

3D LiDAR and image data-level fusion for traffic vehicle detection

Xinpeng Yao^{1,†}, Ruini Zhang^{2,†}, Yunchao Li², Wen Rong¹, Zijian Wang^{1,✉}, Han Zhang¹

 Cite this article: Yao X P, Zhang R N, Li Y C, et al. *J Intel Connect Veh* 2026, 9(1): 9210074. <https://doi.org/10.26599/JICV.2025.9210074>

ABSTRACT: Vehicle detection is a crucial part of perception technologies for unmanned vehicles, which provides perceptive guarantees for route plans and vehicle control. Current multimodal fusion approaches for image and light detection and ranging (LiDAR) data primarily include feature-level fusion, object-level fusion, and data-level fusion strategies. However, feature-level and object-level fusion methods fail to fundamentally address the core challenges of low-quality sparse point cloud data with insufficient information and high noise levels, resulting in low confidence in fused features or detection results. Research on data-level fusion methods remains limited. Aiming at the ineffective representation of distant vehicles by sparse LiDAR clouds, this study proposes a vehicle detection method based on the data-level fusion of image and LiDAR data. First, this method estimates the depth of image pixels to generate the pseudo point clouds. Then, the K-dimensional (KD) tree is employed to reduce the noise and eliminate the outlier of pseudo points to fuse with LiDAR data. Finally, the fused point clouds are fed to the point-based region-based convolutional neural network (PointRCNN) to achieve vehicle detection. The advantage of the proposed method is that it utilizes pseudo point clouds to compensate for sparse LiDAR data, which contributes to improving the detection accuracy of distant vehicles. This study utilizes the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset to conduct comparative experiments and ablation studies. The experimental results reveal that the proposed method improves 7.8% and 4.13% in the aspects of average precision for three-dimensional objects (AP3D) and average precision for bird's eye view (APBEV) with comparison methods, respectively. Meanwhile, in terms of distant vehicle detection, the proposed method raises 9.13% and 9.58% in the aspects of AP3D and APBEV respectively by comparison with frustum-based PointNet (F-PointNet).

KEYWORDS: vehicle detection; three-dimensional light detection and ranging (3D LiDAR); image fusion; point-based region-based convolutional neural network (PointRCNN); K-dimensional (KD) tree

1 Introduction

Three-dimensional (3D) object detection aims to obtain semantic information (such as category) and geometric information (including position, size, and orientation) about objects in traffic scenarios. This task is a critical component of environmental perception technology in autonomous vehicles, providing strong support for accurate obstacle avoidance, safe driving, and path planning in self-driving cars (Chen et al., 2020; He et al., 2025; Zhao et al., 2025).

To improve the robustness of traffic target detection, such as performance under occlusion and poor illumination, researchers have used sensor solutions such as deep vision, light detection and ranging (LiDAR), and millimeter-wave radar in target detection (Cao et al., 2022; Hao et al., 2024; Lesani et al., 2020; Li et al., 2021; Lin et al., 2022; Yang et al., 2025). LiDAR is robust and insensitive to a wide range of light or adverse weather conditions; thus, it is extensively used for 3D object detection tasks in autonomous driving. Qi et al. (2024) proposed a deep learning network (GIC-

Net) with geometric information-based constraints, which was specifically designed to solve the sparsity and noise interference problem of LiDAR point clouds under severe weather and significantly improved the precision and robustness of 3D target detection. Due to LiDAR's fixed-angle scanning mode of operation, scene point clouds usually remain dense for near objects but sparse for far objects. The sparse point clouds at far distances fail to effectively represent the contours of objects, resulting in lower accuracy for distant object detection. The intrinsic cause of this issue is the limited amount of point cloud information and high noise levels associated with distant objects, which makes it challenging to extract high-recognition features for detection and identification. To address this problem, existing methods utilize multisensor data fusion for object detection, with the integration of color-textured images (Gao et al., 2025; Liu et al., 2023; Splietker and Behnke, 2021) and structural information from LiDAR point clouds (Maturana and Scherer, 2015) being particularly effective. Current methods for fusing LiDAR point clouds and image data primarily include object-level fusion

† Xinpeng Yao and Ruini Zhang contributed equally to this work.

¹ Shandong High-speed Group Innovation Research Institute, Jinan 250098, China. ² School of Information Engineering, Chang'an University, Xi'an 710064, China.

✉ Corresponding author. E-mail: 2022024021@chd.edu.cn

Received: May 22, 2025; Revised: August 19, 2025; Accepted: December 5, 2025

© The Author(s) 2025. This is an open access article under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0, <http://creativecommons.org/licenses/by/4.0/>).

(Sualeh and Kim, 2020; Zheng et al., 2019), feature-level fusion (Chen et al., 2017; Ku et al., 2018), and data-level fusion (Vora et al., 2020). However, neither feature-level nor object-level fusion fundamentally addresses the issues of low-quality data with sparse point cloud information and high noise, leading to low confidence in the fused features or detections. In contrast, data-level fusion addresses these limitations by operating directly on raw sensor data. Wang et al. (2019) and You et al. (2019) demonstrated that pseudo-LiDAR generated from stereo vision can bridge the performance gap in 3D object detection. Their findings validate the feasibility of data-level fusion methods. These methods enhance sparse point clouds with image-derived spatial information.

Current data-level methods exhibit certain deficiencies. These deficiencies motivate our methodological innovations. For example, PointPainting (Vora et al., 2020) performs fusion in two-dimensional (2D) space. This provides limited benefits for accurate target localization compared to 3D fusion methods. Song et al. (2021) demonstrated that pseudo point cloud enhancement can improve detection accuracy. However, their method lacks filtering or noise reduction for pseudo point cloud data. This limits the improvement in detection accuracy. In this study, a 3D vehicle detection method is proposed, which fuses image and LiDAR data, solves the problem of sparse point representation, and improves the detection accuracy of distant objects. The method first estimates the depth of image pixels and projects them back into 3D space to generate pseudo point clouds. Then, the K-dimensional (KD) tree is employed for position modification, noise and outlier elimination before performing data fusion with the LiDAR point clouds. Finally, a depth-optimized point-based region-based convolutional neural network (PointRCNN) detector trains both the depth map generation network and the 3D vehicle detection network simultaneously to achieve 3D object detection. This method effectively solves the problem of sparse point clouds for distant objects in two ways: (1) Pseudo-Point Cloud Generation: By training and constraining the pixel depth values using the depth-optimized PointRCNN network, the pixel depth can be accurately estimated to enhance the precision of the pseudopoint clouds. (2) Fusion stage: The KD tree is utilized to revise the pseudopoint clouds by eliminating noise and outliers, which guarantees effective matching and fusion with the LiDAR point clouds. This method effectively solves the problem of point cloud sparsity for distant objects from two aspects: (1) Sparse point clouds cannot represent the contours of objects. As the distance of the object increases, the number of point clouds on the surface gradually decreases. When the number of point clouds is not sufficient to accurately represent the object, the detection accuracy decreases dramatically. Our method generates a dense pseudopoint cloud that matches and fuses with the sparse point cloud to more comprehensively represent distant objects. (2) The absence of LiDAR point clouds for distant objects can impair detection accuracy. In such cases, our method directly employs pseudo point clouds for object detection. Experiments conducted on the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset demonstrate that our proposed approach exhibits strong performance in 3D vehicle detection, achieving high detection accuracy for distant vehicles. The main work of this study is as follows:

1) We propose a comprehensive 3D vehicle detection method based on data-level fusion of image and LiDAR data. This method generates dense pseudo point clouds from stereo images to compensate for sparse LiDAR data. It effectively addresses the limitation of distant vehicle representation and improves the detection accuracy of distant targets.

2) In the process of point cloud and image fusion, we propose a pseudopoint cloud correction method. This method utilizes a KD tree to modify the position of pseudopoint clouds. It can effectively remove noise and outliers in pseudo point clouds. This ensures spatial consistency with LiDAR point clouds.

3) We enhance the detection performance of distant vehicles by using a depth-optimized PointRCNN detector. The depth loss eliminates pixel depth errors to generate accurate pseudo point clouds. Experiments demonstrate that our fusion method has improvements in distant vehicle detection. It performs better than existing 3D object detection methods.

2 Related work

According to the different modes of scene data, 3D object detection methods can be categorized into three groups: vision-based methods, LiDAR point cloud-based methods, and multimode fusion methods. The details are discussed as follows.

2.1 Vision-based 3D object detection

Vision-based methods are low-cost and rich in texture features, so they are used extensively in object detection. Furthermore, they are categorized into monocular vision and stereo/depth vision methods.

Monocular-based detection methods mainly include directional truncated signed distance function (DTSDF) (Splietker and Behnke, 2021) and Monocular 3D region proposal network (M3D-RPN) (Brazil and Liu, 2019). DTSDF (Splietker and Behnke, 2021) achieves end-to-end 3D object detection using monocular vision. However, since monocular cameras cannot directly obtain depth information, these methods tend to have lower localization accuracy in multiobject detection scenarios. The end-to-end approach returns 3D data and attitude parameters of the camera directly (Kim and Hwang, 2021), which improves the detection efficiency (Li et al., 2025). M3D-RPN (Brazil and Liu, 2019) introduces an end-to-end region proposal network for multiclass 3D object detection. It employs a unique approach that shares anchor points between 2D and 3D, utilizing the strong relationship between the 2D scale and 3D depth to process the data. While the M3D-RPN significantly improves monocular 3D object detection performance, it still exhibits considerable gaps compared to LiDAR point cloud-based methods.

Stereo/depth vision methods include AD3D (Deng and Latecki, 2017), F-PointNets (Qi et al., 2018), and PL++ (You et al., 2019). AD3D (Deng and Latecki, 2017) employs Fast RCNN (Girshick, 2015) for 2D region extraction and initializes 3D detection boxes using depth features, adjusting the box parameters through a regression network for object localization. F-PointNets (Qi et al., 2018) first utilizes a 2D detection network to recognize objects in monocular images, then converts the image coordinates to world coordinates to determine the point cloud proposal regions, and finally employs a PointNets (Qi et al., 2017a) 3D object recognition network to detect the point clouds within these regions, determining the 3D information such as position and orientation. PL++ (You et al., 2019) predicts the scene depth map from the image, back-projects the image pixels into 3D space to generate a pseudopoint cloud, and ultimately employs a 3D object detector for detection. Compared to monocular systems, stereo cameras can effectively capture the spatial positions of scene objects. However, they remain vulnerable to factors such as lighting, occlusion, and noise, leading to significant localization errors for distant targets.

Recent work has also proposed a framework for object identification (FOI) that utilizes only camera sensors to detect objects and extract behavioral features, including motion, position, velocity, and distance, in complex driving environments (Lateef et al., 2023). While such camera-only approaches demonstrate the potential of comprehensive object understanding from single sensors, they face inherent limitations in accurate depth estimation, particularly for distant objects.

In summary, vision-based (monocular/stereo) 3D object detection methods often lack depth information, resulting in poor accuracy in spatial localization and pose estimation of objects.

2.2 LiDAR point cloud-based 3D object detection

LiDAR point clouds provide accurate depth information and important 3D features compared to image data and thus can be widely used in 3D object detection tasks. Current algorithms utilizing LiDAR point cloud data can be categorized into three paradigms: voxel-based, view-based, and point-based methods. (1) Voxel-Based Models: These models convert raw point clouds into voxels before inputting them into a network for detection. For example, VoxNet (Maturana and Scherer, 2015), processes occupied voxel grids using a 3D convolutional neural network (CNN), enabling reliable 3D object detection. Additionally, VoxelNet (Zhou and Tuzel, 2018) was the first to employ learning methods (PointNet (Qi et al., 2017a)) to represent features within voxel grids, effectively enhancing the utility of voxel features. VoxelNet (Zhou and Tuzel, 2018) voxelizes raw point clouds and learns based on voxel features to output 3D detection boxes. Although these methods demonstrate satisfactory detection performance, their time and space complexity grows cubically with point cloud density, making them less efficient for processing dense 3D point cloud data. (2) View-Based Models: These models project 3D point clouds onto 2D planes to create discrete grids, generating multiview 2D images, which are then processed using 2D CNNs. For instance, multi-view convolutional neural network (MVCNN) (Qi et al., 2016) converts 3D objects into 2D rendered images and fuses image features from multiple views. PIXOR (Yang et al., 2018) generates bird's-eye view maps based on height and reflectance channels from point clouds, employing a structurally fine-tuned RetinaNet for object detection and localization. This method retains spatial depth information and the geometric shape of objects, partially addressing occlusion issues. The PointPillars (Lang et al., 2020) network first converts point clouds into sparse projection images and then utilizes a 2D convolutional network for detection, regressing to generate 3D detection boxes. This approach improves the detection speed by avoiding 3D convolution operations. Overall, these methods have limited performance improvements due to the loss of one-dimensional information during point cloud processing. (3) Point-Based Models: Unlike the previous two approaches, point-based models learn features directly from irregular, unordered 3D point clouds to achieve 3D object detection, thus avoiding information loss during quantization. PointRCNN (Shi et al., 2019) is the first 3D object detection network based on raw point clouds. It innovatively adopts a bottom-up candidate region proposal network inspired by the RCNN (Girshick, 2015) algorithm, extracting effective target regions from each frame of point cloud data and generating candidate boxes, followed by object detection and localization using the PointNet++ (Qi et al., 2017b) algorithm. This method demonstrates high vehicle detection performance on the KITTI (Geiger et al., 2012) dataset. Therefore, this study

chooses PointRCNN (Shi et al., 2019) as the foundation for the 3D object detector applied to the fused dense point clouds.

2.3 Multimodal fusion-based 3D object detection

Currently, algorithms for 3D object detection based on multimodal information focus on how to effectively organize and fuse multimodal data to compensate for the shortcomings of single data streams, ultimately enhancing object detection accuracy. Existing methods for fusing image and point cloud data can be categorized into three main types (Wang and Zhang, 2024): (1) feature-level fusion, (2) object-level fusion, and (3) data-level fusion (Luo and Qin, 2021).

Feature-level fusion is currently the most widely used fusion method, which utilizes convolutional neural networks (CNNs) to separately extract features from point clouds and images and then fuses the two types of features. For instance, MV3D (Chen et al., 2017) converts LiDAR point clouds into bird's-eye view (BEV) and front view (FV) based on viewpoint pose and uses the RGB image corresponding to the same view as input. Then, CNNs are applied to extract feature maps from these three views. 3D candidate boxes are generated based on the bird's-eye view feature map and projected onto the feature maps of the three viewpoints to obtain the corresponding region features. Finally, region features from different views are deeply fused, and boundary region regression and category judgment are performed on the fusion results. AVOD (Ku et al., 2018) extracts features from bird's-eye views constructed from images and LiDAR point clouds, selecting the top K proposal regions based on confidence, followed by cropping and adjusting the features before fusion. ContFuse (Liang et al., 2018) has a structure consisting of two branches: one branch extracts image features, while the other extracts features from the point cloud bird's-eye view. To effectively fuse the image and point cloud feature maps of different resolutions, ContFuse designs continuous fusion layers to link multiple intermediate layers from both network branches and encode the geometric relationships between the two modalities. Continuous convolutions extract information from image features corresponding to each point in the bird's-eye view, ultimately generating precise detection results from the fused "image + bird's-eye view". However, since this method requires simultaneous computation of image and bird's-eye view features, it incurs considerable computational overhead, and the lack of accurate spatial correspondence between the bird's-eye view and the image can lead to inaccuracies in feature fusion.

Object-level fusion methods independently perform object detection on image and LiDAR data and then combine the detection results. For example, Sualeh and Kim (2020) utilized laser point clouds and image data for vehicle detection, determining the class and spatial pose of targets through the fusion of the two datasets' detection outcomes. Zheng et al. (2019) performed object detection on images and point clouds using the Hungarian algorithm to achieve optimal matching of the detection results and selected unmatched detection boxes based on the projection distance from point clouds to images. Following classification of the fusion list, the best candidates are output. Object-level fusion methods do not fully leverage the advantages of both data types, nor do they adequately compensate for their respective weaknesses, resulting in limited improvements in detection accuracy in scenarios with severe occlusion and complex environments.

Data-level fusion matches point clouds and images based on their projection relationships for target recognition. For instance, the PointPainting (Vora et al., 2020) algorithm uses an image

recognition method to segment pixels, which are then used to color the point clouds before processing. Song et al. (2021) proposed an algorithm for pseudo point cloud correction that enhances LiDAR data by merging pseudo point clouds generated from images with LiDAR point clouds to create dense point clouds, resulting in targets with more complete shapes and profiles, thereby effectively improving 3D object detection accuracy. Beyond automotive applications, data-level fusion challenges exist across various domains involving multisensor integration. For instance, Ikonomakis et al. (2022) addressed position measurement validation and correction in maritime navigation by fusing global navigation satellite system (GNSS) and automatic identification system (AIS) data, demonstrating that data quality issues and sensor reliability are fundamental concerns in multimodal sensor fusion systems. However, the precision of the generated dense point clouds depends heavily on the accuracy of surface reconstruction, and since no filtering or noise reduction is applied to the pseudo point cloud data, the improvement in detection accuracy is limited. Research on data-level fusion methods is currently scarce. The method proposed in this study also falls under data-level fusion, transforming the 2D pixels from images into 3D pseudo point cloud data and subsequently fusing it with point clouds in the 3D space. Compared to PointPainting (Qi et al., 2017b), which performs fusion in 2D space, the 3D fusion approach is more beneficial for accurate target localization. Additionally, in contrast to the method presented in Song et al. (2021), this study utilizes a KD tree to filter the pseudopoint clouds, generating denser point cloud data with reduced noise, which enhances the detection accuracy of distant targets.

3 Methodology

The framework of the proposed method is illustrated in Fig. 1, which primarily consists of three parts: pseudo point cloud generation, pseudo point cloud revision and fusion, and depth-optimized vehicle detection. First, pseudo point cloud generation

aims to estimate image depth from the stereo images of the scene and convert pixels into spatial pseudo point clouds. Second, pseudo point clouds are generated after noise reduction and outlier removal. Finally, depth-optimized traffic vehicle detection focuses on optimizing the depth values of the scene image to enhance the performance of 3D vehicle detection.

3.1 Pseudo point cloud generation

Pseudopoint clouds are generated from scene images to create the corresponding spatial point clouds. First, a depth map is established from the scene image, and then the depth map is back-projected into 3D space to generate pseudo point clouds. The accuracy of the pseudopoint cloud depends on the precision of the pixel depth. Currently, depth estimation using stereo vision spatial constraints is more accurate than that using monocular vision (Wang et al., 2019). Therefore, this study uses a stereo vision method to estimate pixel depth by calculating the disparity between pixels in the images (Chang and Chen, 2018). Depth estimation is performed using a pair of images left view l_l and right view l_r with their baseline offset, taking l_l as the reference, and output a disparity map $D(u, v)$. $D(u, v)$ records the horizontal disparity of each pixel (u, v) to l_r . In an ideal scenario, both $l_l(u, v)$ and $l_r(u, v + D(u, v))$ correspond to a 3D point in spatial coordinates. Consequently, the depth map can be calculated as

$$Z(u, v) = \frac{f_u \times b}{D(u, v)} \quad (1)$$

where f_u is the camera's horizontal focal length and $Z(u, v)$ is the depth value of pixel (u, v) in the left view. Once the pixel depth is determined, the spatial coordinates corresponding to the pixel can be calculated using:

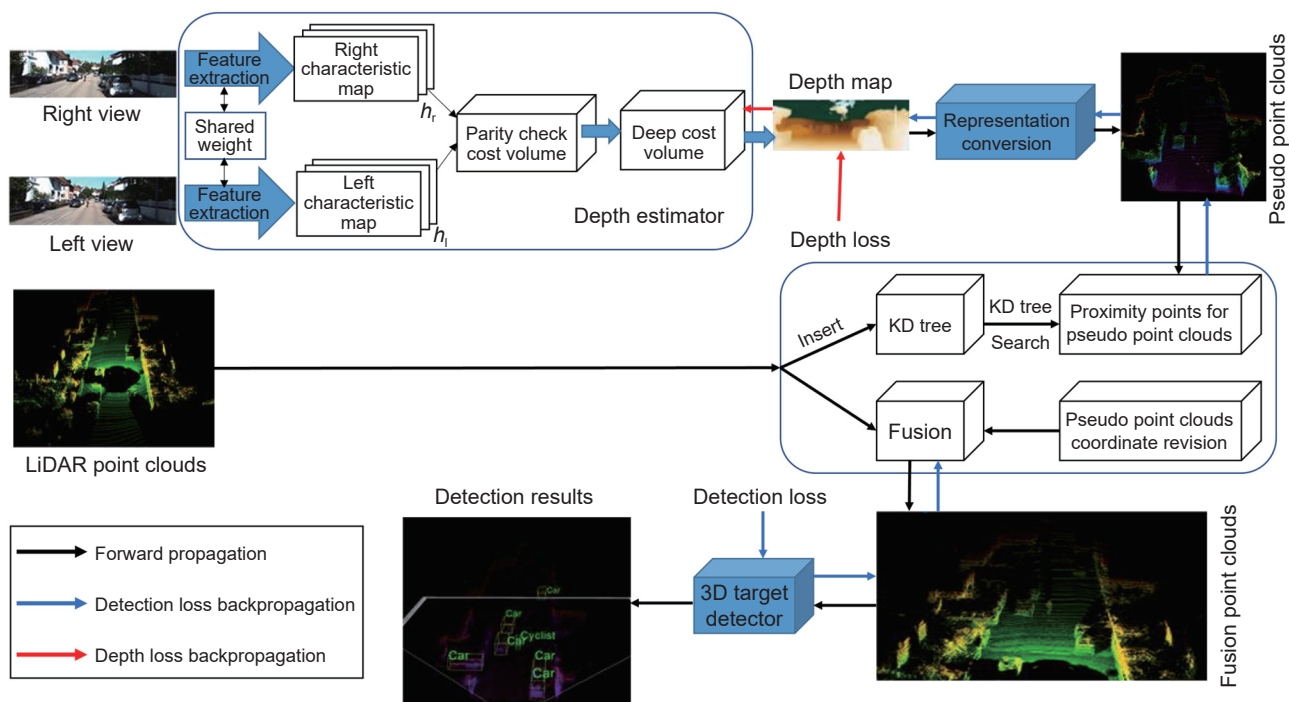


Fig. 1 Overview of our proposed framework: First, utilize the stereo images to estimate the depth image; then, the image points are projected to 3D space to generate the pseudo point clouds; in the stage of fusion, utilize the KD tree to revise the position of pseudo point clouds, eliminate the noises and outliers; finally, feed fusion data to the detection network for training and detection.

$$\begin{cases} x = \frac{(u - c_u) \times z}{f_u} \\ y = \frac{(v - c_v) \times z}{f_v} \\ z = Z(u, v) \end{cases} \quad (2)$$

where (c_u, c_v) is the camera center, f_u is the focal length in the horizontal direction, and f_v is the focal length in the vertical directions. After converting pixels into spatial point clouds in this manner, the generated pseudo point clouds can be treated as signals from a LiDAR sensor for further processing.

Disparity refers to the vector difference in the position of corresponding pixel points, which results from projecting a point in space onto the left and right imaging planes. As shown in Eq. (1), the quality of the disparity map $D(u, v)$ directly impacts the accuracy of depth estimation, indirectly reflecting the depth information of objects. This study utilizes an advanced method for disparity estimation, PSMNet (Chang and Chen, 2018), which is employed in PL++ (You et al., 2019). However, the disparity estimation network can overly focus on nearby objects, leading to relatively inaccurate depth estimation for distant objects. For example, in the KITTI (Geiger et al., 2012) dataset, the depth error for a single pixel at a distance of 5 m is 0.1 m, while at a depth of a point far from 50 m, the depth error is 5.8 m. As indicated in Eq. (3), the depth error $\delta Z(u, v)$ increases quadratically with the image depth for a given visual error δD .

$$Z(u, v) \propto \frac{1}{D} \Rightarrow \delta Z(u, v) \propto \frac{1}{D^2} \delta D \Rightarrow \delta Z(u, v) \propto Z(u, v)^2 \delta D \quad (3)$$

Therefore, we utilize SDN to optimize the stereo network for direct depth estimation and directly optimize the depth loss by learning the stereo network:

$$\sum_{(u, v) \in A} \uparrow (Z(u, v) - Z^*(u, v)) \quad (4)$$

where Z and Z^* can be obtained using Eq. (1). This process ensures the accuracy of depth estimations across a range of distances. Additionally, a new depth layer is constructed to predict pixel depth (see Eq. (5)), further narrowing the error gap.

$$Z(u, v) = \sum_z \text{softmax}(-S_{\text{depth}}(u, v, z)) \times z(u, v) \quad (5)$$

3.2 Pseudo point cloud revision and fusion

3.2.1 Pseudo point cloud preprocessing

The processing of pseudo point clouds primarily involves removing redundant and irrelevant information from the point clouds to enhance the speed and accuracy of the object detection. Irrelevant information refers to a large number of point clouds that are not related to the target (e.g., sky, ground), and redundant information refers to high-density point clouds within scene targets (e.g., dense coverage of vehicles).

First, irrelevant information (such as the sky and ground) is eliminated since the presence of these point clouds can affect the subsequent position revision of the pseudopoint clouds. This study uses the algorithm (Himmelsbach et al., 2010) for ground segmentation, which can quickly segment steeply sloped ground with continuously changing curvature, so the algorithm is suitable for complex traffic scenarios.

Then, regarding the redundant information in pseudo point clouds, the number of point clouds generated from the depth map

far exceeds that of the LiDAR point clouds. For instance, a pseudo LiDAR may generate an average of 300,000 points per frame, while a real LiDAR produces only 18,000 points per frame. Although denser point clouds can improve detection accuracy, they significantly reduce the operation speed of the object detection network. Therefore, downsampling the pseudo point clouds is necessary. Here, an angle-based sparsification method (Qian et al., 2020) is used to sample the pseudopoint clouds after defining multiple intervals in 3D space via discretized spherical coordinates. First, the pseudo point clouds are transformed to the spherical coordinate system. Second, the polar and azimuthal angles of each point cloud are quantized. Finally, from the pseudo point clouds within the same interval, randomly select some to be retained, and eliminate the remaining pseudo point clouds to achieve the downsampling.

3.2.2 Pseudo point cloud position modification

This section primarily focuses on revising the deviations between pseudopoint clouds and LiDAR data to achieve an accurate fusion of the two data. Due to the discreteness of image pixels, stereo vision has a fundamental limitation: disparity, the difference in horizontal coordinates between corresponding pixels, must be quantified at the individual pixel level, and estimating successive depths using discrete pixels will result in quantization errors. Although increasing image resolution can reduce quantization errors, the time complexity of depth estimation cubically grows with increasing image resolution. Currently, there are two main cases of pseudo point cloud coordinate deviation caused by visual perspective. In the first situation, as shown in Fig. 2, pedestrian targets are stretched far beyond their actual physical proportions, resulting in a “tail” in the target point cloud. This issue prevents precise localization of the target position, thereby affecting the accuracy of the generated detection boxes. The “tails” cannot be filtered out with a radius filter. In the second situation, as shown in Fig. 3, pseudo point cloud coordinate shifts and target angle deviations caused by depth estimation errors make the fused point cloud unable to accurately represent the features of the target. Therefore, this study proposes a pseudo point cloud revision method that utilizes the original point clouds to rectify these deviations. This method effectively solves the above issues and improves the fusion of pseudo point clouds with LiDAR point clouds, ultimately enhancing 3D object detection results.

The KD tree is a tree data structure for organizing and retrieving KD space data points. This study uses the KD tree for fast nearest neighbor point searches to obtain three nearby LiDAR points centered around the pseudo point cloud. We then determine whether these three points originate from the same LiDAR scan line. If the three points are collinear, we utilize these points to correct the coordinates of the pseudo point cloud. If they are not collinear, the three points define a plane, and we project the point onto this plane for coordinate correction. Let $\{o_i\}_{i=1}^{N_o}$, $\{P_i\}_{i=1}^{N_p}$, and $\{FP_i\}_{i=1}^{N_{fp}}$ represent the original point cloud, the original point cloud after ground removal, and the point cloud pseudo after ground removal, respectively, with N_o , N_p , and N_{fp} denoting the number of points in each cloud. The points from N_{fp} are sequentially added to $\{P_i\}_{i=1}^{N_p}$ to obtain $\{P_i\}_{i=1}^{N_p} *$, and $\{P_i\}_{i=1}^{N_p} *$ are inserted into a 3D KD tree. Using the newly added pseudo point cloud $P_f(x, y, z)$ as an anchor point, we search for the three nearest points within a radius r in the KD tree. Let the search results be $\{Q_i\}_{i=1}^{N_Q}$, where N_Q is the number of nearby points found. If $N_Q = 3$, we proceed P_f to the next step. If $N_Q < 3$, we consider P_f to be a distant point or an outlier, then calculate the distance of P_f and, if it exceeds a set threshold, we regard it as a point detected by the camera but not by LiDAR, retaining its

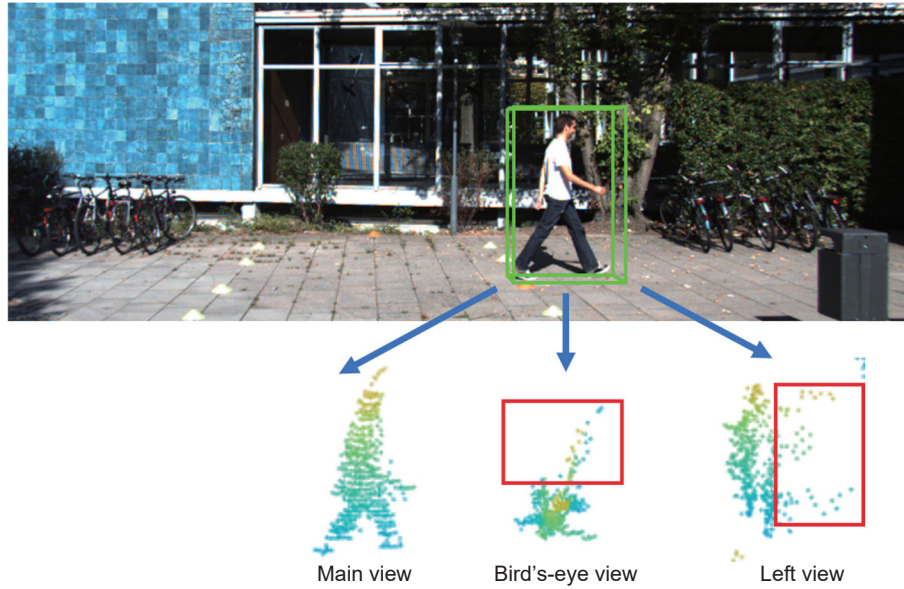


Fig. 2 Outliers in the pseudo point clouds.

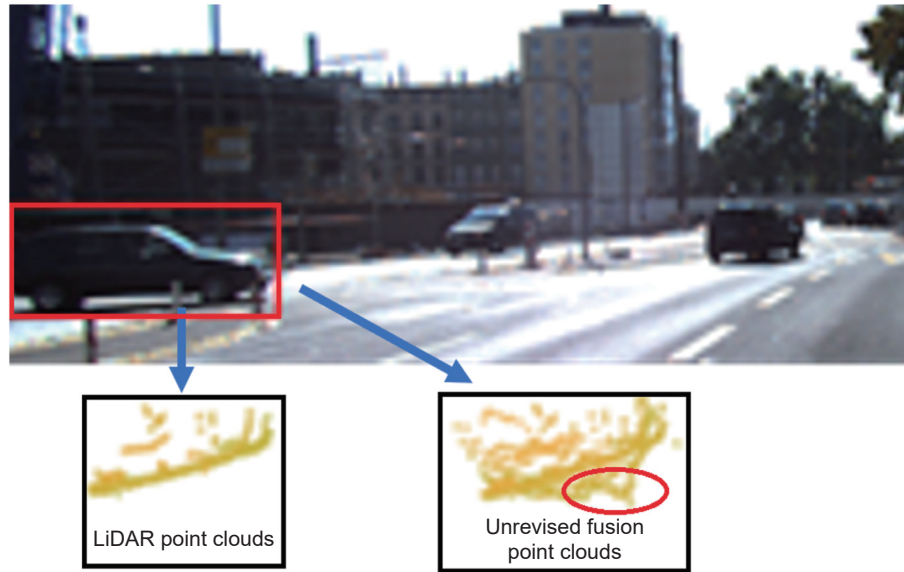


Fig. 3 Position shifting of pseudo point clouds.

coordinates. Otherwise, we consider it an outlier, discard it and proceed to the next iteration.

The collinearity of three points should be determined. Without loss of generality, three points are expressed as $Q_1(x_1, y_1, z_1)$, $Q_2(x_2, y_2, z_2)$, and $Q_3(x_3, y_3, z_3)$, and their Euclidean distances are expressed as D_{12} , D_{13} , and D_{23} . Then, the unit vectors e_{12} , e_{13} , and e_{23} are obtained by $e_{ij} = \frac{P_i P_j}{|P_i P_j|}$; if e_{12} is equal to e_{23} , the three points are collinear. The deviations in the pseudo point clouds primarily occur in the x and y coordinates, and the corrected coordinates of the pseudo point $P_f(x, y, z)$ are given by

$$\begin{cases} x^* = \frac{x_1 + x_2 + x_3}{3} \\ y^* = \frac{y_1 + y_2 + y_3}{3} \\ z^* = z \end{cases} \quad (6)$$

If the three points are not collinear, the pseudo points are projected onto the plane defined by three points. The equation of the plane can be expressed as $Ax + By + Cz + D = 0$.

$$\begin{cases} A = (y_3 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1) \\ B = (x_3 - x_1)(z_3 - z_1) - (x_2 - x_1)(z_3 - z_1) \\ C = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \\ D = -(Ax_3 + By_3 + Cz_3) \end{cases} \quad (7)$$

where the line L_{OP} is determined by the origin O of the LiDAR coordinate system and the pseudo point P_f . We calculate the intersection point C of line L_{OP} and plane α that contains the triangle formed by the three points $\{Q_i\}_{i=1}^{N_Q}$ and verify whether point C lies inside the triangle. If C is inside the triangle, then C is considered the corrected pseudo point cloud. If C lies outside, we calculate the distance D between C' and P_f , where C' is the intersection of the normal vector of the plane defined by the three points. A threshold is set to evaluate whether point C' is noise or not. When the distance D is less than the threshold, point C' is regarded as a noisy point; otherwise, point C' is revised.

3.3 Depth optimization-based PointRCNN detector

This study proposes an enhanced PointRCNN network for 3D

vehicle detection by adding depth loss to the original loss function. The depth loss can eliminate pixel depth errors to correct the accurate pseudo point cloud for improving the detection of vehicles at long distances.

Due to resolution limitations, LiDAR often cannot capture point cloud data for distant targets. In such cases, the pseudo point cloud correction method based on LiDAR point clouds becomes ineffective, and distant targets can only rely on uncorrected pseudo points. This situation leads to two main issues.

On the one hand, point cloud-based object detectors heavily depend on precise 3D point clouds located on or near the object surface, which results in poor detection performance for distant targets with sparse point clouds. On the other hand, since the background and nearby objects occupy a significant proportion of the pixels in the image, training the depth estimator can overly emphasize the background and nearby objects while neglecting distant targets (Chen et al., 2015). This results in a significant amount of noise in the pseudopoint clouds generated from the image for distant objects. Therefore, by incorporating pixel depth as a loss term in the PointRCNN network, the pixel depth can be optimized after reducing pseudo point cloud noise. The joint loss for the object detector and depth estimator is given by

$$L = \lambda_{\text{det}} L_{\text{det}} + \lambda_{\text{depth}} L_{\text{depth}} \quad (8)$$

where L_{det} is the 3D object detection loss, L_{depth} is the depth estimation loss, and λ_{det} and λ_{depth} are the corresponding coefficients for the respective losses. The object detection loss L_{det} comprises the object classification loss L_{cls} and the candidate box regression loss L_{reg} :

$$L_{\text{det}} = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{reg}} L_{\text{reg}} \quad (9)$$

The classification loss aims to assign the correct label (e.g., pedestrian, vehicle) for detected 3D boxes. The regression loss focuses on optimizing the size, position, and orientation of these candidate boxes. Let Z denote the predicted depth and Z^* denote the corresponding ground truth. The depth estimation loss can be written as

$$L_{\text{depth}} = \frac{1}{|A|} \sum_{(u,v) \in A} \uparrow(Z(u,v) - Z^*(u,v)) \quad (10)$$

where A is the set of pixels with ground truth depth, and $\uparrow(x)$ is the smooth loss $L1$, defined as

$$\uparrow(x) = \begin{cases} 0.5x^2, & x < 1 \\ |x| - 0.5, & \text{else} \end{cases} \quad (11)$$

The importance of the depth loss lies in its ability to compensate for the pixels that are affected by the object detection loss. Without sacrificing the accuracy of depth estimation for the background and nearby objects, depth loss can facilitate accurate depth estimation for distant objects (Qian et al., 2020).

4 Experiments and analysis

The experiments are divided into four parts: comparative experiments with existing methods, ablation experiments, case studies of long-distance vehicle detection, and visualization. Through these experiments, the effectiveness of the proposed method is validated.

4.1 Experimental setup

4.1.1 Experimental dataset

All algorithms discussed in this study are trained and evaluated on the KITTI (Geiger et al., 2012) open-source dataset, which contains 7481 training samples and 7518 testing samples. Since the labels for the testing set are not publicly available, we utilize the training set to evaluate the detection methods. The KITTI dataset is divided into two parts: 3712 samples for training and 3769 samples for testing (Chen et al., 2015). For each sample, KITTI provides corresponding left and right images, 64-line LiDAR point clouds, camera calibration matrices, and ground truth for object detection. Additionally, KITTI has been classified into three levels: easy, moderate, and hard according to the target size and occlusion status.

4.1.2 Evaluation metrics

The primary evaluation metrics for object detection are AP_{3D} (average precision of 3D object detection), AP_{BEV} (average precision of bird's eye view), and the precision recall (PR) curve. Although precision accuracy and recall rate are not directly used in this experiment, the metrics, AP_{3D} , AP_{BEV} , and the PR curve are calculated based on these metrics. The evaluation metrics are elaborated as follows.

Precision accuracy represents the proportion of correctly identified samples among all detected results. Recall indicates the percentage of accurately detected objects out of the total number of objects in the corresponding dataset. These metrics are defined by

$$\text{precision} = \frac{TP}{TP + FP} \times 100\% \quad (12)$$

$$\text{recall} = \frac{TP}{TP + FN} \times 100\% \quad (13)$$

where TP is the number of true positives (i.e., samples where both predicted and true values are positive), FP is the number of false positives (samples where the predicted value is positive but the true value is negative), and FN is the number of false negatives (i.e., samples where the predicted value is negative while the true value is positive).

The PR curve reflects the relationship between precision accuracy and recall rate at different thresholds, providing insight into the overall detection capability of the model. The area under the PR curve represents the average precision (AP). A larger area indicates higher average precision and better model performance. AP_{3D} and AP_{BEV} represent the AP (average precision) for the 3D bounding box and bird's eye view tasks, respectively.

4.1.3 Comparative experiments with existing fusion methods

All experiments were conducted on a computer with Intel Xeon(R) E5-2678 v3 @ 2.50 GHz octa-core CPU, 16 GB of RAM and Nvidia GeForce GTX 1080 Ti. All methods are implemented by the Python code with the PyTorch 1.1 framework and trained with CUDA 10.0 and cuDNN 7.4.

4.2 Comparative experiments with existing fusion methods

This study follows the official KITTI evaluation protocol and assesses vehicle detection results under over union (IoU) thresholds of 0.7 and 0.5. The proposed methods are compared with four multimodal fusion detection methods: MV3D, AVOD, F-PointNet, and ContFuse.

Table 1 shows the detection results of the bird's eye view between our method and other state-of-the-art fusion detection methods. The proposed method outperforms other methods at all

Table 1 Detection results in bird's eye view: AP on the KITTI dataset

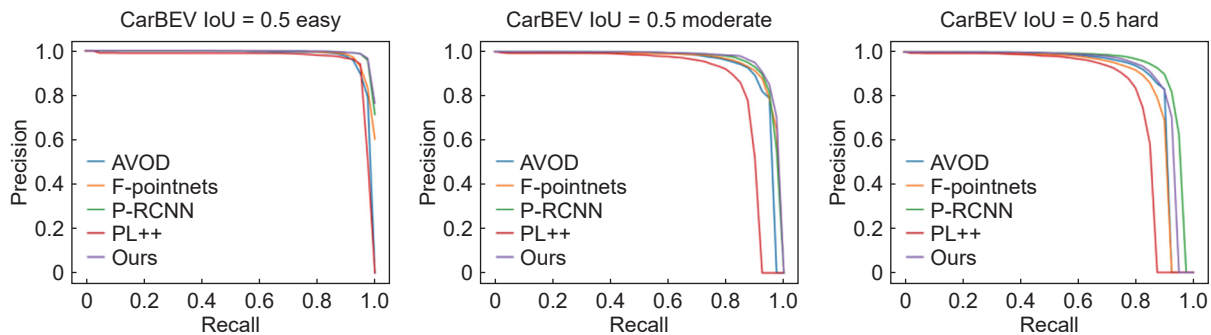
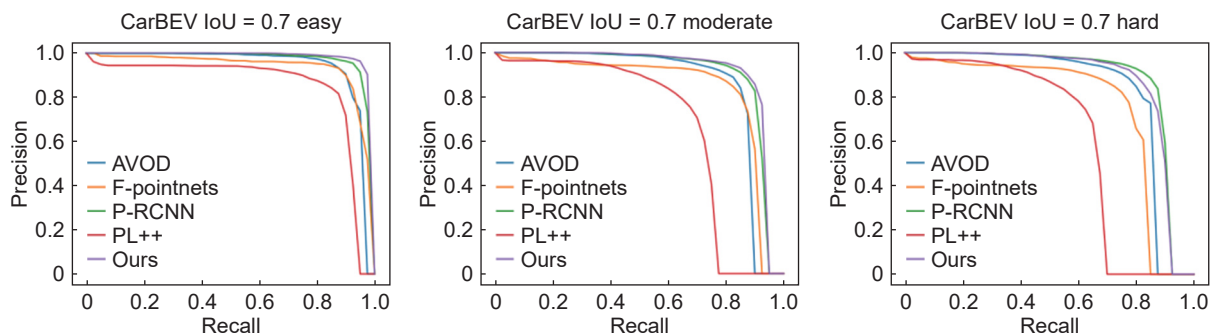
(Unit: %)

Approach	Data	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Difficult	Easy	Moderate	Difficult
MV3D	Image + LiDAR	—	—	—	86.0	76.9	68.5
AVOD-FPN	Image + LiDAR	90.5	89.3	88.3	88.5	83.7	77.9
F-PointNet	Image + LiDAR	96.1	89.4	86.5	88.7	84.0	75.3
ContFuse	Image + LiDAR	—	—	—	88.8	85.8	77.3
This study	Image + LiDAR	97.4	90.8	87.4	92.3	88.4	84.2

difficulty levels except for the difficulty level at $\text{IoU} = 0.5$, indicating that the proposed detection method is effective. ContFuse, which employs continuous convolutions to merge image features with bird's eye view features for detection, achieved good results in bird's eye view detection. However, our method still leads by 6.9% in the difficult level at $\text{IoU} = 0.7$. Additionally, compared to MV3D, F-PointNet, and AVOD-FPN, our method improves by 15.7%, 6.3%, and 8.9%, respectively, demonstrating a strong localization capability for vehicles in difficult scenarios.

Figs. 4 and 5 display the PR curves for bird's eye view localization under the two IoU conditions of 0.5 and 0.7 for our method and existing classic approaches. It is evident that our method outperforms other comparative methods in recognition accuracy rates at each difficulty level. Even at the difficult level, where it is slightly lower than the pure point cloud input method P-RCNN, it still reflects the effectiveness of our fusion approach in

bird's eye view localization. In contrast to bird's eye view object detection, 3D object detection more accurately reflects the performance of detection algorithms and presents additional challenges. This is because 3D detection has two additional degrees of estimations for object localization compared to the bird's eye view, specifically the position and scale of objects along the Z-axis. The comparison results of our method with the other fusion detection methods based on the AP_{3D} metric are shown in Table 2. It is clear that our fusion detection method surpasses comparison methods in vehicle detection, achieving the best results in multiple metrics. For instance, at $\text{IoU} = 0.5$, our method improves by 1.5% in the easy level compared to F-PointNet and by 0.5% in the moderate level compared to AVOD-FPN. At $\text{IoU} = 0.7$, our method improves by 6.2% in the easy level compared to ContFuse and by 7.1% in the moderate level compared to AVOD-FPN.

**Fig. 4** PR figure of bird-eye view detection ($\text{IoU} = 0.5$).**Fig. 5** PR figure of bird-eye view detection ($\text{IoU} = 0.7$).**Table 2** Detection results in 3D space: AP on the KITTI dataset

(Unit: %)

Approach	Data	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Difficult	Easy	Moderate	Difficult
MV3D	Image + LiDAR	—	—	—	71.0	62.3	55.1
AVOD-FPN	Image + LiDAR	90.5	89.2	88.1	81.9	71.8	66.3
F-PointNet	Image + LiDAR	95.9	89.0	79.8	81.2	70.4	62.2
ContFuse	Image + LiDAR	—	—	—	82.5	66.2	64.0
This study	Image + LiDAR	97.4	89.7	87.1	88.7	78.9	76.2

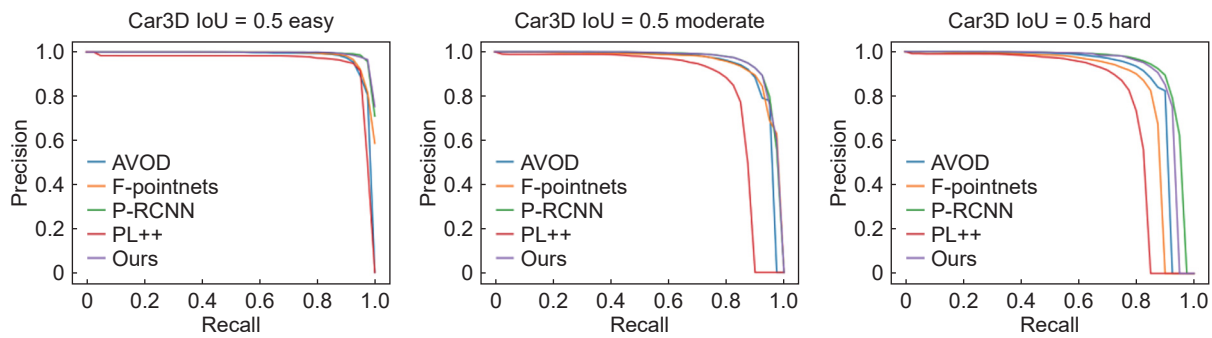


Fig. 6 PR figure of 3D vehicle detection (IoU = 0.5).

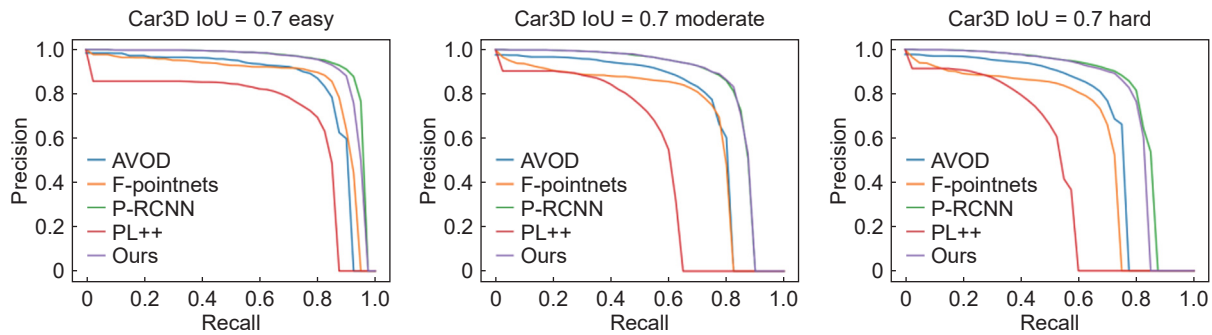


Fig. 7 PR figure of 3D vehicle detection (IoU = 0.7).

Figs. 6 and 7 illustrate the comparison of PR curves for 3D vehicle detection between our method and existing classic methods under IoU values of 0.5 and 0.7. It is clear that the proposed method has a higher recognition rate at each difficulty level compared to existing fusion methods, although it is slightly lower than the pure point cloud input method P-RCNN. This demonstrates the effectiveness of our fusion method in 3D vehicle detection.

4.3 Ablation experiments

This experiment primarily verifies the effectiveness of the pseudo point cloud correction module within the overall framework. By comparing the detection performance of the proposed algorithm with and without the pseudo point cloud correction module, we demonstrate the validity of our point cloud correction module. What we need to declare is that the “unrevised” and “revised” models trained with the same network architecture and loss functions. Additionally, we establish the effectiveness of our fusion method by comparing it to purely point cloud-based and image-based methods.

As shown in Table 3, using the pseudo point cloud correction module significantly improves both AP_{BEV} and AP_{3D} among the

three difficulty levels. Specifically, at the item of 0.5 IoU, the improvements of average precision are 1.1%, and 1.23% in the aspects of AP_{BEV} and AP_{3D} , respectively. At the item of 0.7 IoU, the improvements of average precision are 4.76% and 3.5% in the aspects of AP_{BEV} and AP_{3D} , respectively. These pronounced enhancements prove that the pseudo point cloud correction module is effective for noise reduction and vehicle detection.

Table 4 shows the comparative results between our method, the existing image-based method PL++ and the point cloud-based method P-RCNN. It is evident that our fusion method significantly outperforms PL++ and surpasses P-RCNN at both the easy and moderate levels. This confirms the effectiveness of the proposed fusion method.

4.4 Case study: Long-distance vehicle detection

As the distance increases, the density of LiDAR point clouds decreases rapidly. Especially for distant vehicles, sparse point clouds cannot effectively represent vehicle shapes or contours. Therefore, 3D detection of long-distance vehicles becomes challenging. This experiment aims to evaluate the detection ability of distant vehicles.

Table 5 shows the detection results of comparative methods for

Table 3 Detection results by revised and unrevised methods: AP_{BEV}/AP_{3D} on the KITTI dataset

(Unit: %)

Methods	IoU = 0.5			IoU = 0.7		
	Easy	Moderate	Difficult	Easy	Moderate	Difficult
Unrevised	96.4/96.3	89.0/88.7	87.0/85.5	89.3/83.9	81.9/76.0	79.4/73.4
Revised	97.4/97.4	90.8/89.7	87.4/87.1	92.3/88.7	88.4/78.9	84.2/76.2

Table 4 Average precision (AP_{BEV}/AP_{3D}) by fusion and single-data vehicle detection methods on the KITTI dataset

(Unit: %)

Methods	Input	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Difficult	Easy	Moderate	Difficult
PL++: P-RCNN	Image	89.8/89.7	83.8/78.6	77.5/75.1	82.0/67.9	64.0/50.1	57.3/45.3
P-RCNN	LiDAR	97.3/97.3	89.9/89.8	89.4/89.3	90.2/89.2	87.9/78.9	85.5/77.9
This study	Image + LiDAR	97.4/97.4	90.8/89.7	87.4/87.1	92.3/88.7	88.4/78.9	84.2/76.2

Table 5 APBEV/AP3D value of distant vehicle detection by comparative methods on the KITTI dataset

(Unit: %)

Methods	Input	> 30 m		> 40 m		> 50 m	
		IoU = 0.5	IoU = 0.7	IoU = 0.5	IoU = 0.7	IoU = 0.5	IoU = 0.7
PL++-P-RCNN	Image	62.2/54.5	32.6/20.0	46.3/39.8	19.3/6.8	11.2/4.6	10.2/4.2
P-RCNN	LiDAR	77.5/76.9	71.2/56.4	46.7/45.2	38.4/26.3	14.0/13.4	12.5/9.0
AVOD-FPN	Image + LiDAR	65.6/63.3	57.9/41.6	41.9/41.5	34.5/22.1	9.1/9.1	9.1/9.1
F-PointNet	Image + LiDAR	79.4/77.9	64.4/42.4	57.7/51.9	34.0/18.0	25.8/25.6	11.5/3.7
Ours	Image + LiDAR	76.3/75.2	72.0/63.9	62.2/60.3	43.6/27.1	21.4/19.7	15.4/12.3

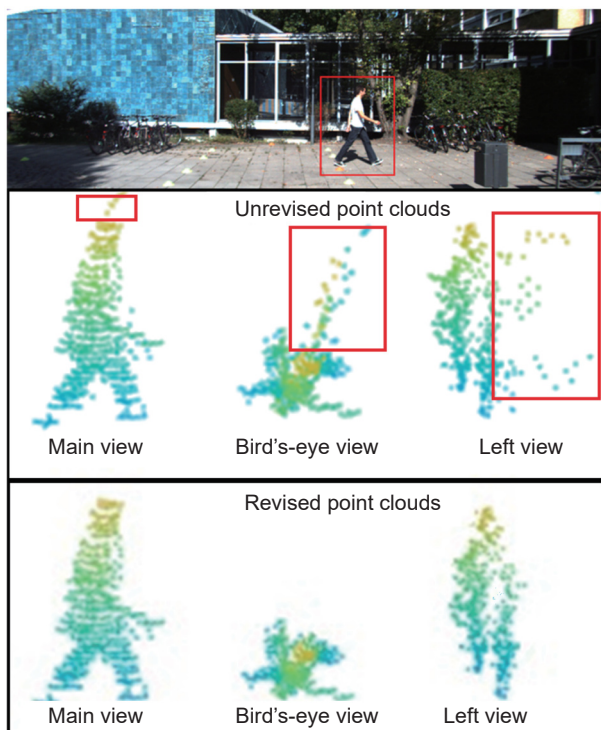
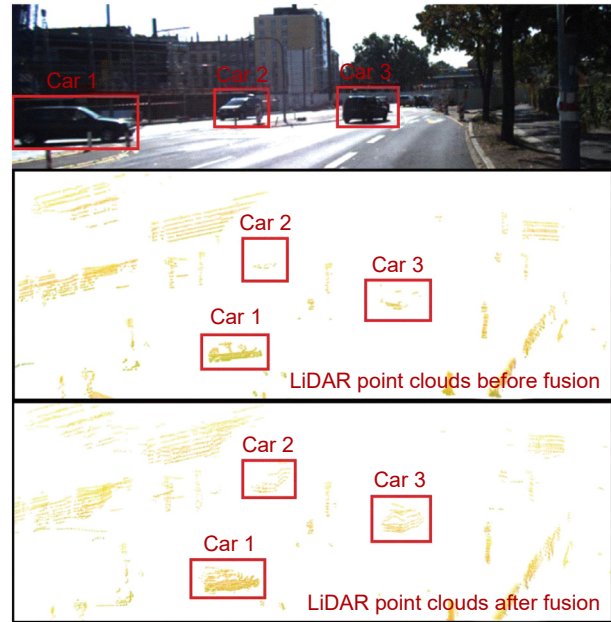
vehicles beyond 30, 40, and 50 m. Compared to the comparative methods, our method can detect distant vehicles more accurately in most situations except those of “IoU = 0.5, > 30 m” and “IoU = 0.5, > 50 m”.

4.5 Visualization of vehicle detection

Fig. 8 depicts that a pedestrian is walking on the road. The middle and bottom of Fig. 8 show the original point cloud and revised point cloud in main, bird's, and left views. The red boxes in the middle subfigure indicate the position noise of the point clouds caused by inaccurate disparity estimation. After modifying the positions of the pseudopoint clouds, the noise and outliers are significantly reduced in the revised point cloud, which contributes to accurate detection.

Fig. 9 presents the image, original point cloud, and fused point cloud of a road traffic scene, where the vehicle targets are marked with red boxes. A comparison of the middle and bottom subfigures shows that the number of point clouds has significantly increased after fusion. For example, the point clouds of “Car 2” and “Car 3” in the middle subfigure fail to effectively represent the characteristics of the vehicles due to the far distances. However, the fused point clouds exhibit a more complete shape and contour for effective feature extraction, which is beneficial for 3D object perception.

Fig. 10 shows some results of 3D vehicle detection, which includes highly accurate 3D bounding boxes for the “vehicle”

**Fig. 8** Figures of original and fusion pseudo point clouds.**Fig. 9** Comparison figures of original and fusion clouds.

category. The red boxes represent the predicted bounding boxes, while the green boxes denote the ground truth. It is clear that, except for a few cases of severe occlusion, our method effectively achieves vehicle detection tasks in most scenarios with high detection accuracy. Additionally, it can accurately detect vehicles at long distances.

5 Conclusions

This study proposes a 3D vehicle detection method based on data-level fusion of image and LiDAR data. First, the method estimates the depth information of scene image pixels and generates pseudo point clouds using the depth information. Second, the method fuses LiDAR point clouds and pseudo point clouds. During the fusion process, the KD tree is used for pseudo point cloud position correction and compensation. Meanwhile, noise and outliers in the pseudo point clouds are eliminated. Finally, the enhanced PointRCNN network is used to train and detect the fused point cloud data. Experiments on the KITTI dataset show that the fused point clouds have significantly improved point cloud density compared to the original LiDAR point clouds. Objects above the ground in the point clouds have visually more complete shapes and contours. Object features are also more obvious. The proposed method can effectively complete 3D vehicle detection tasks in most scenarios with high detection accuracy. The detection performance is significantly improved, especially for distant vehicles.

In future research, the detection of other targets, such as pedestrians and bicycles, in traffic scenarios needs to be conducted. Multiple different types of 3D datasets should be used to verify the generalizability of the proposed method. Due to

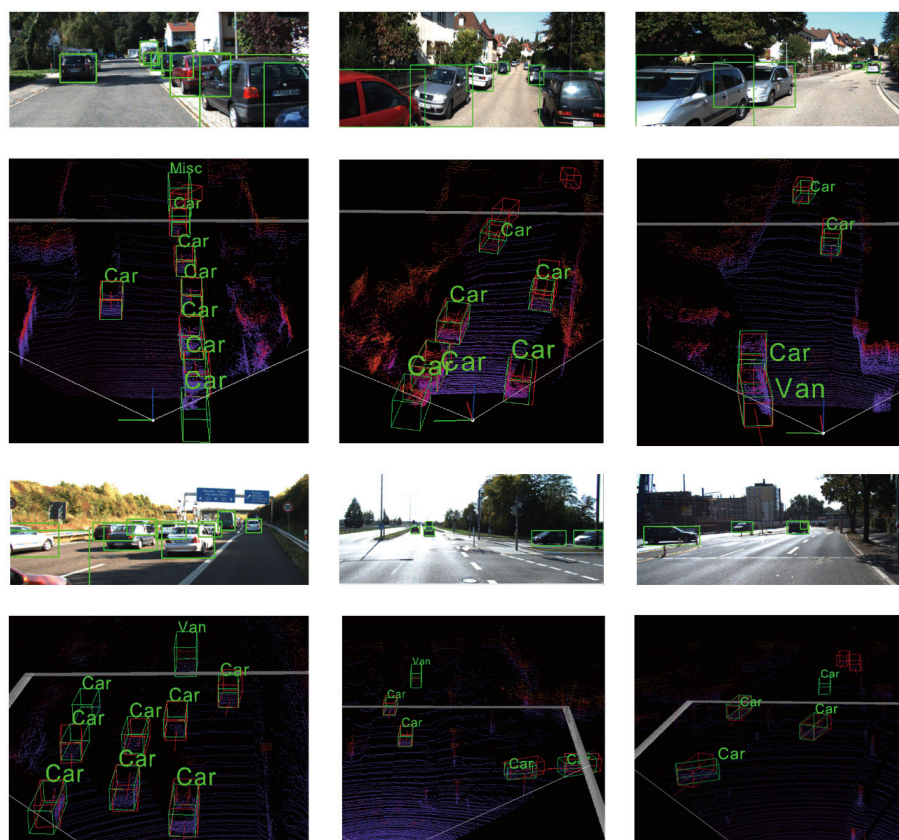


Fig. 10 Vehicle detection on the LiDAR and images (green boxes: ground truth, red boxes: detection vehicle).

experimental limitations, the proposed detection method has not been tested on real vehicles. In future research, this study can be implemented in intelligent connected vehicles to better validate the research results.

Replication and data sharing

All algorithms discussed in this study are trained and evaluated on the KITTI open-source dataset. The code used to support the findings of this study can be downloaded from <https://github.com/RuiniZ/CodeForJICV>.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 52432013), the Shaanxi Province Outstanding Youth Scholar Scientific Project (No. 2023-JC-JQ-45), and the Natural Science Foundation of Shaanxi Province (No. 2025JC-YBMS-408).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

- Brazil, G., Liu, X., 2019. M3D-RPN: Monocular 3D region proposal network for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 9286–9295.
- Cao, P., Xiong, Z., Liu, X., 2022. An analytical model for quantifying the efficiency of traffic-data collection using instrumented vehicles. *Transp Res Part C Emerg Technol*, **136**, 103558.
- Chang, J. R., Chen, Y. S., 2018. Pyramid stereo matching network. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5410–5418.
- Chen, H., Guo, L. L., Gong, X., Gao, B. Z., 2020. Automotive control in

intelligent era. *Acta Automatica Sinica*, **46**, 1313–1332.

- Chen, X., Kundu, K., Zhu, Y., Berneshawi, A., Ma, H., Fidler, S., et al., 2015. 3D object proposals for accurate object class detection. In: Proceedings of the 29th International Conference on Neural Information Processing Systems, 424–432.
- Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017. Multi-view 3D object detection network for autonomous driving. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6526–6534.
- Deng, Z., Latecki, L. J., 2017. Amodal detection of 3D objects: Inferring 3D bounding boxes from 2D ones in RGB-depth images. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 398–406.
- Gao, Z., Jia, B., Xie, D., Wang, W., Wu, J., 2025. A discussion on the complexity and transit mechanisms of urban traffic systems. *Engineering*, **44**, 24–29.
- Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 3354–3361.
- Girshick, R., 2015. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), 1440–1448.
- Hao, H., Wang, Y., Chen, J., 2024. Empowering scenario planning with artificial intelligence: A perspective on building smart and resilient cities. *Engineering*, **43**, 272–283.
- He, Y., Gao, Y., Liu, Y., Qu, X., 2025. LiDAR-IMU SLAM framework in autonomous modular bus docking systems. *Green Energy Intell Transp*, **4**, 100343.
- Himmelsbach, M., Hundelshausen, F. V., Wuensche, H. J., 2010. Fast segmentation of 3D point clouds for ground vehicles. In: 2010 IEEE Intelligent Vehicles Symposium, 560–565.
- Ikonomakis, A., Nielsen, U. D., Holst, K. K., Dietz, J., Galeazzi, R., 2022. Validation and correction of auto-logged position measurements. *Commun Transp Res*, **2**, 100051.
- Kim, S. H., Hwang, Y., 2021. A survey on deep learning based methods and datasets for monocular 3D object detection. *Electronics*, **10**, 517.

- Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S. L., 2018. Joint 3D proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1–8.
- Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2020. PointPillars: Fast encoders for object detection from point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12689–12697.
- Lateef, F., Kas, M., Ruichek, Y., 2023. Motion and geometry-related information fusion through a framework for object identification from a moving camera in urban driving scenarios. *Transp Res Part C Emerg Technol*, **155**, 104271.
- Lesani, A., Nateghinia, E., Miranda-Moreno, L. F., 2020. Development and evaluation of a real-time pedestrian counting system for high-volume conditions based on 2D LiDAR. *Transp Res Part C Emerg Technol*, **114**, 20–35.
- Li, M., Beck, B., Rathnayake, T., Meng, L., Chen, Z., Cosgun, A., et al., 2025. A benchmark for cycling close pass detection from video streams. *Transp Res Part C Emerg Technol*, **174**, 105112.
- Li, Y., Allu, K. R., Sun, Z., Tok, A. Y. C., Feng, G., Ritchie, S. G., 2021. Truck body type classification using a deep representation learning ensemble on 3D point sets. *Transp Res Part C Emerg Technol*, **133**, 103461.
- Liang, M., Yang, B., Wang, S., Urtasun, R., 2018. Deep continuous fusion for multi-sensor 3D object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), 663–678.
- Lin, C., Tian, D., Duan, X., Zhou, J., Zhao, D., Cao, D., 2022. CL3D: Camera-LiDAR 3D object detection with point feature enhancement and point-guided fusion. *IEEE Trans Intell Transp Syst*, **23**, 18040–18050.
- Liu, Y., Wu, F., Liu, Z., Wang, K., Wang, F., Qu, X., 2023. Can language models be used for real-world urban-delivery route optimization? *Innovation*, **4**, 100520.
- Luo, Y. T., Qin, H., 2021. 3D object detection method for autonomous vehicle based on sparse color point cloud. *Automot Eng*, **43**, 492–500.
- Maturana, D., Scherer, S., 2015. VoxNet: A 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 922–928.
- Qi, C. R., Liu, W., Wu, C., Su, H., Guibas, L. J., 2018. Frustum PointNets for 3D object detection from RGB-D data. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 918–927.
- Qi, C., R., Su, H., Mo, K., Guibas, L. J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 77–85.
- Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L. J., 2016. Volumetric and multi-view CNNs for object classification on 3D data. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5648–5656.
- Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. <https://doi.org/10.48550/arXiv.1706.02413>
- Qi, Y., Liu, C., Scaioni, M., Li, Y., Qiao, Y., Ma, X., et al., 2024. Geometric information constraint 3D object detection from LiDAR point cloud for autonomous vehicles under adverse weather. *Transp Res Part C Emerg Technol*, **161**, 104555.
- Qian, R., Garg, D., Wang, Y., You, Y., Belongie, S., Hariharan, B., et al., 2020. End-to-end Pseudolidar for Image-based 3D Object Detection. <https://doi.org/10.48550/arXiv.2004.03080>
- Shi, S., Wang, X., Li, H., 2019. PointRCNN: 3D object proposal generation and detection from point cloud. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 770–779.
- Song, X., Dai, S., Lin, C., Zhan, S., Zhao, Y., 2021. LiDAR data enhancement via pseudo-LiDAR point cloud correction. *J Image Graph*, **26**, 1157–1168.
- Splietker, M., Behnke, S., 2021. Rendering and tracking the directional TSDF: Modeling surface orientation for coherent maps. In: 2021 European Conference on Mobile Robots (ECMR), 1–8.
- Sualeh, M., Kim, G. W., 2020. Visual-LiDAR based 3D object detection and tracking for embedded systems. *IEEE Access*, **8**, 156285–156298.
- Vora, S., Lang, A. H., Helou, B., Beijbom, O., 2020. PointPainting: Sequential fusion for 3D object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4603–4611.
- Wang, W., Zhang, W., 2024. CRRFNet: An adaptive traffic object detection method based on camera and radar radio frequency fusion. *Transp Res Part C Emerg Technol*, **166**, 104791.
- Wang, Y., Chao, W. L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K. Q., 2019. Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8437–8445.
- Yang, B., Luo, W., Urtasun, R., 2018. PIXOR: Real-time 3D object detection from point clouds. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7652–7660.
- Yang, Y., Zhan, J., Liu, Y., Qu, X., 2025. Policy-aware cross-city learning: A framework for sustainable urban governance. *Innovation*, **6**, 100986.
- You, Y., Wang, Y., Chao, W. L., Garg, D., Pleiss, G., Hariharan, B., et al., 2019. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. <https://doi.org/10.48550/arXiv.1906.06310>
- Zhao, Y., Chen, X., Liu, P., Nielsen, C. P., McElroy, M. B., 2025. Future ultrafast charging stations for electric vehicles in China: Charging patterns, grid impacts and solutions, and upgrade costs. *Engineering*, **48**, 309–322.
- Zheng, S., Li, W., Hu, J., 2019. Vehicle detection in the traffic environment based on the fusion of laser point cloud and image information. *Chin J Sci Instrum*, **40**, 143–151.
- Zhou, Y., Tuzel, O., 2018. VoxelNet: End-to-end learning for point cloud based 3D object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4490–4499.



Xinpeng Yao received the M.S. degree in control theory and control engineering from Liaoning Technical University, in 2008. Since 2020, he has been the Deputy Director of Information Technology Research Center of Shandong High-speed Group Innovation Research Institute.



Ruini Zhang is currently pursuing the M.S. degree in computer science and technology at Chang'an University. Her main research interests are intelligent transportation systems and accelerated testing for connected and autonomous vehicles.



Yunchao Li received the M.S. degree in computer science and technology from Chang'an University in 2022. His main research interests are target detection and intelligent transportation systems.



Wen Rong received the M.S. degree in signal & information processing from Shandong University in 2010. From 2022 to now, he stayed in Shandong High-speed Group Innovation Research Institute, as a research fellow.



Han Zhang is a Ph.D. candidate at Shandong University and Research Specialist at Shandong High-speed Group Innovation Research Institute, focusing on intelligent transportation systems. He has published over 20 papers and holds 15 patents in V2X communication and traffic perception technologies, contributing to major national research projects to enhance infrastructure efficiency and safety.



Zijian Wang received the M.S. degree in electronic and computer engineering from Portland State University, in 2012. He is the Vice President of Shandong High-speed Group Innovation Research Institute. His research interests include traffic information engineering and control. He is mainly engaged in highway informatization, traffic economy, and industry.