velo}^{cam} & t_{velo}^{cam} \\ 0 & 1 \end{bmatrix} \quad (3)$$

where R_{rect}^0 is the rotation matrix, t_{velo}^{cam} is the transformation matrix from the LiDAR to the camera coordinate system, and T_{proj} is the projection matrix from the camera coordinate systems. To be more specific, the object-detection space is set to $\{[x, y, z]^T | x \in [0, 70]m, y \in [-40, 40]m, z \in [-3, 3]m\}$.

By leveraging the projection matrix, pixels are projected onto equivalent points in 3D space, and the remaining pixels are discarded. The color features of the image pixels are then associated with the corresponding 3D data, generating a 7D colored point cloud. Thus, each generated 7D colored data point not only comprises 3D coordinates and the reflection intensities, but also the surface color of the corresponding pixel points on the image plane, maintaining symmetry in the data representation. The 7D colored point cloud is generated by augmenting a traditional 3D point cloud with additional features, providing richer information and enhancing the object-detection capacities. Each colored point cloud feature represents $p_i = (x_i, y_i, z_i, r_i, R_i, G_i, B_i)$. Specifically, the x_i, y_i, z_i dimensions represent 3D spatial coordinates; r_i represents the reflectance value from the LiDAR data; and R_i, G_i, B_i represent the color information from the camera image.

3.1.2. Ground-Height-Filtering Module

Typically, a single frame of LiDAR data contains more than one million point clouds; therefore, it is difficult to meet real-time requirements using point-wise coding. The ground filter was adopted in our model to eliminate invalid point clouds and accelerate the subsequent network encoding speed.

The ground-height-filtering module in the 2SFNet model effectively distinguishes between ground and non-ground points in the frame. By filtering out ground points, the quality of detected objects is enhanced by focusing on relevant features in the frame data above the ground level, which improves the accuracy of object-detection tasks.

The module leverages the height difference relationships in the point cloud to categorize the data. For any given point cloud P , the module calculates the height difference between p and its corresponding reference point P_r within the neighborhood. Ensuring symmetry in the specific classification criteria is achieved as follows:

$$Filtered\ Points(r, p) = \begin{cases} 0, & \text{if } \|z_i - z_j\| > t_h \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where z_i represents the height value of point P , z_j represents the height value of reference point P_r , and t_h is the height difference threshold. By calculating the height difference relationships between points, this model can determine the category of each point. When the result of Equation (4) is 0, point p is classified as an object and will be retained; otherwise, it is classified as a road point and will be filtered. A visualization of the 7D color sparse data within the image field of view is shown in Figure 4.

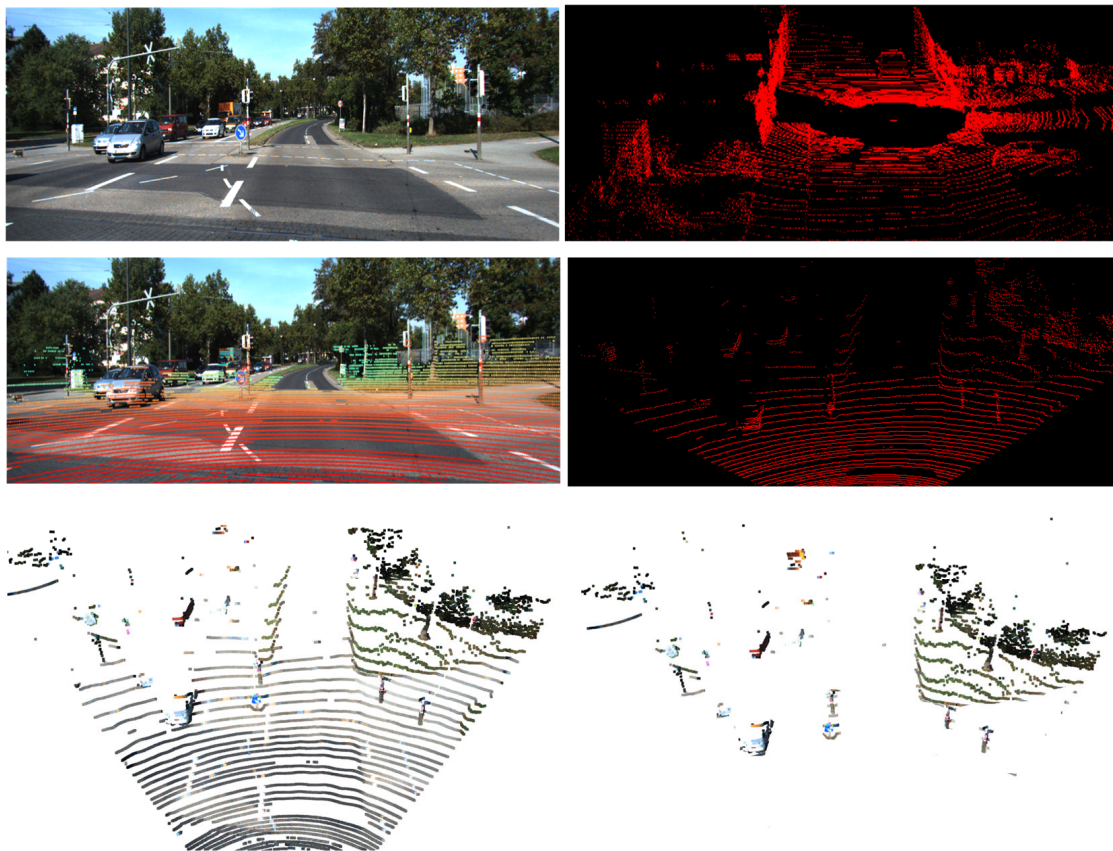


Figure 4. Some visual examples of RGB image, raw point cloud, calibrated image, point cloud in image FOV, colored point cloud in image FOV, and corresponding filtered colored point cloud.

As demonstrated in Figure 4, the RGB image provides the color and texture of the road environment, while the 3D point cloud presents spatial information about the scanned

object and its surrounding environment. The multi-dimensional colored data significantly heighten the color and texture of the 3D point cloud. It can be observed that the amount of information for pixels is considerably larger than the points in all categories. The seven-dimensional colored data established not only preserve the spatial sparsity, but they also enrich the surface color and texture of the points. By limiting the detection region, the computational cost of data processing is greatly reduced.

3.2. Pseudo-Image Generation

To enhance the visualization of LiDAR points, this module utilizes prior knowledge and spatial ensemble constraints to filter the generated 7D colored data. The 7D colored point cloud data are uniformly split into 2D grids with a shape of (H, W) and a resolution r . The non-empty grids are resampled into N pillar grids, and after attaching the color texture features, each pillar grid is represented as $(x_v, y_v, z_v, r_v, R_v, G_v, B_v)$ with additional color texture features. Therefore, in this paper, we took (C_{in}, N) as the input to the underlying PointNet network, where $C_{in} = 7$ and the output is (N, C_{out}) , from which a 2D pseudo-image with a size of (H, W, C_{out}) was created.

3.3. Dilated Feature Fusion Network

CNN models can learn features by several convolutional and pooling layers. Directly handling the extracted features may not be efficient for a complex environment. The previous method directly used bottom-up convolutional layers. Some valid features may be ignored during multiple layers. Inspired by atrous convolution [48], a novel dilated feature fusion network based on a pyramid [49,50] with the receptive field network was designed; the overall architecture is illustrated in Figure 5.

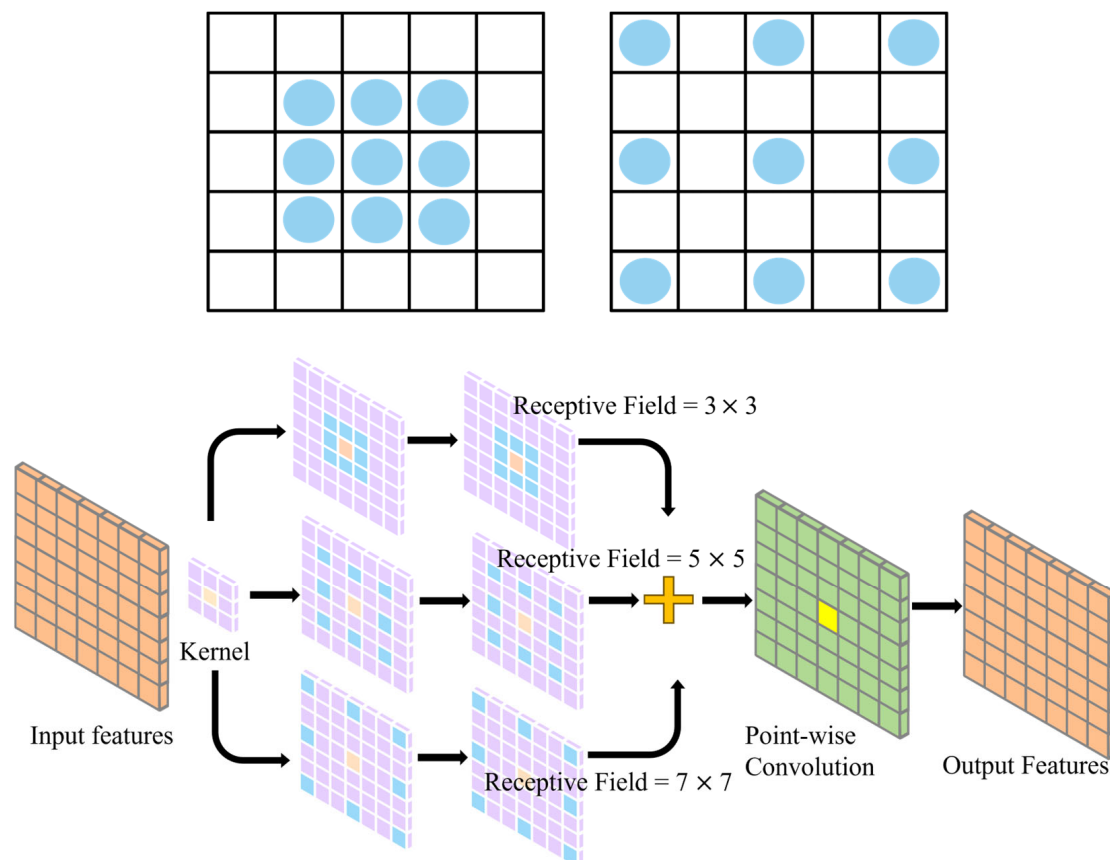


Figure 5. 1-Dilated Convolution, 2-Dilated Convolution and Framework of atrous convolution.

The feature fusion network consists of an encoder, receptive field expansion, and a decoder. The encoder is a pyramid feature extractor based on ResNet-50 and is used to acquire high-level features with multiple perceptual layers. Figure 6 shows a conception of how the receptive field changes according to the dilatation rate.

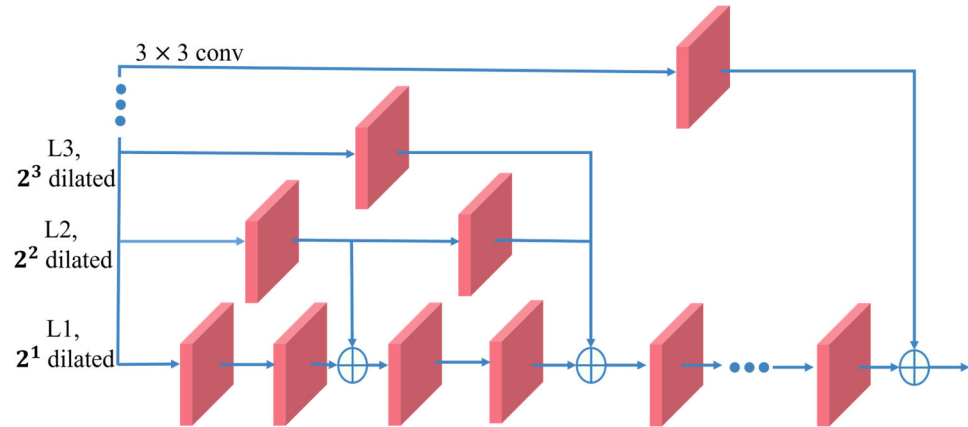


Figure 6. Framework of the receptive field network.

In the pyramid feature module, atrous convolutional layers with dilation rates of 1, 2, and 3 and 1×1 information are used; the output feature maps of all the layers are fused through addition. That is, the high-level features further fuse pyramid features with rich multiscale features for each level. For low and high levels, a dilated feature fusion network is used to obtain fused features. At this stage, different scale features with different receptive information are combined. Finally, a point-wise convolution is applied to the fused feature map. The term “conv” indicates a convolutional layer; “+” shows the concatenation of features. The simple decoder consists of ResBlock and up-sampling. Finally, the module reconstructs the fused features with more detailed information. The framework of the receptive field network is illustrated in Figure 6.

The skip connection operation is applied in the receptive field module due to the different dilated convolutions in each level. When the skip connection is denoted as T, the dilation size can be represented as $(t, t = 1, 2, 3)$, expanding the receptive field.

To attain an accurate object position and semantics, it is necessary to acquire the semantic texture of the object through a continuous down-sampling operation, and then concatenate high- and low-level feature maps to achieve multi-level fusion. This module heavily reduces the input by means of a ground-filtering module and removes certain critical points in the vicinity of the ground.

3.4. Loss Function

The overall loss function is designed to maintain symmetry in the evaluation of the model’s performance, similar to PointPillars [11] and SECOND [16]. It is composed of three parts: $smooth\ l_1$ loss for position regression, L_{cls} loss for object classification, and L_{dir} loss for direction (yaw angle), ensuring the symmetry in the criteria.

The 3D object is surrounded by a rectangle detection box whose parameters are defined by $(x, y, z, w, l, h, \theta)$, where x, y, z are the center coordinates of the rectangle box and w, l, h are the width, length, and height. In the current autopilot field, objects are on the ground, so only the yaw angle θ needs to be considered. The relevant parameters are listed as follows:

$$\Delta x = \frac{x_g - x_a}{d_a}, \Delta y = \frac{y_g - y_a}{d_a}, \Delta z = \frac{z_g - z_a}{d_a} \quad (5)$$

$$\Delta l = \log\left(\frac{l_g}{l_a}\right), \Delta h = \log\left(\frac{h_g}{h_a}\right), \Delta w = \log\left(\frac{w_g}{w_a}\right), \Delta \theta = \theta_g - \theta_a \quad (6)$$

where $\Delta x, \Delta y, \Delta z$ represent the offsets between the ground truth value x_g, y_g, z_g and the predicted values x_a, y_a, z_a ; l_g, h_g, w_g represent the length, height, and width of the ground

truth value; l_a, h_a, w_a represent the length, height, and width of the predicted values; and d_a represents the diagonal length. These offsets are normalized by the diagonal length d_a of the detection box to ensure consistent scaling and measurements across different sizes of boxes: $d_a = \sqrt{(l_a)^2 + (w_a)^2}$.

- (a) The regression position loss (L_{loc}) is calculated alongside the classification predictions, which are supervised using the cross-entropy (CE) loss.

$$smooth - l_1(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{others} \end{cases} \quad (7)$$

- (b) To address the issue of sample imbalance in traffic scenes, our model employed a focal loss function (object classification focal loss (L_{cls})). This function places greater emphasis on hard-to-classify examples and helps to alleviate the influence of abundant easy-to-classify samples, promoting a better performance across all object classes. The significant disparity between positive and negative sample ratios critically impacts the vehicle detection performance. Typically, the network generates around 7000 boxes and there are only a limited number of ground truths, with each instance yielding just 4 to 6 positives samples. This leads to an extreme imbalance between vehicle and background classes. To mitigate the challenge, focal loss is utilized, effectively focusing on hard-to-classify negative samples, thereby improving the performance.

$$L_{cls} = -\alpha(1-p)^\gamma \log(p) \quad (8)$$

where p represents the classification probability of the predicted box, α is a weighted factor used to balance the significance of positive and negative examples, and γ serves as a focusing parameter that adjusts the rate at which easy examples are down-weighted. For this implementation, α and γ were set to 0.25 and 2, respectively.

- (c) Directional loss (L_{dir}) addresses the challenge of angle regression, as the orientation between two possible directions $\{+, -\}$ cannot be inherently distinguished. To overcome this limitation, a softmax function was employed to calculate the discretized orientation loss. Specifically, if the heading angle around the Z-axis of the ground truth is greater than 0, it is classified as positive; otherwise, the orientation is negative.

By combining the losses discussed above, the overall loss function can be formulated as follows:

$$L = \frac{1}{N_{pos}} (L_{loc}\beta_{loc} + L_{cls}\beta_{cls} + L_{dir}\beta_{dir}) \quad (9)$$

where N_{pos} represents the number of accurately detected boxes and the weights β_{loc} , β_{cls} , and β_{dir} correspond to the contributions of the regression loss, classification loss, and direction loss, which were assigned the values of 2.0, 1.0, and 0.2, respectively.

4. Experiments

4.1. Dataset and Metrics

The KITTI [47] object dataset has become a widely recognized benchmark for evaluating the performance in autonomous driving. The dataset is divided into two parts: a training dataset containing ground truths, with a total of 7481 samples, and a test dataset without ground truths, comprising 7581 samples.

In addition to each dataset, KITTI contains sensor calibration and the corresponding point cloud for each image. It contains three annotated category labels, including vehicles, pedestrians, and cyclists, which are officially provided for evaluation. For the quantitative evaluation of each category, the data are classified into three difficulty levels: easy, moderate, and hard. These classifications are based on factors such as the size, visibility range, and truncation level, with a bounding box overlap (IOU threshold) of 70% for vehicles and 50% for the other categories. Furthermore, the most commonly used evaluation metrics

are the AP (average precision) and the mAP (mean average precision); the former reflects the precision for a specific category across the entire dataset, while the latter provides an overall average precision across all categories.

4.2. Implementation Details

This method used common settings and a limited range. For validation purposes, the training set was divided into two non-overlapping subsets: one for the training split, consisting of 3712 examples, and another for the validation split, comprising 3769 examples.

In the training phase, the model adopted the adaptive moment estimation (Adam) optimizer to train the 2SFNet with 300 epochs, and the batch size was 12. The momentum was adjusted within the range of 0.85 to 0.95, while the initial learning rate was established at 0.001 and the fixed-weight decay coefficient was 0.001 to assist in regularizing the module during training. For the receptive field module, the basic dilation rate was set to (1, 3, 5) to maintain symmetry in the feature extraction process. The model training and evaluation were executed with the Pytorch 1.6 DL framework on the local hardware platform with an NVIDIA RTX3090 GPU. And the detection visualization was achieved using the Mayavi tool, which can show the detection results in 2D and 3D space. Furthermore, several artificial intelligence software tools are suitable for 3D object detection with symmetry-aware colored point clouds, including TensorFlow and Open3D. These platforms can facilitate the development and implementation of advanced 3D object-detection algorithms. During model evaluation, this method uniquely utilizes a 7D colored point cloud as the input, while the others rely on the original 3D point cloud. Consequently, this approach requires minimal adjustments to the network, primarily involving modifications to the number of channels dedicated to processing the input.

4.3. Experimental Results

4.3.1. Experimental Comparison Between 7D Colored Point Cloud and 3D Point Cloud

To verify the effectiveness of the proposed method, this section further evaluates the 2SFNet with different combinations of processing, i.e., 3D point clouds and 7D colored point clouds, as shown in Table 1. Table 1 presents a comparison of the detection results from various input data types for the KITTI validation dataset. The first row indicates the results obtained using a 3D point cloud as the network input, while the second row corresponds to the results from the 7D colored point cloud. Notably, when employing a 3D point cloud, the input channel of the network was adjusted to three.

Table 1. Comparison results under different inputs for validation dataset (AP, %).

Input Data	Vehicles (IoU = 0.7)			Pedestrians (IoU = 0.7)			Cyclists (IoU = 0.7)		
	E	M	H	E	M	H	E	M	H
3D Point Cloud	86.37	76.87	74.45	58.03	56.71	52.31	63.65	57.42	54.09
7D Colored Point Cloud	88.71	77.10	75.17	59.31	58.08	53.19	64.98	59.17	55.15

The difficulty levels of “easy (E)”, “medium (M)”, and “hard (H)” are established by the official KITTI website and pertain to the three categories of vehicles, pedestrians, and cyclists. As illustrated in Table 1, the detection results using the 7D color point cloud as the input surpassed those obtained with the 3D data in all three categories: vehicles, pedestrians, and cyclists. Especially in the categories of pedestrians and cyclists, the accuracy improvement from using the 7D color point cloud was more obvious, with improvements of 1.37% and 1.75% at the difficulty level of “medium”, respectively. The reason for this is that color information enriches the semantic content. Additionally, the feature fusion network, which incorporates an expanded respective field, exhibited distinct advantages in detecting objects over long distances. To gain an intuitive understanding of the detection performance, this paper compared the predicted results with the ground truth by using a 3D bounding box, and the visualization results are presented in Figure 7.

The purple indicates the visualization result of the real label (ground truth), while the blue represents the visualization of the network's predictions.

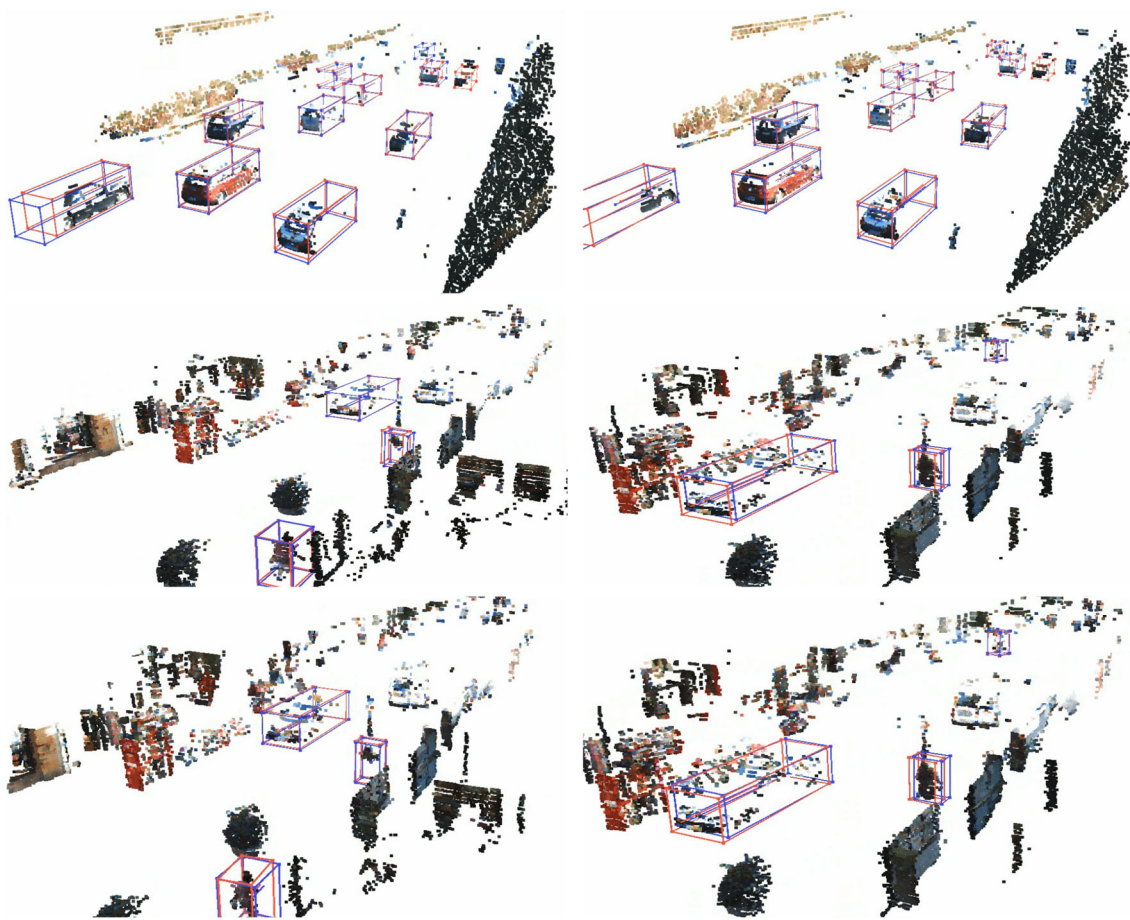


Figure 7. Visualization results for the 2SFNet model predictions and ground truths. The purple is the visualization result of the truth label and the blue is the visualization result of the network prediction.

As can be seen in Figure 7, the accuracy of the center point and the length, width, and aspect of the 3D bounding box were infinitely close to the ground truth. This approach could generate accurate predictions for challenging scenes, such as those involving partial occlusion and distant objects. This demonstrates that the detection network leveraging multimodal data fusion attained a high accuracy and can alleviate the common leakage of missed detections and false positives in traffic scenarios.

It can easily be observed that the dilated mapping range contains more information regarding the object and its surrounding environment, which can assist the network in making better predictions when the sparse point clouds are insufficient for the detection task.

4.3.2. Evaluation of KITTI Object Benchmark Test Dataset

The 2SFNet was evaluated using the KITTI test set and compared with other approaches. For the test with no labels, the predictions were obtained by submitting them to the official KITTI test server. This section evaluates the detection performance of the 2SF (with colored 7D data) in comparison with recently published 3D object-detection models, including LiDAR-based methods (BirdNet+ [7], PointPillars [11], PVRCNN [13], VoxelNet [15], SECOND [16], and SegVoxelNet [17]) and multimodal sensor fusion-based methods (MV3D [27], AVOD [28], MVAf-Net [31], MMF [32], MEnet [35], Contfuse [38], and F-PointNet [45]). The evaluation results (average precision, mAP) for the KITTI testing set of several 3D object detectors from the official KITTI leaderboard are reported in Table 2.

The difficulty levels of easy, moderate, and hard are based on definitions provided by the KITTI official website. The best results in each category are highlighted in bold.

The 3D detection results of the 2SFNet for the KITTI testing set are presented in Table 2. The proposed method achieved AP values of 88.31% and 77.85% for vehicles at the easy and moderate levels, respectively. Compared with our baseline of PointPillars [11], the 2SFNet exhibited a considerable improvement. In particular, the 2SFNet improved PillarsNet for vehicles by 5.73% and cyclists by 0.13%, which proved the effectiveness of the early-level fusion strategy. In comparison to LiDAR-only methods, the 2SFNet achieved the best results for these levels across all three categories, demonstrating effectiveness. Furthermore, the performance of the 2SFNet surpassed that of VoxelNet [15] by 10.84% and SECOND [16] by 3.02%, highlighting the superiority of our LiDAR–camera fusion module in delivering improved results.

In comparison to multi-sensor fusion-based methods, for the vehicle category, the 2SFNet outperformed existing techniques, with the exception of MMF [32] and MENet [35] at the easy level and MVAf-Net [31] at the moderate and hard levels. Specifically, the proposed method surpassed MV3D [27] by 13.34%, AVOD [28] by 5.28%, and MVAf-Net [31] by 0.44% at the easy level. For the pedestrian and cyclist categories, the 2SFNet drastically narrowed the performance gap between fusion approaches, with the exception of a marginal difference in the easy level for pedestrians. In the vehicle category, this method may not be the top performer, likely because objects at this level are relatively easy to detect with sufficient point clouds. In such cases, LiDAR–camera fusion may not be the best choice for enhancing the performance.

To visually compare the detection effects of the proposed method, the 2SFNet, and PointPillars, Figures 8 and 9 further provide a qualitative analysis by visualizing the detection results in the case of partial occlusion and distant objects. The left column is the visualization result of PointPillars, and the right is the visualization result of the 2SFNet. Red and pink highlights the critical regions in images space and 3D space. The purple indicates the vehicles, while blue represents the orientation.

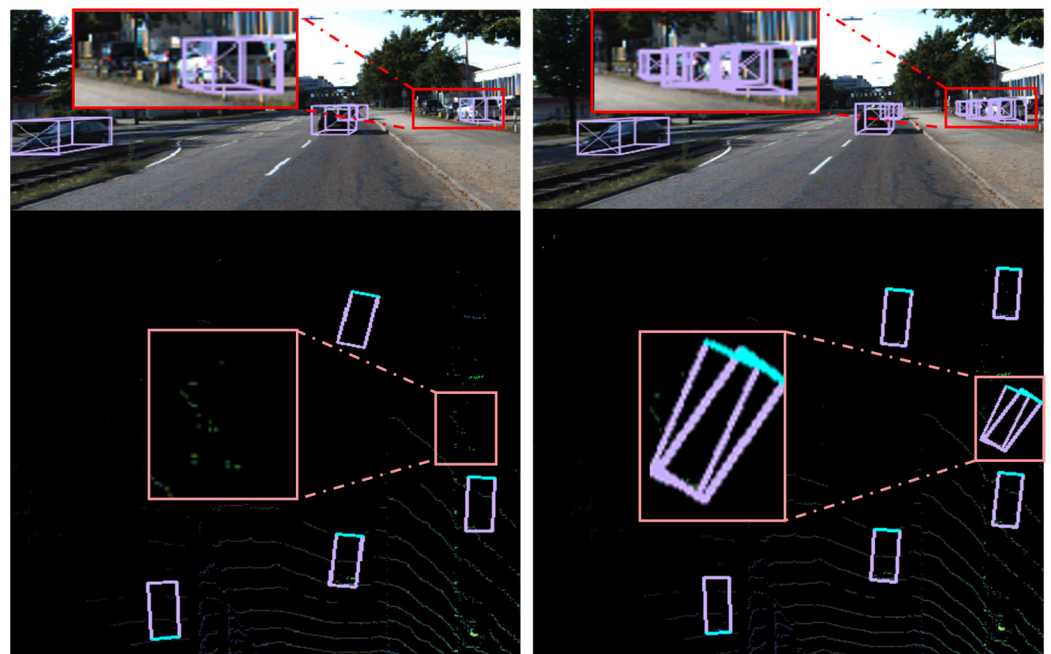


Figure 8. Visualization comparison for distant-object-detection results with PointPillars (left) and our 2SFNet (right). Notice that the predictions are entirely based on BEV maps derived from point clouds. Re-projecting to image space is for illustrative purposes only.



Figure 9. Visualization comparison for partially occluded-object-detection results with PointPillars (**left**) and our 2SFNet (**right**). Notice that predictions are entirely based on BEV maps derived from point clouds. Re-projecting to image space is for illustrative purposes only.

These figures include some qualitative visualizations for comparison between both the image view and a bird's-eye view on the KITTI object-detection test set.

The left columns and right columns display the detection performance of the 2SFNet and PointPillars for the vehicle class, respectively. The selected frames contain distant and partially occluded objects, which illustrates that, when an object is far away from an autonomous vehicle, the number of scanned point clouds is also extremely small and sparse. It can be observed that the LiDAR-only-based PillarsNet misidentified the orientation of the vehicle due to the similar geometric shapes and failed to detect the farthest vehicle.

This 2SFNet can clearly generate complete predictions under challenging scenes like those with crowded and faraway objects; the reason is that it has a superior capability regarding the effective use of image information and the fusion of sensors.

Table 2. Quantitative comparison of LIDAR-based and multimodal sensor fusion-based methods using the KITTI testing set. The results were evaluated by the mean average precision with the 40 recall position. L and C represent the LiDAR sensor and camera sensor, respectively (/%).

Methods	Sensor Modality	Time (s)	3D AP (%)								
			Vehicles			Pedestrians			Cyclists		
			Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
BirdNet+ [7]	L (BEV)	0.1	70.14	51.85	50.03	37.99	31.46	29.46	67.38	47.72	42.89
Pointpillars [11]	L (BEV)	0.016	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92
PVRCNN [13]	L (Point)	-	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
VoxelNet [15]	L (Voxel)	0.23	77.47	65.11	57.73	-	-	-	-	-	-
SECOND [16]	L (Voxel)	0.01	85.29	76.60	71.77	43.04	35.92	33.56	71.05	55.64	49.83
MVAFNet [31]	L + C	0.06	87.87	78.71	75.48	-	-	-	-	-	-
MMF [32]	L + C	0.05	88.40	77.43	70.22	-	-	-	-	-	-
MV3D [27]	L + C	0.36	74.97	63.63	54.00	-	-	-	-	-	-
AVOD [28]	L + C	0.08	83.07	71.76	65.73	50.46	42.27	39.04	63.76	50.55	44.93
F-PointNet [45]	L + C	0.17	82.19	69.79	60.59	50.53	42.15	38.08	72.27	56.12	49.01
MENet [35]	L + C	-	89.41	78.82	78.36	74.79	66.23	59.80	85.04	66.27	62.73
2SFNet (ours)	L + C	0.01	88.31	77.85	75.13	50.74	43.36	40.71	77.23	66.07	53.01

In these circumstances, it can be concluded that, in a complex and crowded environment, a LiDAR-only-based method has a tendency to miss objects owing to the absence of color information, while the 2SFNet ensures a higher detection accuracy by taking advantage of color and texture.

There are several improvements to be considered for the future. First, advanced deep learning techniques should be explored, such as attention mechanisms and generative models, to further enhance the performance of the 2SFNet model. Second, although the early fusion strategy has been taken into consideration, the combination of various fusion strategies should be considered in future research. Finally, an expanded dataset that includes more diverse scenes and object classes should be considered; this could improve the model's applicability in real-world situations.

5. Conclusions

This paper proposed a LiDAR–camera fusion-based detection network, the 2SFNet, and explored the use of symmetry-aware colored point clouds for 3D object detection. It has significant practical applications in fields such as autonomous driving, robotics, and augmented reality. A symmetry-aware 2SFNet mode consists of a filtered colored point-cloud-generation module, a pseudo-image-generation module, and a dilated feature fusion network. In the former, the 2SFNet utilizes an early fusion module and a ground-height-filtering module to construct non-ground colored point cloud data. The early fusion module leverages the symmetry in the spatial relationship between image pixels and points, while the ground-height-filtering module maintains the symmetry of the relevant points, focusing on object-centric information. Subsequently, the colored point cloud data are uniformly divided into 2D grids, allowing the dilated feature fusion network to find corresponding features with a matched scale and a receptive field. Experiments and evaluations were conducted using the KITTI dataset to demonstrate the effectiveness of the 2SFNet.

The 2SFNet can effectively operate across various driving scenarios, including urban, highway, and rural environments, by utilizing LiDAR and camera data. However, its performance may fluctuate based on the driving conditions, and adverse weather such as heavy rain, snow, and fog can impact the sensor reliability.

To maximize the potential of symmetry-aware detection methods, several areas warrant further investigation. For instance, these areas include developing more sophisticated algorithms that can identify and exploit different types of symmetry (e.g., reflective, rotational) within complex point clouds, which may improve the detection rates in cluttered environments. Second, investigating various fusion strategies can be integrated with other

data modalities, such as 2D images or depth maps, to create a more holistic detection framework that benefits from the strengths of multiple data sources. Finally, extensive experiments should be conducted across various datasets and real-world scenarios to evaluate the generalizability of symmetry-aware detection methods.

Author Contributions: Conceptualization, L.W. and P.Z.; methodology, L.W. and M.L.; software, L.W.; validation, L.W. and M.L.; formal analysis, L.W. and P.Z.; investigation, L.W.; writing—original draft preparation, F.Z., L.W. and L.W.; writing—review and editing, F.Z., L.W. and P.Z.; project administration, L.W.; funding acquisition, L.W. and L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was sponsored by the Start-Up Fund for New Ph.D. Researchers of the Suzhou Chien-Shiung Institute of Technology (2023); the Jiangsu Basic Science Foundation of Colleagues and Universities (nature science), China (grant No. 24KJD520009); the Jiangsu “Qing Lan Project” Excellent Young Teacher Grant ([2023]27); and the National Nature Science Foundation of China (grant No. 12401679).

Data Availability Statement: The code is available at <https://github.com/baka-19C/2SFnet> (accessed on 11 November 2024). We used the open-source data for KITTI, which are available online at https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d (accessed on 10 August 2019). The datasets generated during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, L.; Huang, Y. A Survey of 3D Point Cloud and Deep Learning-Based Approaches for Scene Understanding in Autonomous Driving. *IEEE Intell. Transp. Syst. Mag.* **2021**, *14*, 135–154. [[CrossRef](#)]
2. Wang, X.; Li, K.; Chehri, A. Multi-Sensor Fusion Technology for 3D Object Detection in Autonomous Driving: A Review. In *IEEE Transactions on Intelligent Transportation Systems*; IEEE: Piscataway, NJ, USA, 2024; Volume 25, pp. 1148–1165. [[CrossRef](#)]
3. Li, B.; Zhang, T.; Xia, T. Vehicle Detection From 3D LiDAR Using Fully Convolutional Network. Robotics: Science and Systems Foundation. *arXiv* **2016**, arXiv:1608.07916.
4. Minemura, K.; Liau, H.; Monroy, A.; Kato, S. LMNet: Real-time Multiclass Object Detection on CPU Using 3d LiDAR. In Proceedings of the 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Singapore, 21–23 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 28–34.
5. Zhou, J.; Tan, X.; Shao, Z.; Ma, L. FVNet: 3D front-view proposal generation for real-time object detection from point clouds. In Proceedings of the 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Suzhou, China, 19–21 October 2019; pp. 1–8.
6. Beltr'an, J.; Guindel, C.; Moreno, F.M.; Cruzado, D.; Garc, F.; De La Escalera, A. Birdnet: A 3D Object Detection Framework From LiDAR Information. In Proceedings of the IEEE Conference on Intelligent Transportation Systems Conference (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3517–3523.
7. Barrera, A.; Guindel, C.; Beltran, J.; García, F. BirdNet+: End to-end 3D object detection in LiDAR bird’s eye view. In Proceedings of the IEEE Conference on Intelligent Transportation Systems Conference (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6.
8. Simon, M.; Milz, S.; Amende, K.; Gross, H.M. Complex-yolo: Real-time 3d object detection on point clouds. *arXiv* **2018**, arXiv:1803.06199.
9. Zeng, Y.; Hu, Y.; Liu, S.; Ye, J.; Han, Y.; Li, X.; Sun, N. Rt3d: Real-time 3-D Vehicle Detection in LiDAR Point Cloud for Autonomous Driving. *IEEE Rob Auto Lett.* **2018**, *3*, 3434–3440. [[CrossRef](#)]
10. Yang, B.; Luo, W.; Urtasun, R. PIXOR: Real-time 3D Object Detection from Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7652–7660.
11. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast Encoders for Object Detection from Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 15 June 2019; pp. 12697–12705.
12. Yang, B.; Liang, M.; Urtasun, R. HDNET: Exploiting HD maps for 3D Object Detection. In Proceedings of the 2nd Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018; pp. 146–155.
13. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PVRCNN: Point-voxel feature set abstraction for 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 19 June 2020; pp. 10526–10535.
14. Graham, B.; van der Maaten, L. Submanifold Sparse Convolutional Networks. *arXiv* **2017**, arXiv:1706.01307.
15. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18 June 2018; pp. 4490–4499.

16. Yan, Y.; Mao, Y.; Li, B. SECOND: Sparsely Embedded Convolutional Detection. *Sensors* **2018**, *18*, 3337. [[CrossRef](#)] [[PubMed](#)]
17. Yi, H.; Shi, S.; Ding, M.; Sun, J.; Xu, K.; Zhou, H.; Wang, Z.; Li, S.; Wang, G. SegVoxelNet: Exploring Semantic Context and Depthaware Features for 3D Vehicle Detection From Point Cloud. In Proceedings of the International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 2274–2280.
18. Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; Li, H. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection. *Int. J. Comput. Vis.* **2022**, *131*, 531–551. [[CrossRef](#)]
19. Hu, J.S.K.; Kuai, T.; Waslander, S.L. Point density-aware voxels for LiDAR 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18 June 2022; pp. 8459–8468.
20. Charles, R.Q.; Su, H.; Mo, K.; Leonidas, J.G. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–27 July 2017; pp. 77–85.
21. Chen, Y.; Yang, Z.; Zheng, X.; Chang, Y.; Li, X. Pointformer: A dual perception attention-based network for point cloud classification. In Proceedings of the Asian Conference on Computer Vision, Macao, China, 4–8 December 2022; pp. 3291–3307.
22. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D object proposal generation and detection from point cloud. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 15 June 2019; pp. 770–779.
23. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. PointPainting: Sequential Fusion for 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 19 June 2020; pp. 4604–4612.
24. Xie, C.; Lin, C.; Zheng, X.; Gong, B.; Liu, H. Dense sequential fusion: Point cloud enhancement using foreground mask guidance for multimodal 3-D object detection. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–15. [[CrossRef](#)]
25. Wen, L.H.; Jo, K.H. Fast and Accurate 3D Object Detection for Lidar-Camera-Based Autonomous Vehicles Using One Shared Voxel-Based Backbone. *IEEE Access* **2021**, *9*, 22080–22089. [[CrossRef](#)]
26. Wang, C.; Ma, C.; Zhu, M.; Yang, X. Pointaugmenting: Cross-modal augmentation for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 11794–11803.
27. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D object detection network for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
28. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3D proposal generation and object detection from view aggregation. In Proceedings of the IEEE Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–8.
29. Yoo, J.H.; Kim, Y.; Kim, J.; Choi, J.W. 3D-CVF: Generating joint camera and LiDAR features using cross-view spatial feature fusion for 3D object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 720–736.
30. Raffiee, A.; Irshad, H. Class-specific Anchoring Proposal for 3D Object Recognition in LIDAR and RGB Images. *arXiv* **2019**, arXiv:2011.00652.
31. Wang, G.; Tian, B.; Zhang, Y.; Chen, L.; Cao, D.; Wu, J. Multi-View Adaptive Fusion Network for 3D Object Detection. *arXiv* **2020**, arXiv:2011.00652.
32. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-task multi-sensor fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7345–7353.
33. Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.; Han, S. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. *arXiv* **2022**, arXiv:2205.13542.
34. Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; Tang, Z. Bevfusion: A simple and robust lidar-camera fusion framework. *arXiv* **2022**, arXiv:2205.13790.
35. Liu, M.; Chen, Y.; Xie, J.; Zhu, Y.; Zhang, Y.; Yao, L.; Bing, Z.; Zhuang, G.; Huang, K.; Zhou, J.T. MENet: Multi-Modal Mapping Enhancement Network for 3D Object Detection in Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 9397–9410. [[CrossRef](#)]
36. Yin, J.; Shen, J.; Chen, R.; Li, W.; Yang, R.; Frossard, P.; Wang, W. Is-fusion: Instance-scene collaborative fusion for multimodal 3d object detection. *arXiv* **2024**, arXiv:2403.15241.
37. Zhang, W.; Shi, H.; Zhao, Y.; Feng, Z.; Lovreglio, R. MMAF-Net: Multi-view multi-stage adaptive fusion for multi-sensor 3D object detection. *Expert Syst. Appl.* **2024**, *242*, 122716. [[CrossRef](#)]
38. Li, X.; Shi, B.; Hou, Y.; Wu, X.; Ma, T.; Li, Y.; He, L. Homogeneous multi-modal feature fusion and interaction for 3D object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022.
39. Bai, X.; Hu, Z.; Zhu, X.; Huang, Q.; Chen, Y.; Fu, H.; Tai, C.-L. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022; pp. 1090–1099.
40. Jiao, Y.; Jie, Z.; Chen, S.; Chen, J.; Wei, X.; Ma, L.; Jiang, Y.-G. Msmdfusion: Fusing lidar and camera at multiple scales with multidepth seeds for 3d object detection. *arXiv* **2022**, arXiv:2209.03102.
41. Liu, H.; Liao, K.; Lin, C.; Zhao, Y.; Guo, Y. Pseudo-LiDAR Point Cloud Interpolation Based on 3D Motion Representation and Spatial Supervision. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 6379–6389.

42. Gu, S.; Yang, J.; Kong, H. A cascaded lidar-camera fusion network for road detection. In Proceedings of the International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 13308–13314.
43. Liang, M.; Yang, B.; Wang, S.; Urtasun, R. Deep continuous fusion for multi-sensor 3D object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 641–656.
44. Wang, Z.; Zhan, W.; Tomizuka, M. Fusing bird's eye view lidar point cloud and front view camera image for 3d object detection. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
45. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum PointNets for 3D object detection from RGB-D data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 17–23 June 2018; pp. 918–927.
46. Wang, Z.; Jia, K. Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 1742–1749.
47. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
48. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122. Available online: <http://arxiv.org/abs/1511.07122> (accessed on 9 December 2024).
49. Raza, A.; Huo, H.; Fang, T. PFAF-Net: Pyramid Feature Network for Multimodal Fusion. *IEEE Sens. Lett.* **2020**, *4*, 5501704. [[CrossRef](#)]
50. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.