Automotive Innovation (2019) 2:276–283
https://doi.org/10.1007/s42154-019-00083-z

3D Vehicle Detection Based on LiDAR and Camera Fusion

Yingfeng Cai[1] · Tiantian Zhang[2] · Hai Wang[1] · Yicheng Li[1] · Qingchao Liu[1] · Xiaobo Chen[1]

Received: 24 January 2019 / Accepted: 13 November 2019 / Published online: 30 November 2019

Abstract
Nowadays, the deep learning for object detection has become more popular and is widely adopted in many fields. This paper focuses on the research of LiDAR and camera sensor fusion technology for vehicle detection to ensure extremely high detection accuracy. The proposed network architecture takes full advantage of the deep information of both the LiDAR point cloud and RGB image in object detection. First, the LiDAR point cloud and RGB image are fed into the system. Then a high-resolution feature map is used to generate a reliable 3D object proposal for both the LiDAR point cloud and RGB image. Finally, 3D box regression is performed to predict the extent and orientation of vehicles in 3D space. Experiments on the challenging KITTI benchmark show that the proposed approach obtains ideal detection results and the detection time of each frame is about 0.12 s. This approach could establish a basis for further research in autonomous vehicles.

Keywords Vehicle detection · LiDAR point cloud · RGB image · Fusion

Abbreviations

BEV	Bird's-eye view of the LiDAR point cloud
IOU	Intersection over union
ROI	Region of interest
AOS	Average orientation similarity

1 Introduction

An intelligent driving vehicle refers to a complex system that combines perception, decision-making, and control technologies. Environmental perception provides fundamental information for path planning, decision-making and control. Vehicle detection is an extremely important task in environmental perception systems of autonomous vehicle. At present, LiDAR and cameras are the mainstream of obstacle detection sensors. Cameras are widely used in intelligent driving, especially in traffic sign identification and lane recognition thanks to their low cost and capability to obtain the texture and color of objects.

During the past few years, 2D object detection from camera images has seen significant progress [1–3]. However, there is still large improvement potential when it comes to object localization in 3D space. As a camera is sensitive to light and shadows, it cannot provide accurate and sufficient positional information, which often results in low real-time performance and poor robustness. In contrast, LiDAR can obtain the distance and 3D information of a detection object, and it has been widely used in environmental perception. In general, there are two methods for processing the LiDAR point cloud spatially before the 3D information is used. The first method is to establish a 3D grid map on the LiDAR point cloud [4, 5] and then process the LiDAR point cloud on the grid map. Although the 3D grid representation preserves most of the raw information of the point cloud, it usually requires much more complex computation for subsequent processing. The second method is to project the LiDAR point cloud into 2D space [6], which can reduce the amount of calculations. Although the LiDAR-based algorithm is widely used in target detection, the resolution of the LiDAR point cloud decreases as the detection distance increases.

As a single sensor cannot meet the needs of autonomous driving, the application of multi-sensor fusion schemes that include both cameras and LiDAR in intelligent vehicles has been gradually increasing. Multi-sensor data fusion takes full use of the data collected by multiple sensors. At present, according to the level of information processing, fusion

✉ Hai Wang
wanghai1019@163.com

[1] Institute of Automative Engineering, Jiangsu University, Zhenjiang 212013, China

[2] School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China

systems are divided into three levels: (1) Pixel-level fusion [7, 8] integrates the collected data directly and then extracts feature vectors from the fused data to identify the detected objects. The data for fusion that have not been processed lead to an enormous amount of calculations. (2) Feature-level fusion [9, 10] extracts representative features from the data collected by each sensor and fuses the features into a single feature vector for processing. Because of the abandonment of a portion of the data, the accuracy is reduced. (3) Decision-level fusion [11] is based on the independent detection and classification of each sensor. It makes an optimal global decision by integrating the recognition results of multiple sensors.

In this paper, the proposed network architecture fuses the LiDAR point cloud and RGB image to achieve high performance in autonomous vehicles. Firstly, the LiDAR point cloud is projected to the BEV (bird's-eye view of the LiDAR point cloud). Then the processed LiDAR point cloud and RGB image are used as network input. Inspired by the idea of FPN (feature pyramid network) [12], a new feature extractor structure is proposed to generate a high-resolution feature map from the LiDAR point cloud and RGB image that has high detection performance for objects. Secondly, the high-resolution feature map is fused to generate reliable 3D vehicle proposals. Further, ROI (region of interest) pooling [3] for each feature map is employed to obtain equal-length feature vectors and the ROI pooling feature map is fused using an element-wise mean operation [13]. Thirdly, 3D box regression is performed to predict the extent and orientation of vehicles in 3D space. The architectural diagram of the proposed fusion method is shown in Fig. 1.

The contributions of this research can be summarized as follows: (1) A new vehicle detection method is proposed based on the fusion of a feature map that is generated from LiDAR point cloud and RGB images. (2) A new feature extractor is proposed that can generate a high-resolution feature map which is suitable for subsequent processing. (3) A new 3D bounding box is proposed to predict the extent and orientation of a vehicle.

2 3D Vehicle Detection Method Architecture

2.1 3D Point Cloud Representation

That front vehicles occlude rear vehicles in the front view of the LiDAR point cloud affects the detection result. Therefore, to retain the information from the LiDAR point cloud data more effectively, a more compact representation of the LiDAR point cloud is proposed by projecting the 3D point cloud onto a BEV map. The BEV map is encoded according to height and density, and is represented by a 2D grid with a resolution of 0.1 m. As a 3D grid representation requires complex and extensive computation for feature extraction and the aim is to obtain more detailed information of the detection vehicle [14], the LiDAR position is set as the center, with the maximal left and right positions set to [−40 m, 40 m] and the front position to [0, 70 m] on the BEV map to align with the image detection range. The scope of the cropped LiDAR point cloud in the BEV map is shown in Fig. 2.

According to the actual physical height of a vehicle, the point cloud along the Z-axis [0, 2.5 m] is divided into five equal height channels. Each channel is projected onto the 2D ground grid ($Z=0$) and is encoded with the maximum height of the points in each grid cell. The point cloud density

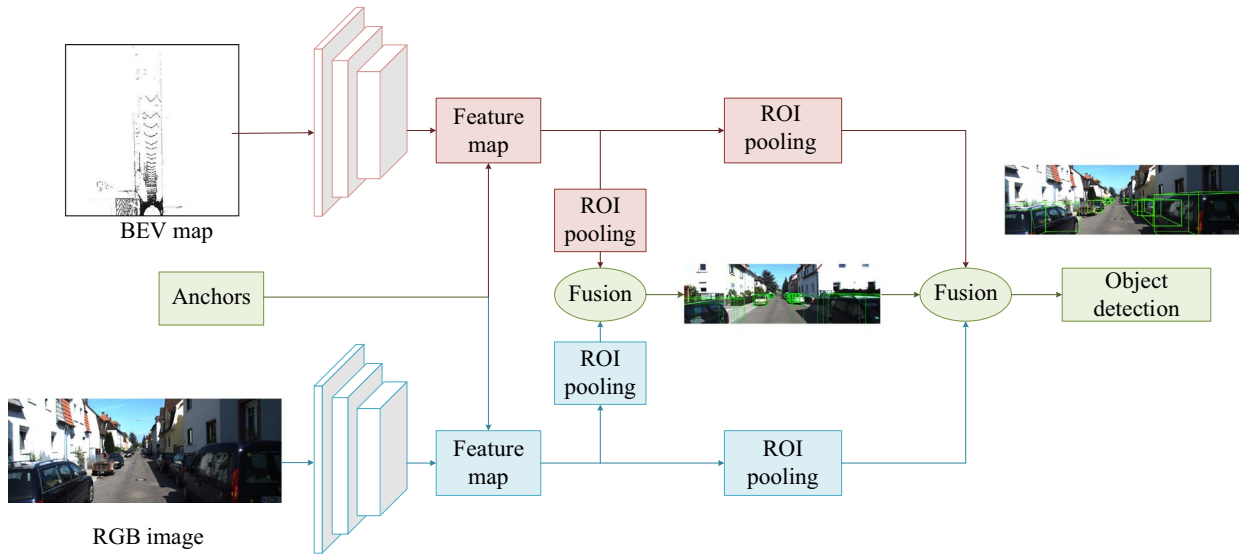

Fig. 1 Architecture of the proposed fusion method

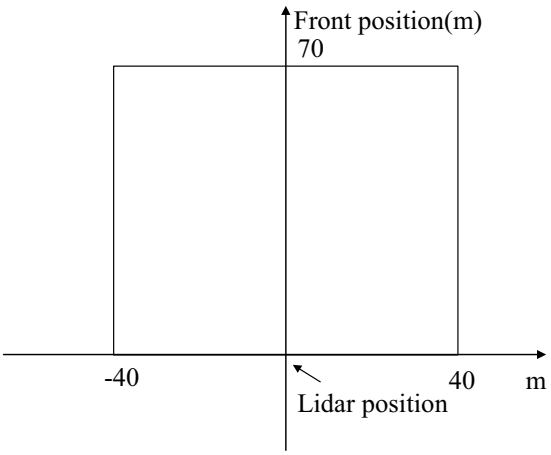

Fig. 2 Scope of the cropped LiDAR point cloud in the BEV map

is set to be the sixth channel, which refers to the number of points in each cell, and the value of each cell is normalized as follows:

$$\min\left(1.0, \frac{\ln(N+1)}{\ln 16}\right) \tag{1}$$

where N is the number of points in the cell.

2.2 Feature Map Generation and Feature Extraction

Inspired by the FPN which has become the key component of 2D object detection, features are extracted from the BEV map and the RGB image, respectively, to recognize and locate the vehicles. The bottom-level semantic information of an image is poor, whereas the physical information is accurate; moreover, the high-level semantic information is rich, whereas the physical information is not sufficiently abundant. Therefore, to make full use of the information of the original bottom-level feature map, an 1×1 convolution operation is performed to fuse the bottom-level feature with the high-level feature so that all the scales of the feature map have rich semantic information and physical information and the final feature map is suitable for subsequent processing.

The feature extractors are based on the VGG-16 architecture [15]. Assuming that the input sizes of both RGB image and BEV map are $H \times W \times D$, the first four convolution layers of the VGG-16 network for down-sampling are used, which results in a feature map output eight times smaller than the corresponding input. Hence, the output size of the feature map is $\frac{H}{8} \times \frac{W}{8} \times 256$. An 1×1 convolution layer is used to reduce the number of channel dimensions. The up-sampled map is then merged with the corresponding down-sampled map by element-wise addition [16]. Finally, a 3×3 convolution layer on each merged map is employed to generate the final feature map to reduce the aliasing effect. A feature map with high resolution and high semantic information is then obtained, and the size of this final feature map is $\frac{H}{2} \times \frac{W}{2} \times 256$. The feature extraction framework is shown in Fig. 3.

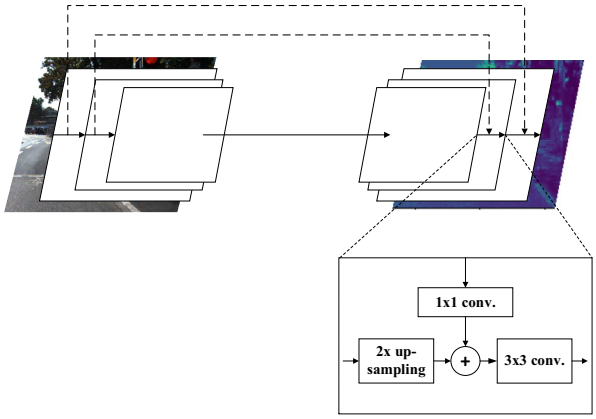

Fig. 3 Feature exaction framework

2.3 3D Proposal Network Design

Inspired by the idea of the RPN (region proposal network) which is an important component of the Faster R-CNN [3] network for 2D object detection, a 3D proposal network is designed to generate 3D proposals for the prediction of the vehicle orientation and extent. 3D box proposals from a set of 3D anchor boxes are generated to cover most of the vehicles in 3D space. Each 3D anchor box is parameterized by (x, y, z, l, w, h), where triplets (x, y, z) denote the center of the 3D anchor box and triplets (l, w, h) represent the size of the 3D predicted box. In addition, (l, w) of the anchor box takes values of (3.8 m, 1.6 m) and (1.0 m, 0.6 m), and the height h is fixed to 1.63 m. By rotating the 3D anchor by 90° and sampling the 3D anchor boxes at intervals of 0.5 m in the BEV map and RGB image, respectively, a total of 44,800 anchors are finally generated. Because the LiDAR point cloud is sparse, most of the 3D anchor boxes are empty. The empty anchors are removed to reduce the amount of calculations, and the final number of anchors is kept between 8000 and 15,000. Because features of the BEV and RGB image feature maps have different resolutions, the ROI pooling for each view is employed to obtain feature vectors of the same length. Given a generated 3D anchor box, the anchor is projected onto the BEV and RGB image feature maps, and the output of the ROI pooling feature maps is fused using an element-wise mean operation.

A binary label is assigned to each anchor that shows whether it is an object or background. By calculating the IOU (intersection over union) [17] between the anchor and the ground-truth bounding box, a positive label is assigned to two types of anchors: In the first type, the IOU determined by

the anchors and ground-truth bounding box is greater than 0.5, and in the second type, the IOU between the anchor generated at the same point and the ground-truth bounding box is the highest, even though it is less than 0.5. A ground-truth bounding box can assign positive labels to multiple anchors. A negative label assigned to an anchor with the IOU is less than 0.3 for all ground-truth bounding boxes. However, non-positive and nonnegative anchors have no effect on training objects, and are ignored in subsequent processing. Using the fused feature map to regress the anchor box, 3D box regression is used to generate the 3D proposals. A multitask loss is used to simultaneously classify vehicle/background and 3D box regression, Smooth L_1 loss is used for the 3D box regression, and class entropy loss is used for determining whether the anchor is positive or negative. The total loss L is as follows:

$$\begin{cases} L_{\text{cls}}\left(p_i, p_i^*\right) = -\left[\left(p_i \ln\left(p_i^*\right)\right) + \left(1 - p_i\right)\ln\left(1 - p_i^*\right)\right] \\ L_{\text{reg}}\left(\boldsymbol{t_i}, \boldsymbol{t_i^*}\right) = \begin{cases} 0.5\left(\boldsymbol{t_i} - \boldsymbol{t_i^*}\right)^2 & \left|\boldsymbol{t_i} - \boldsymbol{t_i^*}\right| \le 1 \\ \left|\boldsymbol{t_i} - \boldsymbol{t_i^*}\right| - 0.5 & \left|\boldsymbol{t_i} - \boldsymbol{t_i^*}\right| > 1 \end{cases} \\ L = \dfrac{1}{N}\sum_{i=1}^{N_{\text{cls}}} L_{\text{cls}}\left(p_{\text{i}}, p_i^*\right) + \dfrac{1}{N}\sum_{i=1}^{N_{\text{reg}}} p_i^* L_{\text{reg}}\left(\boldsymbol{t_i}, \boldsymbol{t_i^*}\right) \end{cases} \tag{2}$$

where L_{cls} is the class entropy loss, L_{reg} is the *Smooth* L_1 *loss*, p_i is the probability of an anchor predicted as an object, ground-truth label p_i^* is 1 if the anchor is positive and is 0 if the anchor is negative, N is the number of anchor, $\boldsymbol{t_i} = \left(t_x, t_y, t_z, t_l, t_w, t_h\right)$ is the offset of the predict box relative to the 3D anchor box, and $\boldsymbol{t_i^*} = \left(t_x^*, t_y^*, t_z^*, t_l^*, t_w^*, t_h^*\right)$ is the offset of the ground-truth box relative to the 3D anchor box. The calculation is expressed as follows:

$$\begin{cases} t_x = \dfrac{x_{\text{g}} - x_{\text{a}}}{d_{\text{a}}}; t_y = \dfrac{y_{\text{g}} - y_a}{d_{\text{a}}}; t_z = \dfrac{z_{\text{g}} - z_{\text{a}}}{h_a} \\ t_l = \lg\left(\dfrac{l_g}{l_{\text{a}}}\right); t_w = \lg\left(\dfrac{w_{\text{g}}}{w_{\text{a}}}\right); t_h = \lg\left(\dfrac{h_{\text{g}}}{h_{\text{a}}}\right) \\ t_x^* = \dfrac{x_{\text{p}} - x_{\text{a}}}{d_{\text{a}}}; t_y^* = \dfrac{y_{\text{p}} - y_{\text{a}}}{d_{\text{a}}}; t_z^* = \dfrac{z_{\text{p}} - z_{\text{a}}}{h_{\text{a}}} \\ t_l^* = \lg\left(\dfrac{l_{\text{p}}}{l_{\text{a}}}\right); t_w^* = \lg\left(\dfrac{w_{\text{p}}}{w_{\text{a}}}\right); t_h^* = \lg\left(\dfrac{h_{\text{p}}}{h_{\text{a}}}\right) \\ d_{\text{a}} = \sqrt{\left(l_{\text{a}}\right)^2 + \left(w_{\text{a}}\right)^2} \end{cases} \tag{3}$$

where $(x_{\text{g}}, y_{\text{g}}, z_{\text{g}}, l_{\text{g}}, w_{\text{g}}, h_{\text{g}})$ is the ground-truth box, $(x_{\text{a}}, y_{\text{a}}, z_{\text{a}}, l_{\text{a}}, w_{\text{a}}, h_{\text{a}})$ is the 3D anchor box, $(x_{\text{p}}, y_{\text{p}}, z_{\text{p}}, l_{\text{p}}, w_{\text{p}}, h_{\text{p}})$ is the predict box, and d_a is the diagonal length of the 3D anchor box. NMS (non-maximum suppression) at a threshold of 0.8 on the BEV map is applied to retain the top-1000 proposals during training, and the top-300 proposals are used only in testing. The 3D proposals are projected onto the RGB image, as shown in Fig. 4.

Fig. 4 Projection of the 3D proposals onto the RGB image

2.4 Region Proposal Fusion

Information fusion is a technology that is used to integrate and optimize a variety of information, and it retains useful information according to the inherent connections and rules of information. In this paper, the feature-level fusion method is employed.

A fusion network is designed to effectively combine features from the BEV map and RGB image map that jointly performs oriented 3D box regression. Because features from the BEV map and RGB image map have different resolutions, the ROI pooling on the feature map of each box proposal is performed to resize it to 7×7 to obtain equal-length feature vectors; and then the pooling feature map is fused using element-wise mean operation. The fused features are as follows:

$$F_L = H_L\left(H_{L-1}\left(\ldots H_1\left(F_{\text{BEV}} + F_{\text{RGB}}\right)\right)\right) \tag{4}$$

where F_L is the fused feature, H_L is the feature transformation function of layer L, F_{BEV} is the feature of the BEV map, and F_{RGB} is the feature of the RGB image map.

2.5 3D Box Regression

Given the fused features of the fusion network, a further regression operation is required to determine the orientation and classification of each proposal. Hence, the oriented 3D boxes are regressed from the 3D proposals. In particular, the bounding box is encoded using its length, width, center, and two heights, so the regression targets are encoded by $\left(\Delta x, \Delta y, \Delta d_x, \Delta d_y, \Delta h_1, \Delta h_2\right)$. The encoded bounding box is shown in Fig. 5. Compared to the 8-corner box encoding proposed in [14], only six vectors are needed to represent the oriented 3D box, so the proposed encoding procedure reduces the box representation from an overparameterized 24-dimensional vector to a six-dimensional one, which further reduces the redundancy while keeping the physical constraints. A multitask loss is used to simultaneously classify the predicted 3D box as a vehicle or background, and 3D box regression is performed using the Smooth L1 loss for the 3D box regression and class entropy loss for the classification task.

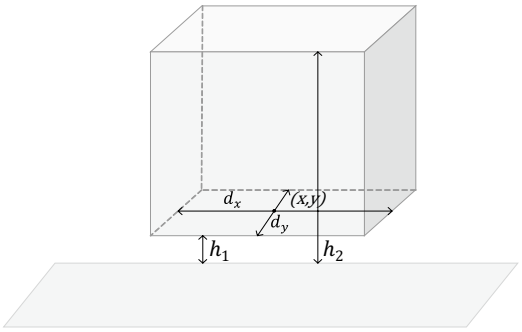

Fig. 5 Encoded 3D bounding box

2.6 KITTI Dataset

In this paper, the proposed network uses the KITTI dataset [18] for training and verification. The collected data scenes are diverse, from highway scenes to countryside scenes. It contains eight obstacle types: cars, vans, trucks, pedestrians, pedestrians (sitting), cyclists, trams, and others. All cars, vans, and trucks are treated as vehicles in this paper.

The dataset mainly consists of three parts: (1) RGB images collected by a camera; (2) LiDAR point clouds collected by a Velodyne HDL-64E laser scanner, which include information about the coordinates (x, y, z) in the LiDAR's coordinate system and reflection intensity of the LiDAR point cloud (about 1.3 million LiDAR cloud points per frame were collected); and (3) the calibration files, which describe the relationship between the camera coordinate system and LiDAR coordinate system. The dataset consists of 7481 training sets and 7518 verification sets.

2.7 Training

This work is based on the known coordinate relationship between the LiDAR point cloud and RGB image. The obtained 7481 training sets are split into two parts [4], resulting in 3712 data samples for training and 3769 data samples for validation.

The main parameters of the experimental platform are as follows: The processor is an Intel(R) core (TM) i5-8600 K CPU@3.60 GHz, the memory is 64 GB, and the graphics card is NVIDIA GeForce GTX1080. The network is trained by the ADAM optimizer through 100,000 global steps at an initial learning rate L_I of 0.001. The decay learning rate L_D is exponentially reduced at every 20,000 decay steps with a decay rate of 0.8, and the decay learning rate is expressed as follows:

$$L_D = L_I \times \text{decay\_rate}^{\left(\frac{\text{new\_step}}{20000}\right)} \tag{5}$$

where *new step* is the epoch of training until the epoch is equal to the global step.

3 Experimental Results

The KITTI dataset is used to evaluate the detection performance of the proposed method. The test results are evaluated based on three levels: easy, moderate, and difficult. An image is an easy image if the vehicles are fully visible and the maximum occlusion rate is 15%; it is a moderate image if the vehicles are partly occluded and the maximum occlusion rate is 30%; it is a difficult image if the vehicles are difficult to see and the maximum occlusion rate is 50%.

The proposed method is compared with several top-performing algorithms: a LiDAR-based approach, RT3D [19], an RGB image-based approach, Stereo R-CNN [20], and a fusion of LiDAR point cloud- and RGB image-based approach A3DODWTD [21]. The runtime and average precision of different methods are analyzed, in which average precision is defined as follows:

$$\begin{cases} P = \dfrac{TP}{TP + FP} \\ R = \dfrac{TP}{TP + FN} \\ AP = \dfrac{1}{11} \displaystyle\sum_{r \in \{0, 0.1, \ldots 1\}} \max[P(R > r)] \end{cases} \tag{6}$$

where P is the precision, R is the recall, AP is the average precision, TP is the number of 3D boxes that are correctly predicted to be vehicles, FP is the number of 3D boxes for which the background is predicted to be a vehicle, and FN is the number of 3D boxes for which a vehicle is predicted to be background. The results are compared in Table 1.

Table 1 Comparison results of several top-performing methods [19–21]

Methods	$AP_{3D}(\%)$			$AP_{BEV}(\%)$			Runtime (s)
	Easy	Moderate	Difficult	Easy	Moderate	Difficult	
RT3D	21.27	23.49	19.81	42.1	54.68	54.68	0.09
Stereo R-CNN	34.05	49.23	28.39	43.89	61.67	36.44	0.30
A3DODWTD	56.81	59.35	50.51	72.86	76.65	76.65	0.08
This paper	68.59	63.72	53.34	73.56	66.75	58.78	0.12

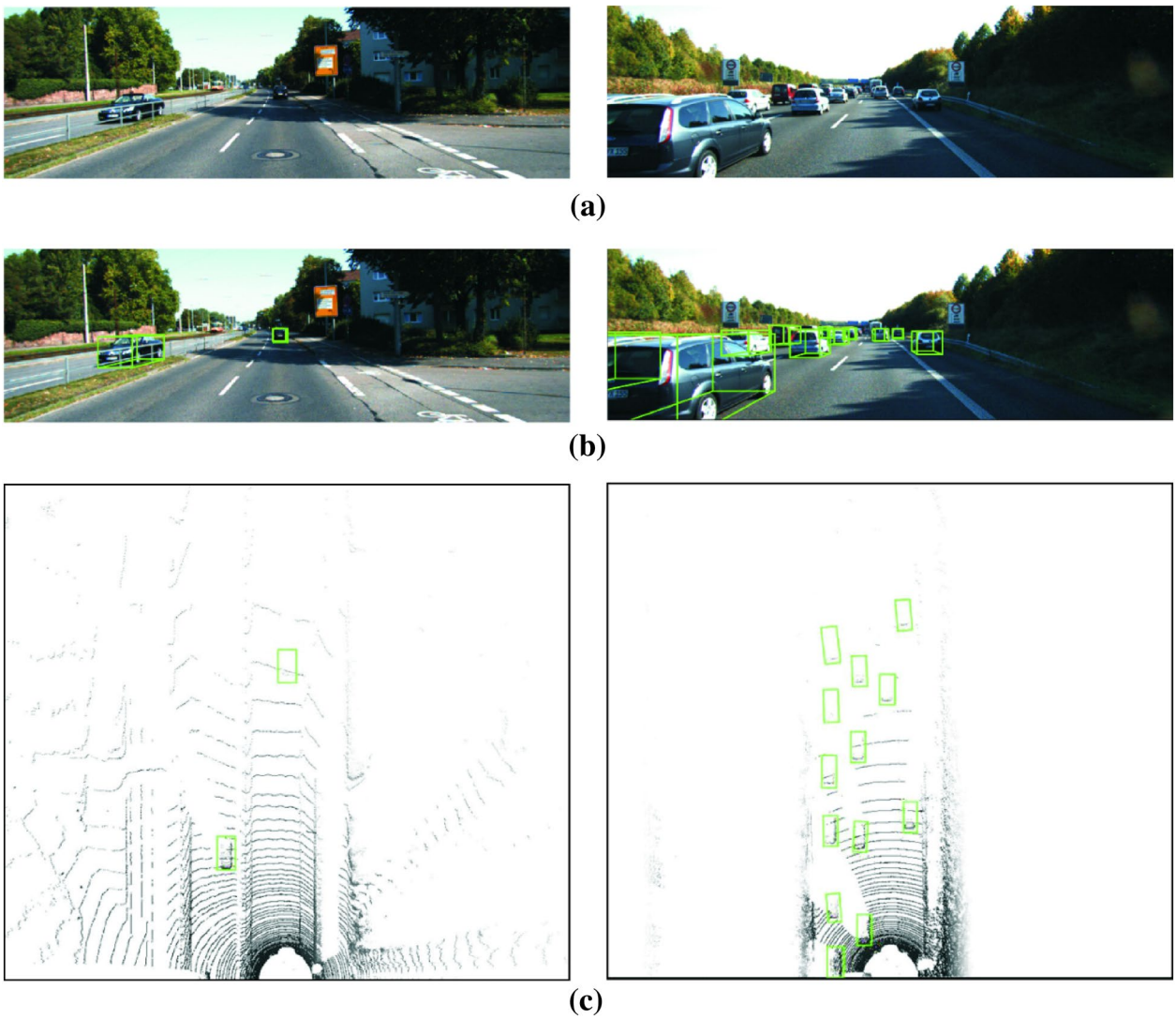

Fig. 6 3D vehicle detection in an easy scene (left) and difficult scene (right): **a** original RGB image; **b** 3D detection result on the RGB image; **c** detection result on BEV map

To verify the detection and real-time performances of the proposed vehicle detection method, two test scenarios with different levels of difficulty are selected. The detection results for a simple scene are presented in the left images of Fig. 6. In this scene, only a few cars are on the road, where the light is poor, and the tree shadows almost cover the right vehicle. The detection results for a difficult scene are shown in the right images of Fig. 6. In this scene, the road is full of vehicles on both sides that are occluding each other.

3.1 Evaluation in 3D Detection

Compared to 2D vehicle detection, 3D vehicle detection is more challenging. The comparison results show that the proposed method in this paper significantly outperforms other approaches with respect to the metric AP_{3D}. Specifically, the proposed method significantly outperforms the fusion method A3DODWTD by 11.78%, 4.37%, and 2.88% on easy, moderate, and difficult images, respectively.

3.2 Evaluation in BEV Detection

The evaluation result is presented in Table 1 for AP_{BEV}. The proposed method consistently outperforms the compared approaches, and it is obviously better than the image-based detection. The important reason for this performance is that vehicles are occluded in the image, so these vehicles will not be detected.

CenterNet [22] is based on image detection, and Fig. 7 shows a comparison of the proposed method and CenterNet with respect to 3D detection and BEV detection.

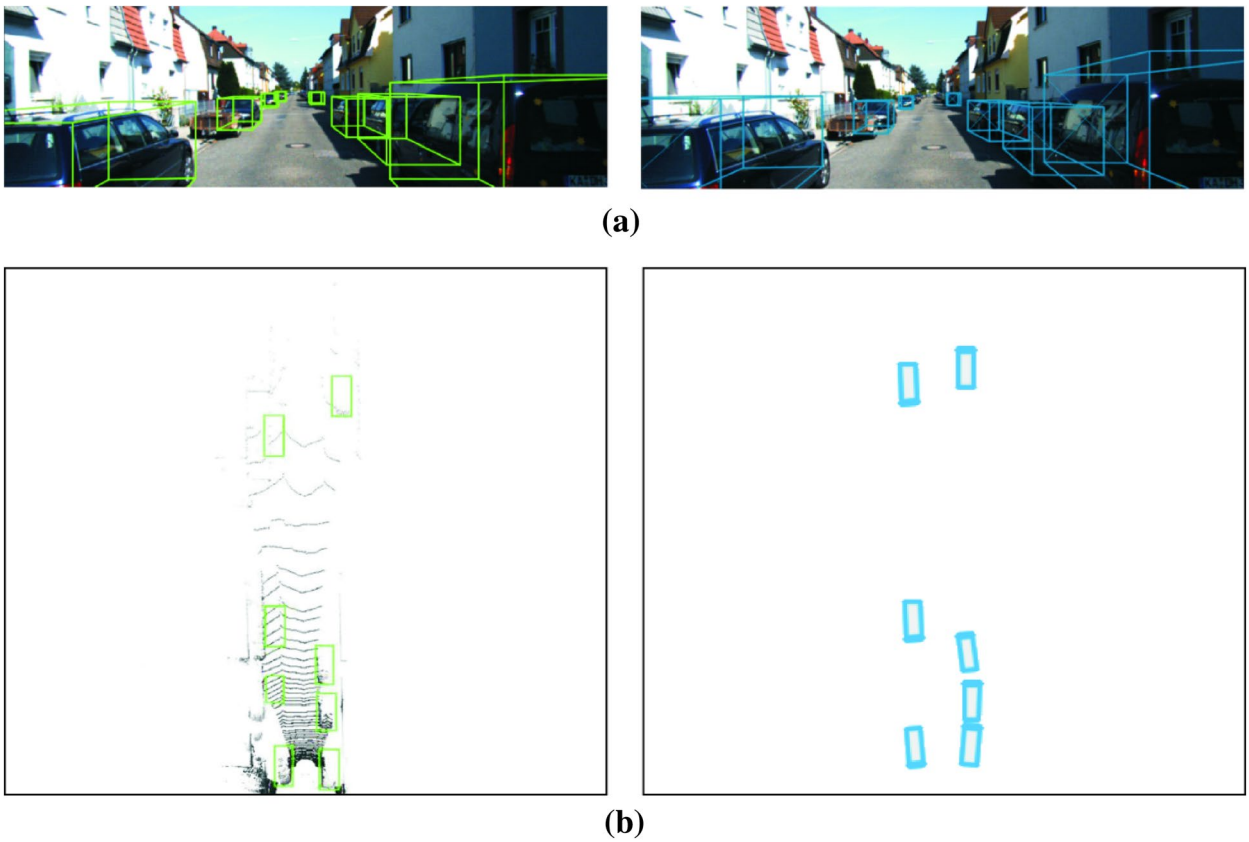

Fig. 7 Qualitative comparison of the proposed method (left) and CenterNet (right) 3D detection results: **a** RGB image detection; **b** BEV detection

3.3 AOS (Average Orientation Similarity) Evaluation

To evaluate the performance of orientation regression, the AOS is used according to the method proposed in [23]. AOS is defined as follows:

$$\begin{cases} s(r) = \dfrac{1}{|D(r)|} \displaystyle\sum_{i \in D(r)} \dfrac{1 + \cos \Delta_{\theta_i} \delta_i}{2}, \begin{cases} \delta_i = 1, & \text{IOU} \le 0.5 \\ \delta_i = 0, & \text{IOU} > 0.5 \end{cases} \\ \text{AOS} = \dfrac{1}{11} \displaystyle\sum_{r \in \{0, 0.1, \ldots 1\}} \max \; s(R > r) \end{cases} \tag{7}$$

where Δ_{θ_i} is the difference between the prediction angle and the ground-truth angle of vehicle i, and R is the recall.

Table 2 shows the AOS performance of different methods reported on the KITTI online evaluation. The AOS of the method proposed in this research outperforms that of the other methods, which well illustrates the advantage of the method.

Table 2 AOS of different methods (%)

Methods	Easy	Moderate	Difficult
[5]	34.0	25.4	22.0
[24]	59.1	45.9	41.1
This paper	65.2	58.4	54.3

The experimental results show that the proposed vehicle detection method achieves good detection results in scenes with different levels of difficulty. Moreover, as given in Table 1, the runtime of the proposed method is 0.12 s, so it could be used as a basis for further research in autonomous vehicles.

4 Conclusion

In this paper, a new method is proposed for 3D vehicle detection based on the fusion of data collected by LiDAR and a camera. The model takes advantage of both the LiDAR point cloud and RGB images. The RPN is improved to obtain the 3D proposals according to the BEV map and RGB image map. Furthermore, a fusion network is presented to fuse the information and perform 3D box regression. An experiment on the KITTI dataset is conducted to verify the detection performance. The experimental results show that the proposed method for 3D vehicle detection is superior to the

existing related methods based on LiDAR and camera data. The proposed method achieves good real-time and reliability performance in the experiment.

However, in the proposed method, the feature fusion of the BEV and RGB image is relatively simple, which could lead to insufficient use of feature information. Therefore, in the future work, the proposed network needs to be optimized to achieve better 3D vehicle detection results.

Acknowledgements This work was supported by the National Key Research and Development Program of China (2017YFB0102603, 2018YFB0105003), the National Natural Science Foundation of China (51875255, 61601203, 61773184, U1564201, U1664258, U1764257, U1762264), the Natural Science Foundation of Jiangsu Province (BK20180100), the Six Talent Peaks Project of Jiangsu Province (2018-TD-GDZB-022), the Key Project for the Development of Strategic Emerging Industries of Jiangsu Province (2016-1094), and the Key Research and Development Program of Zhenjiang City (GY2017006).

Compliance with Ethical Standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Kehl, W., Manhardt, F., Tombari, F., et al.: SSD-6D: making RGB-based 3D detection and 6D pose estimation great again. In: IEEE International Conference on Computer Vision, Computer Vision Foundation, Venice, 22–29 October, 2017
2. Liu, W., Anguelov, D., Erhan, D., et al.: SSD: single shot multi-Box detector. computer science. In: 16th Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Boston, 8–10 June, 2015
3. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
4. Zhou, Y., Tuzel, O.: VoxelNet: End-to-End learning for point cloud based 3D object detection. In:19th Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Salt Lake City, 18–22 June, 2018
5. Behley, J., Steinhage, V., Cremers, A.B.: Laser-based segment classification using a mixture of bag-of-words. In: IEEE International Conference on Computer Vision, Computer Vision Foundation, Tokyo, 7–10 April, 2013
6. Wu, B., Wan, A., Yue, X., et al.: SqueezeSeg: convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In: IEEE International Conference on Robotics and Automation, Faculty of Mathematics and Physics of Charles University, Brisbane, 13–17 August, 2018
7. Li, S., Kang, X., Fang, L., et al.: Pixel-level image fusion: a survey of the state of the art. Inform. Fusion **33**(5), 100–112 (2017)
8. Cvejic, N., Nikolov, S. G., Knowles, H. D., et al.: The effect of pixel-level fusion on object tracking in multi-sensor surveillance video. In: 8th IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Minneapolis, 18–23 June, 2007
9. An, L., Chen, X., Yang, S.: Multi-graph feature level fusion for person re-identification. Neurocomputing **259**(4), 39–45 (2017)
10. Sharma, V., Davis, J.W.: Feature-level fusion for object segmentation using mutual information. In: 7th IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, New York, 17–22 June, 2006
11. Yebo, G., Minglei, Y., Zhenguo, S., et al.: The applications of decision-level data fusion techniques in the field of multiuser detection for DS-UWB systems. Sensors **15**(10), 24771–24790 (2015)
12. Lin, T.Y., Dollár, P., Girshick, R., et al.: Feature pyramid networks for object detection. CVPR **11**(2), 936–944 (2016)
13. Cai, Z., Fan, Q., Feris, R.S., et al.: A unified multi-scale deep convolutional neural network for fast object detection. Comput. Vis. **9908**, 354–370 (2016)
14. Chen, X., Ma, H., Wan, J., et al.: Multi-view 3D object detection network for autonomous driving. In: 17th Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Las Vegas, 26–30 June, 2016
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Comput. Sci. **428**(6), 158–165 (2014)
16. Song, Y., Gong, L.: Analysis and improvement of joint bilateral upsampling for depth image super-resolution. Wireless Communications and Signal Processing, Institute of Electrical and Electronics Engineers, Yangzhou, 13–15 October, 2016
17. Girshick, R., Donahue, J., Darrelland, T., et al.: Rich feature hierarchies for object detection and semantic segmentation. In: 15th Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Tianjin, 3–6 August, 2014
18. Geiger, A., Lenz, P., Stiller, C., et al.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
19. Zeng, Y., Hu, Y., Liu, S., et al.: RT3D: real-time 3D vehicle detection in LiDAR point cloud for autonomous driving. IEEE Robot. Autom. Lett. **8**(11), 125–132 (2018)
20. Li, P., Chen, X., Shen, S.: Stereo R-CNN based 3D object detection for autonomous driving. In:20th Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Long Beach, 16–20 June, 2019
21. Gustafsson, F., Linder-Noren, E.: Automotive 3D object detection without target domain annotations. Dissertation, Linkoping University (2018)
22. Duan, K., Bai, S., Xie, L., et al.: CenterNet: keypoint triplets for object detection. In: 20th Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Long Beach, 16–20 June, 2019
23. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 13th Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Providence, 10–15 June, 2012
24. Lederer, C., Altstadt, S., Andriamonje, S., et al.: Vehicle detection from 3D Lidar using fully convolutional network. Robotics: Science and Systems, University of Michigan, Ann Arbor, 20–22 June, 2016