World Electric Vehicle Journal

MDPI

Article

A Multi-Sensor 3D Detection Method for Small Objects

Yuekun Zhao, Suyun Luo *, Xiaoci Huang and Dan Wei

School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; m310121417@sues.edu.cn (Y.Z.); h2128249@163.com (X.H.); weiweidandan@163.com (D.W.)

* Correspondence: luosuyun196@sues.edu.cn; Tel.: +86-135-8574-7855

Abstract: In response to the limited accuracy of current three-dimensional (3D) object detection algorithms for small objects, this paper presents a multi-sensor 3D small object detection method based on LiDAR and a camera. Firstly, the LiDAR point cloud is projected onto the image plane to obtain a depth image. Subsequently, we propose a cascaded image fusion module comprising multi-level pooling layers and multi-level convolution layers. This module extracts features from both the camera image and the depth image, addressing the issue of insufficient depth information in the image feature. Considering the non-uniform distribution characteristics of the LiDAR point cloud, we introduce a multi-scale voxel fusion module composed of three sets of VFE (voxel feature encoder) layers. This module partitions the point cloud into grids of different sizes to improve detection ability for small objects. Finally, the multi-level fused point features are associated with the corresponding scale's initial voxel features to obtain the fused multi-scale voxel features, and the final detection results are obtained based on this feature. To evaluate the effectiveness of this method, experiments are conducted on the KITTI dataset, achieving a 3D AP (average precision) of 73.81% for the hard level of cars and 48.03% for the hard level of persons. The experimental results demonstrate that this method can effectively achieve 3D detection of small objects.

Keywords: three-dimensional object detection; autonomous vehicles; deep learning; multi-sensor

Citation: Zhao, Y.; Luo, S.; Huang, X.; Wei, D. A Multi-Sensor 3D Detection Method for Small Objects. *World Electr. Veh. J.* **2024**, *15*, 210. https://doi.org/10.3390/wevj15050210

Academic Editor: Michael Fowler

Received: 4 March 2024
Revised: 22 April 2024
Accepted: 7 May 2024
Published: 10 May 2024

1. Introduction

With the rapid pace of urbanization and continuous advancements in technological innovation, smart cities have emerged as a prominent trend in contemporary social development [1]. In the realm of urban transportation, autonomous driving systems are gradually evolving into a pivotal direction for development, representing a significant technological breakthrough. Through the integration of autonomous driving technology, vehicles can achieve intelligent navigation, self-driving capabilities, and automated hazard avoidance mechanisms. Consequently, this facilitates a reduction in traffic accidents while simultaneously enhancing transportation efficiency and mitigating energy consumption and emissions [2].

In the current scenarios of autonomous driving or complex robot navigation, precise detection results play a pivotal role in system performance. Despite significant advancements in 2D object detection technology driven by deep learning, it still falls short in providing precise spatial positioning information and accurately estimating the physical dimensions of objects within 3D space. In comparison to 2D object detection methods, 3D object detection technology provides advanced capabilities in accurately detecting object attributes such as precise position and distance, thereby introducing a novel avenue for research in environmental perception.

Through dependable 3D object detection technology, the autonomous driving system can make more accurate decisions and controls, thereby enhancing driving safety and reliability while effectively adapting to intricate and dynamic traffic environments. Currently,

numerous well-established implementation methods exist for 3D object detection; however, they encounter challenges and difficulties in detecting small objects.

Small objects typically refer to objects with few pixels on the image plane and relatively distant distances in the world coordinate system. The precise detection of the 3D center and size of small objects still present challenges, primarily due to the small proportion that small objects occupy within the entire detection range, the sparse characteristics of the point cloud, and the potential presence of occlusion or incomplete information in the objects to be detected.

This paper aims to address the issue of poor accuracy in the 3D detection of small objects by proposing a model that combines a LiDAR point cloud and camera images. In real-world autonomous driving scenarios, small objects are often challenging to detect, and single sensors typically suffer from limited information and poor robustness in complex environments. Our approach addresses these issues by optimizing the 3D detection of small objects in perceptual scenes, making our method more suitable for autonomous driving perception tasks. The main contributions of this paper are as follows:

1. In order to integrate depth information into image features, we propose a cascaded image fusion module to integrate depth information into image features, enhancing their representational capacity;
2. In order to detect small objects by capturing grid features of different scales, we introduce a multi-scale voxel module to extract features from the point cloud at different scales and fuse them with features from corresponding scale images;
3. To validate the proposed methods in this paper, experiments are conducted on the KITTI dataset, comparing them with current state-of-the-art algorithms.

2. Related Work

2.1. Camera-Based 3D Detection Methods

Three-dimensional object detection can be categorized into single-sensor and multi-sensor methods, with single-sensor options including camera and LiDAR. The most direct approach involves employing neural networks to estimate the 3D box parameters from the image directly. These methods draw inspiration from the architectural design of 2D object detection networks such as fast RCNNs [3], which have demonstrated efficacy in facilitating end-to-end training. FCOS3D [4] introduces a 2D guided detection module for 3D object detection. M3DRPN [5] is an innovative approach that leverages the intrinsic relationship between 2D and 3D information through the utilization of multiple 2D convolutional layers with independent weights. This enables the extraction of features at specific spatial locations, facilitating simultaneous detection in both 2D and 3D domains.

In order to enhance the accuracy of detection results, some methods such as AM3D [6] and DD3D [7] employ pre-trained auxiliary depth estimation networks. Initially, pretrained depth estimators were utilized for generating pseudo-point clouds. Subsequently, a 3D object detection approach based on point clouds or coordinates was employed for pseudo-point cloud prediction.

Although camera-based methods are cost-effective, their performance is typically inferior to LiDAR-based and multi-sensor methods due to the lack of depth information. While images can provide some information regarding the position and appearance features of objects, cameras are not as effective as LiDAR in delivering precise 3D positioning and spatial information. Consequently, camera-based 3D detection methods often exhibit subpar accuracy in terms of bounding box estimation.

2.2. LiDAR-Based 3D Detection Methods

According to different implementation routes, 3D detection methods based on LiDAR can be divided into three categories: point cloud methods, voxelization methods, and depth map methods. PointNet [8] and PointNet++ [9] are methods that directly extract features from the point cloud, enabling tasks such as classification and segmentation of the point cloud. The point cloud methods employed by these two methods serve as the foundation

for other 3D object detection methods based on point clouds. Additional direct approaches for 3D object detection using point clouds include Point2Seq [10] and 3DSSD [11].

Voxelization methods such as VoxelNet [12], SECOND [13], and Pointpillars [14] extract features from the LiDAR point cloud in standardized grids. Compared to point cloud methods, voxelization methods offer faster detection speed and are better suited for autonomous driving scenarios. Pyramid RCNNs [15], CenterPoint [16], and Point RCNNs [17] aim to enhance detection performance through a two-stage process. Firstly, 3D candidate regions are generated in a pyramid structure, and then fine detection of these regions is performed in the second stage.

Additionally, methods like LaserNet [18] utilize depth images for 3D object detection, implementing convolutional operators more suitable for depth images. Depth map methods effectively address the issue of point cloud sparsity while compromising the 3D local relationships within the LiDAR point cloud.

The circumferential distribution of laser rays emitted by LiDAR leads to an uneven spatial distribution of point clouds, resulting in varying densities of the same object type at different distances. This phenomenon presents a challenge for the feature extraction process in 3D object detection. Hence, LiDAR-based methods face challenges in detecting small objects and distant scenes due to the sparsity of the point cloud.

2.3. Multi-Sensor 3D Detection Methods

Multi-sensor 3D detection methods enable the integration of information from diverse sensors, offering solutions to address challenges encountered in LiDAR and camera-based detection methods. The synergistic combination of image and point cloud features exemplifies the significance of sensor fusion, while the integration of multi-sensor aids in mitigating single-sensor failures and enhancing adaptability across diverse environments.

F-PointNet [19] and F-ConvNet [20] represent two-stage fusion approaches; a 2D object detection method like Fast RCNN [3] is initially employed to extract the ROI (region of interest) from the image. Subsequently, a point cloud 3D detector is utilized to perform 3D object detection on the frustum region corresponding to the ROI.

MV3D [21] and AVOD [22] are representative works based on 3D candidate boxes, which fuse ROI features obtained from the camera and LiDAR at the feature level. AVOD initially projects the 3D point cloud onto the BEV (bird's eye view) perspective, generating candidate boxes for 3D objects based on this view. Subsequently, the 3D candidate boxes are projected onto both the front view of the point cloud and image view; then, corresponding features are fused by ROI pooling [3].

Pointpainting [23] and Fusionpainting [24] employ image segmentation networks to extract semantic features from images, which are then integrated with point clouds for enhanced fusion between the two modalities. Subsequently, a point cloud-based object detection method is utilized to detect objects in the enhanced data. Overall, sensor fusion plays a vital role in enhancing the accuracy and robustness of 3D object detection.

However, the aforementioned fusion methods encounter some challenges in real-world scenarios. For instance, F-ConvNet [20] necessitates a series connection of two sensors for execution, while Pointpainting [23] exhibits limited efficiency in harnessing image features. Furthermore, these approaches are not optimized for detecting small objects. Our work achieves bidirectional enhancement between point cloud and image information, fully leveraging multi-sensor data fusion. Considering the difficulty of using a single-scale voxel in complex scenes, we propose a multi-scale voxel module to effectively address object detection requirements at different scales within the scene.

3. Scheme Design

3.1. Cascade Image Fusion Module

It is challenging to directly obtain the distance from the camera to the objects using image features. However, depth features are crucial for 3D object detection. This paper introduces a simple and efficient cascaded image fusion module (CIF) that can extract

features effectively from both the LiDAR point cloud and the camera image, overcoming the issue of lacking depth information in the camera image.

The structure of the cascaded image fusion module is illustrated in Figure 1. It consists of two branches: depth and image. The module takes the RGB image and the depth image obtained from the point cloud as input. The camera image has a size of (416, 1344, 3), while the depth image is generated by projecting the point cloud onto the image plane and has a size of (416, 1344, 1). Each pixel value in the depth image represents the depth value at the corresponding position. The cascaded image fusion module is based on a MobileNetV3-s [25] backbone. Number [0–12] represents different output layers in MobilNetv3, including bottleneck module and convolution module. The channel attention module in this structure allows for the dynamic allocation of attention weights to different sensors by learning adaptive channel weights.

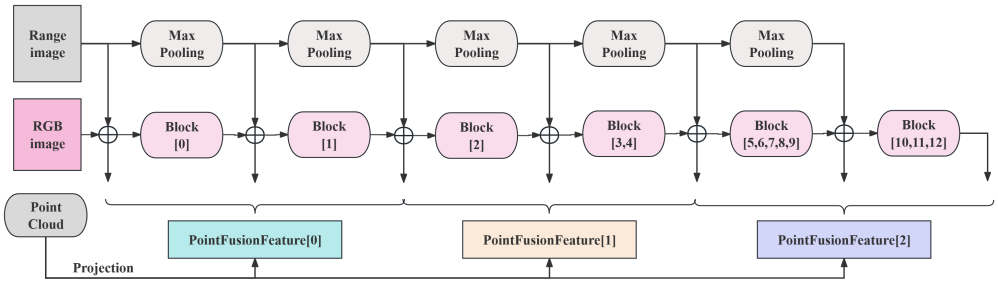

Figure 1. The structure of the cascade image fusion module.

The image branch consists of 11 layers, with downsampling applied in layers [1, 2, 4, 7, 9]. Similarly, the depth branch performs five maxpooling operations on the depth image and merges the results with the image branch through additional channels for feature extraction. The cascaded image fusion module generates a final output of seven-level image features. These features are grouped into three sets [0, 1, 2], [2, 3, 4], and [4, 5, 6], which are then connected to the corresponding 3D point vectors, resulting in three-level fused point features.

The cascaded image fusion module can perform depth completion on image features. Compared to fusion methods that rely solely on image features, this fusion approach integrates features of specific depth levels during the model's autonomous learning process, thereby reducing the overall loss of training.

3.2. Multi-Scale Voxel Module

Due to the non-uniform distribution of point clouds in 3D space, a single-size voxel mesh may not adequately represent all scene information. Previous methods such as SECOND [13] and Pointpillars [14] use different scale encoding structures for different object categories to address this issue. For example, larger voxels may be used for larger objects like vehicles, while smaller voxel sizes are employed for smaller objects such as pedestrians.

However, this method proves less efficient when dealing with multi-category tasks. This paper proposes a multi-scale voxel module (MSV) that covers voxel sizes of three scales for direct multi-scale voxel encoding of point clouds. The larger-scale grid can capture features over a larger range, while the smaller-scale grid increases the number of voxels within the same range of point clouds to enhance the model's ability to capture fine-grained features.

The voxel sizes for the three scales set in this paper are (0.08, 0.08, 0.4), (0.16, 0.16, 0.4), and (0.32, 0.32, 0.4) meters, respectively. The voxel sizes across the three scales are multiplied in the width and height dimensions while keeping the height consistent. Points within different scales of voxels are dynamically voxel encoded to obtain initial voxel features. Subsequently, three sets of initial voxel features are fused with three-level fusion point features. The fusion point features at lower levels contain rich original image information, which benefits the

learning ability for details when matched with small-scale voxels. The fusion point features at higher levels contain more deep semantic information, which enhances local perception when matched with larger voxels.

The multi-scale voxel module utilizes two sets of VFE layers [12] to extract initial voxel features for the three scales of voxels. Algorithm 1 demonstrates the overall procedure of the VFE module. After the second VFE layer, the voxel features are fused at the voxel level with multi-level fusion point features from the cascaded fusion image module.

Algorithm 1 Voxel Feature Encoding (VFE)

Require: Point cloud $P = \{p_1, p_2, \ldots, p_N\}$, where each point p_i includes its position (x, y, z)
Ensure: Set of voxel features V

```
Divide the point cloud P into a voxel grid G
Initialize an empty set of voxel features V
for each non-empty voxel g_i in G do
    Extract all points {p_i1, p_i2, ..., p_iM} within voxel g_i
    Initialize voxel feature v_i to zero vector
    for each point p_ij in g_i do
        Compute point feature f_ij (position of the point relative to the voxel center)
        Accumulate point features: v_i = v_i + f_ij
    end for
    Normalize v_i to obtain the final feature for voxel g_i
    Add the voxel feature v_i to the set V
end for
return the set of voxel features V
```

3.3. Fusion Pipeline

The fusion pipeline in this paper consists of three types of fusion: cascaded image fusion of depth and RGB images, point-level fusion with image features attached to the point cloud, and voxel-level fusion with initial voxel features and multi-level fusion point features. The fusion pipeline is illustrated in Figure 2.

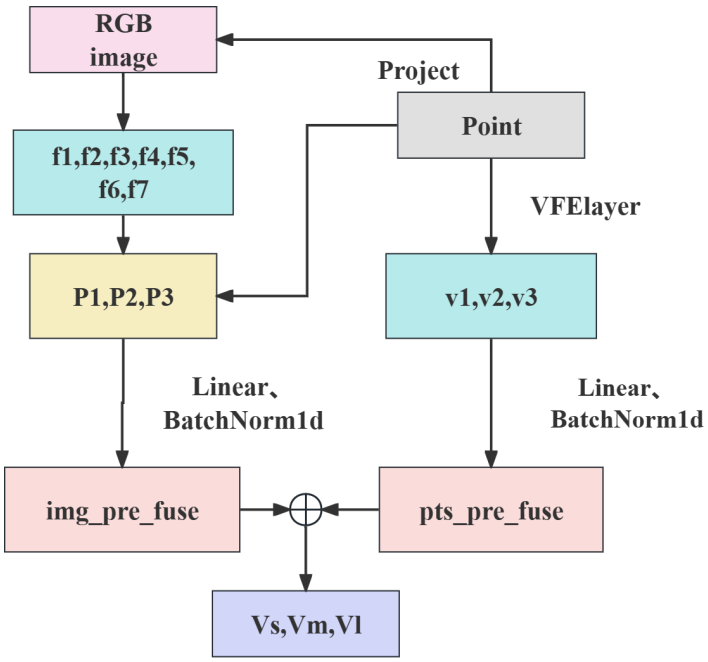

Figure 2. The data pipeline of the network.

Assume a 3D point set is P, the eigenmatrix is K, and the original image is I. The fusion image feature process can be represented as follows:

$$\sigma(Proj(K, M, P), I) \rightarrow f(1-7), \tag{1}$$

where $Proj(\cdot)$ represents projecting the point cloud onto the image plane to generate a point cloud depth image; $\sigma(\cdot)$ represents the feature extraction process of the cascaded image fusion module; $f(1-7)$ represents the output of image features at seven levels. Then, image feature output is divided into three groups, which are fused with the point cloud. The process of obtaining the three level fused point features can be expressed as follows:

$$\mathbf{P} \cdot (F_s, F_m, F_l) \rightarrow (P_s, P_m, P_l), \tag{2}$$

where $\mathbf{P}$ represents the point cloud projected onto the image plane, F_s, F_m, F_l represents three sets of image feature outputs, and P_s, P_m, P_l represents the three-level fusion point features obtained through point level fusion. Finally, fusion point features pass through the FC (fully connected) layer and norm layer to adjust the channel size with initial voxel features and then numerically add them to obtain fused voxel features. The process can be abbreviated as follows:

$$\alpha((v_1, v_2, v_3) \oplus (P_s, P_m, P_l)) \rightarrow (V_s, V_m, V_l), \tag{3}$$

where $\alpha(\cdot)$ represents the activation function, v_1, v_2, v_3 represents initial voxel features at three scales, $\oplus$ represents the sum of voxel features and fusion point features in the corresponding dimension values, and V_s, V_m, V_l represents the fused voxel features of three different scales in the final output.

3.4. Overall Structure

The overall structure of the model, as shown in Figure 3, can be divided into four parts: a cascaded image fusion module, a multi-scale voxel module, a 3D convolution feature integration module, and a standard 3D detector.

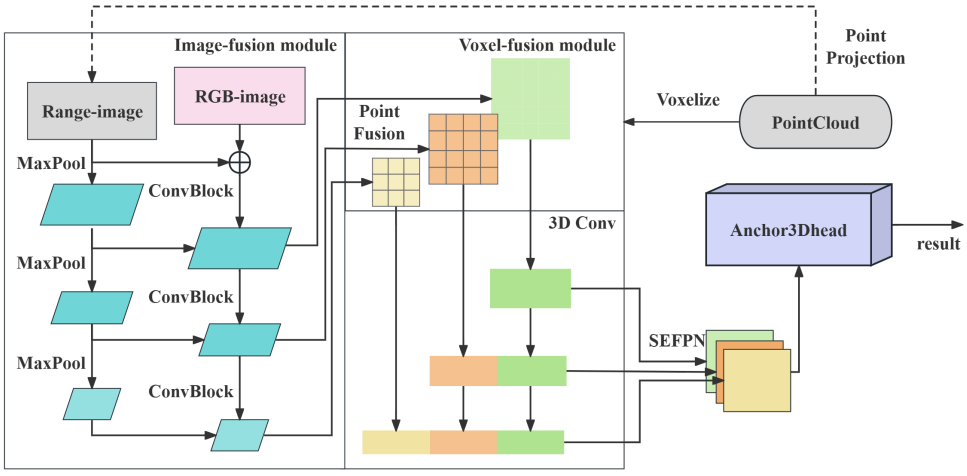

Figure 3. The structure of the network.

The cascaded image fusion module takes depth and RGB images as inputs and generates seven-level image features. These features are then fused with the original point cloud to output three-level fused point features. The multi-scale voxel module encodes the point cloud into voxel representations at three different scales, and the three sets of initial voxel features are fused with the three-level fused point features, respectively, resulting in three sets of fused voxel features.

The 3D convolution feature integration module enhances feature extraction by adjusting the size of the fused voxel features through sparse convolution and SEFPN [13] modules. These enhanced features are combined to form the final fusion feature. The 3D detector utilizes a detection head based on 3D anchor boxes to obtain the final detection results.

The loss function consists of three parts: boundary regression loss, orientation classification loss, and class loss. The boundary regression loss uses the SmoothL1 [3] loss function to ensure smooth boundary localization, while the orientation classification loss uses the Softmax loss. The class loss uses the focal loss [26] function to balance positive and negative samples. The loss function can be represented as follows:

$$L_{total} = (\beta_{loc} L_{loc} + \beta_{cls} L_{cls} + \beta_{dir} L_{dir}), \quad (4)$$

where β_{loc}, β_{cls}, and β_{dir} are the hyper parameters to provide weightage for different losses. For the loss function, the regression loss is set to 2.0, the focal loss hyperparameter is set to 1.0, and the class loss for angles is set to 0.2.

4. Details and Experiments

4.1. KITTI Dataset

The effectiveness of each module was assessed in our work using the KITTI [27] dataset, which encompasses image data captured across diverse environments, including urban and rural areas, as well as highways. Each image within the dataset can exhibit a maximum of 15 vehicles and 30 pedestrians, along with varying levels of occlusion and truncation. The dataset comprises a total of 7481 training samples and 7518 test samples, consisting of RGB image frames and LiDAR point cloud with 64 lines.

Figure 4 illustrates an example from the KITTI dataset. On the left, the point cloud and ground truth annotations are displayed from a bird's-eye-view. The top-right part displays the LiDAR point cloud projected onto the camera image, while the bottom-right part displays the 3D ground truth overlaid onto the camera image. These three components are from the same frame in the dataset. The ground truth of objects of different difficulty levels are represented by boxes of different colors.

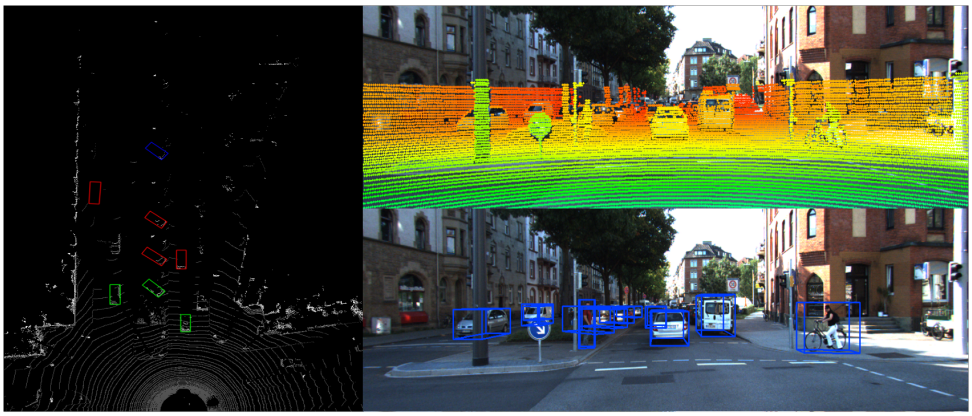

Figure 4. An example from the KITTI dataset.

4.2. Implementation Details

The objects in the dataset were categorized into easy, moderate, and hard levels. The hard category includes smaller-sized objects, which present an increasing level of difficulty for detection. This paper primarily focuses on analyzing the results for the car and pedestrian categories. We only considered the LiDAR point cloud within the camera's field of view for detection.

In terms of time alignment, the LiDAR serves as the timestamp reference, minimizing deviations caused by dynamic objects by triggering the camera shutter accordingly. For spatial alignment, the camera and point data are transformed into a unified coordinate system through extrinsic calibration.

Setting the IoU threshold for each category enables an objective evaluation of model performance. In vehicle detection, where bounding boxes typically exhibit regularity and relatively fixed sizes, a higher IoU threshold is required to assess model performance accurately. Conversely, for pedestrians and cyclists who display diverse poses and smaller sizes, a lower IoU threshold is necessary to effectively evaluate the model's performance in these categories. Specifically, this paper focused on calculating the AP40 [28] for persons and cyclists at an IoU threshold of 0.5, as well as the AP40 for cars at an IoU threshold of 0.7.

The proposed method in this paper was implemented based on Python and the MMDetection3D [29] framework, using the AdamW [30] optimizer for training with an initial learning rate of 0.001 and an exponential decay factor of 0.01. The momentum decay parameter during training varied between 0.95 and 0.99, and training was conducted for 40 epochs on a Tesla V100 GPU.

4.3. Quantitative Evaluation

The comparative results for 3D detection metrics are shown in Table 1. In the Methods column, 'L' denotes LiDAR-based methods, while 'L + C' indicates fusion methods. *N/A* indicates that this data is not disclosed in the relevant work. The performance of the proposed model outperforms the SECOND [13] and PointPillars [14] methods based on LiDAR. Compared to PointPillars [14], the proposed method achieves superior 3D AP across different levels for both person and car categories, with an improvement of 6.54% in the hard level for a person. Due to unrestricted 2D detectors, the proposed model demonstrates superior performance compared to F-PointNet [19]. MVXNet [31] also utilizes point level fusion, and in comparison, the proposed method achieves a 3.95% improvement in 3D AP for the hard level of person categories and a 2.74% improvement for the hard level of car categories.

Table 1. Results on the KITTI test 3D detection benchmark.

	Methods	**Car**			**Person**		
		Easy	**Mod**	**Hard**	**Easy**	**Moderate**	**Hard**
SECOND [13]	L	83.13	73.66	66.20	51.07	42.56	37.29
PointPillars [14]	L	79.05	74.99	68.30	52.08	43.53	41.49
F-PointNet [19]	L + C	81.20	70.39	62.19	51.21	44.89	40.23
MV3D [21]	L + C	71.09	62.35	55.12	*N/A*	*N/A*	*N/A*
AVOD [22]	L + C	81.94	71.88	66.38	50.80	42.81	40.88
MVXNet [31]	L + C	87.77	75.94	71.07	53.15	48.05	44.08
Ours	L + C	88.67	76.74	73.81	56.87	51.74	48.03

This paper further evaluates the model's performance in the BEV detection metrics, as shown in Table 2. Compared to Point RCNN [17] and F-ConvNet [20], the proposed method shows significant improvement, primarily in the person category. Compared to Pointpainting [23], the proposed model achieves a 3.24%, 6.61%, and 6.56% improvement in BEV AP for person category across three levels.

Table 2. Results on the KITTI test BEV detection benchmark.

	Methods	Car			Person		
		Easy	Mod	Hard	Easy	Moderate	Hard
SECOND [13]	L	88.07	79.31	77.95	55.10	46.27	44.76
PointPillars [14]	L	90.07	86.56	82.81	57.6	48.64	45.78
F-PointNet [19]	L + C	91.17	84.67	74.77	57.13	49.51	45.48
F-ConvNet [20]	L + C	91.51	85.84	76.11	57.04	48.96	44.33
Point RCNN [17]	L	92.13	87.39	82.72	54.77	46.13	42.84
LaserNet [18]	L	79.19	74.52	68.45	*N/A*	*N/A*	*N/A*
AVOD [22]	L + C	90.99	84.82	79.62	58.49	50.32	46.98
PointPainting [23]	L + C	92.45	88.11	83.36	58.70	49.93	46.29
Ours	L + C	95.07	88.21	83.90	61.94	56.54	52.85

4.4. Qualitative Results

This paper provides results from the dataset in Figures 5 and 6. In the point cloud on the left, the blue boxes represent ground truth annotations, while the green boxes represent the detection results. The red boxes of the camera images in top-right display the ground truth, while the blue boxes of the camera images in bottom-right shows the detection results. In the vehicle detection scenario, the proposed model accurately obtains the 3D bounding boxes of vehicles parked on the roadside and achieves precise predictions for small objects at long distances. Additionally, even without labeled data in the dataset, the model successfully predicts three heavily occluded vehicle objects. In the pedestrian detection scenario, the model exhibits good detection performance for small objects, especially at long distances. However, there are also some false positives, such as detecting two bikes parked side by side as one bike.

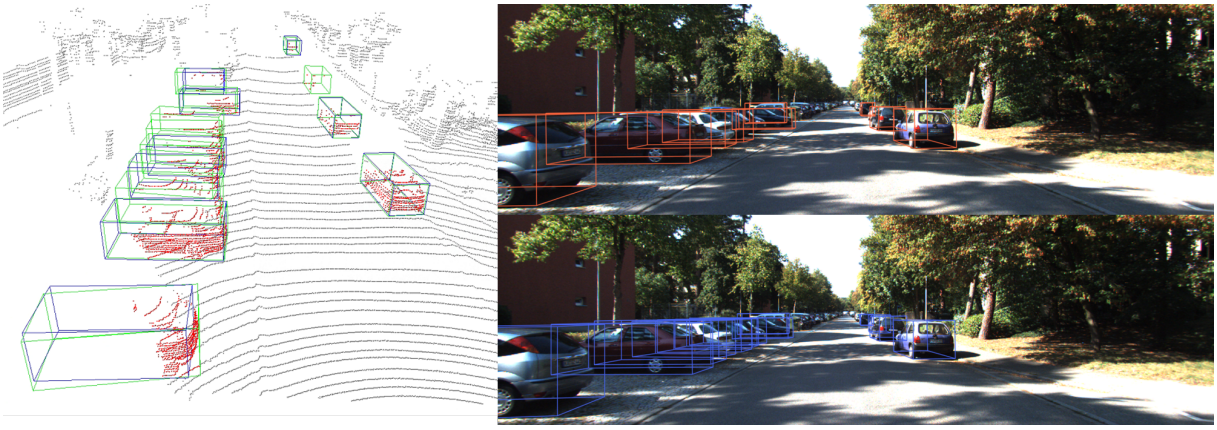

Figure 5. Detection results of vehicle scenes.

In order to evaluate the robustness of the model in the presence of external environmental interference, we conducted experiments to simulate various conditions. The blue boxes represents the detection results in the corresponding scenario. Specifically, we compared the detection performance under image blur, light rain, and heavy rain with fog weather. The actual detection results are presented in Figure 7, where the left side displays images (from top to bottom: original image, blurred scene, light rain scene, and heavy rain with fog scene) and the right side shows the corresponding detection results obtained by our model. It should be noted that during these experiments, no processing was applied to the LiDAR point cloud, and our model was not adaptively trained on disturbed data.

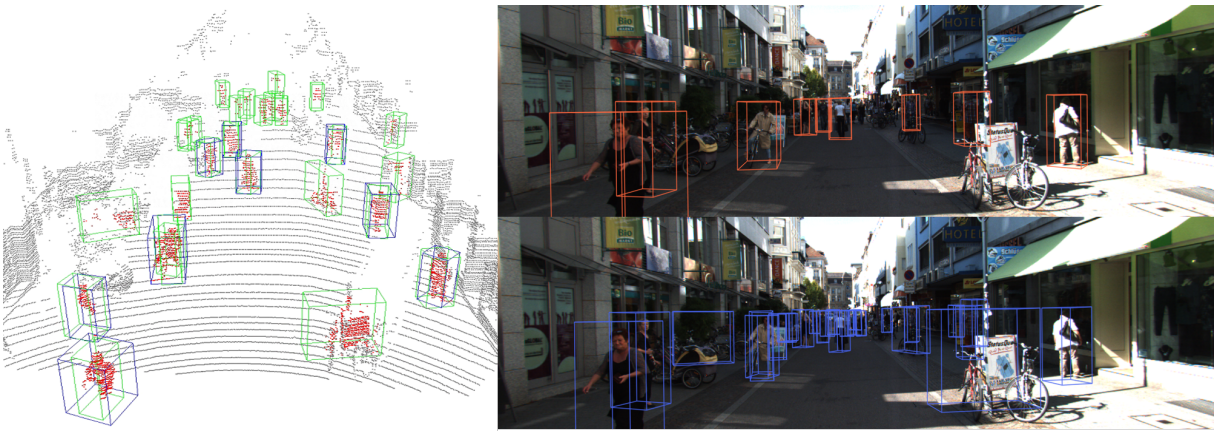

Figure 6. Detection results of pedestrian scenes.

Figure 7. Comparison of model performance in different scenarios.

The model consistently generates accurate detection results in all four scenarios at normal distances, with only some difficulties observed in long-distance object boxes obtained during heavy rain with fog scenes. However, this challenge remains within an acceptable range. Comparative analysis reveals the commendable robustness of the proposed model under diverse environmental conditions.

4.5. Ablation Study

To assess the impact of each module on performance, ablation experiments were conducted on the KITTI validation dataset, using 3D mAP (mean average precision) to evaluate different levels of car, cyclist, and pedestrian detection. The results are presented in Table 3. ✓ means that the corresponding module or method was used during the experimental process.

Table 3. Ablation experiments on the KITTI validation dataset.

CIF	MSV	MLF	3D Detection (mAP)		
			Easy	Moderate	Hard
✓	✓		67.46	54.45	51.18
✓		✓	67.56	54.79	51.34
	✓	✓	68.16	57.21	54.24
✓	✓	✓	70.83	58.87	55.75

First, the effectiveness of the cascaded image fusion module (CIF) was validated by comparing it to MobileNetV3-s [25] without depth image inputs. The cascaded image fusion module showed a 1.21% improvement in the hard level compared to the backbone based on the RGB image. Since no learnable layers were introduced, the model parameter count did not significantly increase.

Next, compared to using single-scale voxel (0.08, 0.08, 4), the multi-scale voxel module (MSV) significantly improved the performance, particularly in moderate and hard levels, with increases of 4.08% and 4.41%, respectively. This improvement resulted from enhanced voxel features at different scales, enhancing the model's ability to small objects.

Lastly, the effectiveness of multi-level fusion (MLF) between corresponding features at different levels was verified and compared with using only a single-level feature correspondence. In single-level feature correspondence, the fusion point features from the final level were only fused with the initial voxel features at different scales. The results demonstrate that utilizing the multi-level fusion between corresponding features at different levels leads to performance improvements.

5. Discussion

We proposed a novel 3D object detection model that integrates LiDAR and camera data for accurate small object detection tasks. The key contribution lies in enhancing the feature extraction process through a cascaded image fusion module and a multi-scale voxel module. The proposed model was investigated and compared with current advanced methods on the KITTI dataset, and the effectiveness of the proposed modules was verified through ablation experiments.

By conducting experimental analysis on the KITTI dataset, we compared our method with current 3D object detection algorithms and verified the effectiveness of the proposed module through ablation experiments. The results of our experiments and data visualization demonstrate that our 3D object detection method achieves superior accuracy across various difficulty levels. Specifically, for vehicles at the hard level, the 3D detection accuracy reached 73.81%, while for pedestrians it was 48.03%. Moreover, our model exhibited excellent performance in detecting occluded objects and small objects at long distances.

The detection method employed in this article belongs to the multi-sensor fusion 3D object detection approach, which enables the acquisition of more comprehensive feature information, thereby resulting in superior detection performance compared to single-sensor methods. The cascaded image fusion module enhances the robustness of image features and ensures reliable detection results even under image disturbances. Furthermore, the multi-scale voxel module equips the model with point cloud features at varying resolutions, enabling exceptional detection performance for distant or small objects. Therefore, it can effectively achieve 3D detection of small objects.

Due to hardware limitations, our model has not yet been deployed on edge computing devices. Further verification of the detection accuracy and performance of our method is required in real-world scenarios.

6. Conclusions

Three-dimensional object detection serves as an upstream subsystem in autonomous driving systems and plays a pivotal role in the development of smart cities. Precise 3D de-

tection outcomes enable vehicles to monitor dynamic objects surrounding them in real-time, identify potential collision hazards, and enhance vehicle safety performance. Therefore, the accuracy and scene robustness of 3D object detection algorithms are crucial factors.

Autonomous vehicles in smart cities can realize intelligent transportation collaboration through data sharing with urban traffic management systems. Based on the data of 3D object detection, vehicles can better interact with traffic signals, road condition monitoring, and other systems to jointly realize traffic flow optimization and congestion alleviation.

This paper proposes a model that integrates point cloud and RGB image features for the 3D detection of small objects. The experimental results demonstrate the efficacy of the proposed method in the 3D detection of small objects. This method offers a viable and effective framework for accomplishing the 3D detection of small objects, thereby providing valuable insights for future advancements in 3D object detection algorithms.

We will further focus on enhancing the interaction between the features at different scales and improving the ability of the detection model in real-time inference. Additionally, we aim to employ feature extraction operators with enhanced cross-modal feature representation capabilities to improve the efficiency of the model for raw information.

Author Contributions: Methodology, Y.Z.; project administration, D.W.; software, Y.Z.; supervision, S.L. and X.H.; validation, Y.Z.; writing—original draft, Y.Z.; writing—review and editing, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62101314.

Data Availability Statement: The data provided in this study can be provided upon request from the corresponding author. These data can be downloaded from the KITTI official website and have been made public. The application page is https://www.cvlibs.net/datasets/kitti/, accessed on 3 March 2024. The code repository for the MMdetection3D framework can be found at https://openmmlab.com, accessed on 3 March 2024.

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

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average Precision
mAP	Mean Average Precision
ROI	Region of Interest
VFE	Voxel Feature Encoder
SEFPN	Spatial Channel Feature Pyramid Network
FC	Fully Connected
IoU	Intersection over Union
GPU	Graphics Processing Unit
BEV	Bird's-Eye-View
CIF	Cascaded Image Fusion Module
MSV	Multi-Scale Voxel Module
MLF	Multi-Level Fusion

References

1. Cirianni, F.; Monterosso, C.; Panuccio, P.; Rindone, C. A review methodology of sustainable urban mobility plans: Objectives and actions to promote cycling and pedestrian mobility. In *Smart and Sustainable Planning for Cities and Regions: Results of SSPCR 2017, Proceedings of the 2nd International Conference on Smart and Sustainable Planning for Cities and Regions—SSPCR 2017, Bolzano, Italy, 22–24 March 2017*; Springer: Cham, Switzerland, 2018; pp. 685–697.
2. Russo, F.; Rindone, C. Smart city for sustainable development: Applied processes from SUMP to MaaS at European level. *Appl. Sci.* **2023**, *13*, 1773. [CrossRef]
3. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.

4. Wang, T.; Zhu, X.; Pang, J.; Lin, D. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 913–922.
5. Brazil, G.; Liu, X. M3d-rpn: Monocular 3d region proposal network for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9287–9296.
6. Ma, X.; Wang, Z.; Li, H.; Zhang, P.; Ouyang, W.; Fan, X. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6851–6860.
7. Park, D.; Ambrus, R.; Guizilini, V.; Li, J.; Gaidon, A. Is pseudo-lidar needed for monocular 3d object detection? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 3142–3152.
8. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
9. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
10. Xue, Y.; Mao, J.; Niu, M.; Xu, H.; Mi, M.B.; Zhang, W.; Wang, X.; Wang, X. Point2seq: Detecting 3d objects as sequences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8521–8530.
11. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11040–11048.
12. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4490–4499.
13. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
14. 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/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
15. Mao, J.; Niu, M.; Bai, H.; Liang, X.; Xu, H.; Xu, C. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 2723–2732.
16. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-based 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.
17. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
18. Meyer, G.P.; Laddha, A.; Kee, E.; Vallespi-Gonzalez, C.; Wellington, C.K. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12677–12686.
19. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 918–927.
20. Wang, Z.; Jia, K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1742–1749.
21. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
22. 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 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–8.
23. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. Pointpainting: Sequential fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 14–19 June 2020; pp. 4604–4612.
24. Xu, S.; Zhou, D.; Fang, J.; Yin, J.; Bin, Z.; Zhang, L. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3047–3054.
25. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
26. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2980–2988.
27. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 3354–3361.
28. Simonelli, A.; Bulo, S.R.; Porzi, L.; Antequera, M.L.; Kontschieder, P. Disentangling monocular 3d object detection: From single to multi-class recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1219–1231. [CrossRef] [PubMed]

29. Contributors, M. MMDetection3D: OpenMMLab Next-Generation Platform for General 3D Object Detection. 2020. Available online: https://github.com/open-mmlab/mmdetection3d (accessed on 3 March 2024).
30. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
31. Sindagi, V.A.; Zhou, Y.; Tuzel, O. Mvx-net: Multimodal voxelnet for 3d object detection. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 7276–7282.