Applied Intelligence (2023) 53:22753–22774
https://doi.org/10.1007/s10489-023-04630-4

3D object detection algorithm based on multi-sensor segmental fusion of frustum association for autonomous driving

Chongben Tao[1,2,3] · Weitao Bian[1] · Chen Wang[1] · Huayi Li[1] · Zhen Gao[4] · Zufeng Zhang[5] · Sifa Zheng[5] · Yuan Zhu[6]

Accepted: 10 April 2023 / Published online: 2 July 2023

Abstract
The rotation characteristics of point clouds are challenging to capture in current multimodal fusion methods for 3D object detection. A single fusion method cannot well balance the accuracy and speed in object detection. Therefore, a multi-sensor segmental fusion of frustum is proposed for 3D object detection in autonomous driving. A monocular camera, lidar, and radar are used for piecewise distributed feature-level fusion through frustum association. Firstly, a fully convolutional network is used to obtain a 2D detection frame and a center point of an object from an image. Frustum is generated according to the depth and scale information in a 3D space. Secondly, region of interest in the lidar and radar point clouds is determined by using the frustum association method. Then, spherical voxelization and spherical voxel convolution are performed on the lidar point cloud while extracting the rotation-invariant feature. Finally, feature-level fusion is performed with object attributes extracted from an image and the radar point cloud to improve the detection results. Meanwhile, a dynamic adaptive neural network of parameters for feature fusion is proposed, and it quickly obtains fusion features and ensures the accuracy of fusion results. The proposed method is both compared with other algorithms on the nuScenes dataset and tested on a severe weather dataset Radiate and in a real scenario. The proposed method has achieved the highest NDS score and the highest average accuracy in severe weather compared with other advanced methods. The experimental results indicate that the proposed method has higher accuracy and more excellent adaptability in various complex and severe weather driving environments.

Keywords 3D object detection · Autonomous driving · Multi-sensor fusion · Frustum association

1 Introduction

Autonomous driving is often equipped with different types of sensors to handle various complex driving environments and to improve system robustness and accuracy. With the rapid development of various sensors, an increasing number of sensor fusion algorithms have emerged, and among them, 3D object detection has the most significant development tendency [1–5]. Most fusion algorithms currently focus on 3D object detection fused by a camera and lidar, which have achieved excellent results, such as multi-view 3D object detection (MV3D) [1] and Frustum PointNets [3]. The advantages of each sensor, particularly in complex environments, can be fully utilized and combined by fusion algorithms. A monocular camera can provide rich RGB information, which can be quickly processed and analyzed by algorithms. Therefore, the detection result can be obtained quickly. However, depth information and the shape contour of an object are difficult to obtain. Lidar has very high accuracy at short distances, which copes with complex environments, including many vehicles, pedestrians, and buildings. However, it loses its detection ability in long-distance detection because of too sparse point clouds as the effective distance is within 100m. The combination of these two types of sensors is sufficient for most normal environments. However, both a camera and lidar, which do not have long-range detection capabilities, are subject to strong interference in bad weather conditions. Radar distinguishes objects by emitting millimeter waves, which has the characteristics of all weather conditions, all day, and all night. Even during bad weather conditions, radar still has a good detection capability, and its detection distance is two or three times longer than that of lidar. Furthermore, the radar point cloud is sparse, which is faster to process than the lidar point cloud. Therefore, radar

✉ Chen Wang
chenwang@usts.edu.cn

Extended author information available on the last page of the article

is fused with a camera and lidar for 3D object detection in this study.

Generally, object rotation information is crucial for driving scenarios. The rotation information of an object is often not obtained using conventional algorithms for anchor-based object detection. Moreover, the rotation invariance and scale invariance of point clouds are not considered by the point cloud model. In PointNet [6], T-net is used to the learn rotation features of point clouds. However, if no data enhancement occurs, the effect of the model remains greatly affected by simply rotating an object. Meanwhile, a conventional point cloud network [7–9], based on common coordinate systems, is limited by the disorder of point clouds as the rotation characteristics of point clouds are difficult to capture. In recent years, center-based methods have been widely used in object detection for their ability of to adapt well to rotational models without complex post-processing. For multi-sensor fusion methods, image detection results of 2D methods are often projected into a 3D point cloud to form a 3D region of interest (ROI) space. Another method is by using the ROI generated from a 2D image to limit the search space of point clouds, which can significantly reduce the amount of computation. This study also adopted this technique to generate a frustum through 2D detection boxes and furthermore combine depth information to determine the ROI in point clouds.

Multi-modality fusion for autonomous driving includes pixel, feature, and decision-level fusion. Most studies are presently focused on pixel-level fusion and have achieved remarkable results and reached the application-level. However, object detection of pixel-level fusion has a poor real-time performance, which is limited by different types of sensor fusion. Meanwhile, a large amount of information is lost in decision-level fusion, and its recognition ability is poor. Object features are extracted from the information of each sensor through feature-level fusion, and feature quantity is obtained using a fusion algorithm to detect objects. Feature-level fusion not only maintains a sufficient amount of valid information and removes redundant information from the object but also improves the detection accuracy. Therefore, a method originated from feature-level fusion is proposed, that is, segmentally fusing features from the three types of sensors. The ROI spaces in the lidar and radar point clouds are determined by a frustum association method using the detection frame of a monocular camera at a close distance. For the detection by the lidar point cloud, a method of spherical voxelization of the point cloud is proposed based on the core concept of a center-based method. A rotation-invariant feature is extracted by spherical voxel convolution and trilinear interpolation, and then the rotation direction of the object is further determined. Moreover, a dynamic adaptive neural network of parameters is used to perform feature-level late fusion, where a fusion feature is used to improve the detection results and supplement object attributes. To predict the position and direction of pedestrians, a monocular camera and radar are used at a long distance so that the autonomous driving system can perform path planning and provide advance warning.

The following are the innovations of this paper:

- Based on the core concept of a center-based method, the center point of an object is detected, and an object model is constructed. Further, the irregular point cloud is spherically voxelized with the center point as the center of sphere. Spherical voxel convolution and trilinear interpolation are used to extract the rotation-invariant feature of points and can obtain the rotation information of an object.
- Based on the different characteristics of the three types of sensors, a segmental distributed feature-level fusion is adopted. Meanwhile, a frustum association method is used to correspond lidar and radar point clouds to the frustum of the ROI, which generated by visual inspection. Furthermore, according to depth, scale, and other types of information, the scope of the ROI is further reduced to remove all irrelevant points outside the scope. Thus, the detection speed and accuracy are improved.
- A dynamic adaptive neural network of parameters is proposed for feature fusion, which solves the divergence problem of fusion networks and improves the operation efficiency of the proposed algorithm. The detection results are improved by the optimized fusion feature, and the robustness of the proposed algorithm is improved as well.

2 Related work

Generally, 3D object detection algorithms for multi-modal fusion can be divided into two categories: early and late fusion. Owing to the continuous development of fusion algorithms, all algorithms cannot be fully included in the above classification scheme [10, 11]. Therefore, multi-modal fusion methods for object detection can be divided into two categories: sequential and parallel fusion.

2.1 Methods based on sequential fusion

Sequential fusion means that latter stage relies on the processing results of previous stage, and multi-level features are used in sequence. Qi et al. proposed Frustum PointNets [3] for 3D object detection. The ROI was first extracted by a 2D detector, and then 2D coordinates were transformed into a 3D space to obtain region proposals from a frus-

tum. The frustum was segmented into blocks to obtain the points of interest for further regression. The method achieved good results by restricting search in a 3D space using a well-established 2D detection method. F-ConvNet [12] was proposed based on the above cascaded method, where the frustum sequence between near and far planes was generated by 2D region proposals. The point-by-point feature in the point cloud was converted into feature vectors at the frustum level and hence improved the running efficiency of the algorithm. CenterFusion [13] based on camera and lidar fusion was proposed based on the concept of a frustum. Preliminary detection results obtained from lidar data and images were associated. 3D bounding boxes of objects and 3D properties (e.g., depth, velocity, and rotation) were estimated by combining them with image features. Since radar was used for attribute regression, this method sacrificed, to some extent, object accuracy (e.g., geometric information of the object) for sufficient object attributes. Tao et al. proposed F-PVNet [14], which made full use of local sensitive points and contextual features by using a frustum to group local points and aggregate them with features obtained from submanifold voxel convolution. Tao et al. proposed a ground culling algorithm for 3D object detection, which reduced the amount of computation to a certain extent and accelerated the running speed [15]. It is an attempt to remove the irrelevant point cloud idea. Point cloud projection is also a sequential fusion method. Vora et al. proposed a method called PointPainting for 3D object detection [16]. 2D semantic segmentation was first performed through a semantic network, and then lidar points were projected into the segmentation mask according to a transformation matrix. Finally, a 3D detector was used for classification and localization. Semantic segmentation information was supplemented by this method with lidar detection, which continuously enhanced existing networks with segmentation scores [17, 18]. However, only depending on semantic segmentation itself, the fusion method has a counterproductive result due to the too low semantic segmentation accuracy. Pseudo-LiDAR [19] was proposed by Wang et al. as another attempt at sequential fusion. A pyramid stereo matching network [20] was first used to estimate depth information. Then each pixel in the image was back-projected into a 3D space, generating a Pseudo-LiDAR signal similar to the lidar point cloud. Finally, the existing lidar detector was used for detection. Based on Pseudo-LiDAR, You et al. proposed Pseudo-LiDAR++, which used very few real and accurate lidar points to correct depth estimation bias. Nakrani et al. attempted to solve the problem of not end-to-end in Pseudo-LiDAR [21]. The technique of changing the representation data was considered to address the problem of poor vision-based depth estimation, providing a valuable idea for image-only perception algorithms.

2.2 Methods based on parallel fusion

Parallel fusion indicates that each fusion stage is carried out simultaneously. Multiple modalities are first fused to obtain a representation result and then input to the network, or each modality is processed by its respective network and then fused. These methods often employ different approaches to integration; they have a wide variety, but lack uniform standards. MV3D [1] was proposed by Chen et al., which only used image and bird's eye view (BEV) of point clouds. The amount of computation was reduced while retaining enough information. The ROI was first extracted from the BEV, which was then projected into the image and the front view of the point cloud. After pooling and integrating into the same dimensional information, features were finally extracted and fused. However, this method has a drawback in small object detection. Unlike MV3D, Ku et al. proposed a 3D object detection method called AVOD [22], which performed fusion before the region proposal stage. Feature maps, including images and the BEV of point clouds, were generated through the FPN [23] network. A 3D anchor frame was used to select corresponding regions in both of them for fusion. The fusion result was finally input into the fully connected layer for 3D object detection. To avoid information loss caused by the point cloud projection, Xie et al. proposed PI-RCNN [24] by using a new fusion method to fuse 2D semantic segmentation into 3D region proposals. Pixel-level fusion is also a form of parallel fusion. Liang et al. proposed ContFuse [25], which used continuous convolution to fuse multi-sensor features at multiple scales through pixel-level fusion. ResNet-18 [26] was first used to extract features from the image and BEV of a point cloud and then adopted multi-scale feature fusion for the image. Then, PCCN [27] was used to fuse it into the BEV to achieve 3D object detection. Yoo et al. proposed a 3D-CVF [28] method.The point cloud was voxelized and then transformed into a 2D BEV feature map through sparse convolution. Then, ResNet-18 [26] was used to extract image feature and fused them into BEV feature maps. The problem of misplaced views was solved by using this method, but feature blur [16] inevitably led to bias due to pixel-level fusion. While a multitasking problem is often solved using different networks, MMF [29] used a single network to solve a multi-tasking problem, which achieved point-wise and ROI-wise feature fusion. EPNet [30], proposed by Huang et al., was a lidar-guided image fusion method. A point-by-point correspondence was directly established between the original point cloud data and image, and the importance of semantic information was estimated to enhance useful features and suppress interfering ones. 4D-Net [31], designed by Piergiovanni et al., combined image, lidar, and temporal

information. Moreover, motion cues were better used in the dynamic connection learning method. Although the performance of fusion algorithms has been greatly improved, a certain gap remained compared with algorithms only based on lidar [32, 33]. Additionally, Cao et al. proposed an accelerated point-voxel representation [34], which fused the features of points and voxels into a single 3D representation. Wang et al. proposed BrT [35], which unified multimodal data from different sources by transformer and achieved seamless fusion of point clouds with multi-view images using an aggregated form of point-to-patch projection. Inspired by the application of transformer in 2D vision tasks, Gao et al. proposed LFT-Net [36] to solve the local feature extraction problem in point cloud segmentation tasks by associating local features with point clouds through a local feature transformer network.

Overall, both types of above methods have advantages and disadvantages. The sequential fusion method is similar to cascade, and the processing results of previous stage directly affect the effectiveness of the latter stage. No direct and strong correlation exists between the stages of the parallel fusion method, but the misalignment of views between sensors is a problem that should be solved by each algorithm. Therefore, combining the advantages of these two types of methods, a multi-sensor segmentation fusion method based on a frustum is proposed, and sensor fusion is performed in two stages. The first stage is based on sequential fusion, where 2D region proposals in an image, preliminary 3D detection boxes, and center points are obtained by CenterNet [37]. A frustum is generated to determine the ROI in the lidar and radar point clouds, filtering out invalid information in the point cloud. In the second stage, the concept of parallel fusion is used, where three types of sensor data are extracted through their respective networks to perform feature fusion. Feature fusion is used to reduce the inaccuracy of initial detection results. The first stage improves the efficiency of feature extraction in the second stage, whereas the second stage reduces the errors caused by the cascade in the first stage.

3 Algorithm of multi-sensor segmental fusion of frustum association

In this paper, a segmental fusion association algorithm is proposed based on three sensors: camera, lidar, and radar for 3D object detection. The detection range of the point cloud is narrowed by using a frustum association method. Then, the lidar point cloud is spherically voxelized. Furthermore, the rotation-invariant feature is extracted through a spherical voxel convolution. Meanwhile, a neural network with dynamic parameter adaptation is used to perform feature-level fusion for the improvement of the detection results. Figure 1 shows the framework of the proposed method.

Firstly, the fully convolutional network is used to obtain the 2D detection frame and center point of objects in the image. At a short distance, ROI in the lidar and radar point clouds which extended to the pillar are determined by the frustum association method. Thereafter, spherical voxelization and spherical voxel convolution are performed on the lidar point cloud to extract the rotation-invariant feature. Finally, feature-level fusion is performed with object attributes, which are extracted from the image and radar point cloud to improve the detection results and generate a feature map. Furthermore, lidar point clouds that are too sparse are not used at a long distance.

3.1 Generation of detection boxes and center points

CenterNet is used to generate detection boxes and center points for frustum association, while object-related properties, such as scale, depth, and 3D position, are regressed. As a representative among anchor-free series of algorithms, CenterNet represents an object as a center point when creating a model, which addresses some problems of anchor-based methods [38]. The CenterNet network uses $I \in R^{W \times H \times 3}$ as input image, W and H are the width and height of the image, respectively. Then, a keypoint heatmap is generated as follows:

$$\hat{Y} \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times C}, \tag{1}$$

where R is the output stride and C is the number of object categories. The detected object of class c is output as $\hat{Y}_{x,y,c} = 1$, and its center point is (x, y). The output of the area where no object is detected is represented by $\hat{Y}_{x,y,c} = 0$.

Each keypoint of the ground truth at position $r \in R^2$ is equivalently replaced with the corresponding keypoint $\tilde{r} = \left[\frac{r}{R}\right]$ on the downsampled low-resolution image. The keypoints of the ground truth are passed through a Gaussian kernel function to scatter onto the heatmap of the ground truth.

$$Y_{xyc} = \exp\left(-\frac{(x - \tilde{r}_x)^2 + (y - \tilde{r}_y)^2}{2\sigma_r^2}\right), \tag{2}$$

where σ_r is the standard deviation of object size adaptation and $Y_{xyc} \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times C}$. If two overlapping Gaussian functions exists for the same class c, the element-wise largest one is selected.

Object information, including depth, dimension, and orientation, is regressed from the detected center points to generate 3D detection boxes through CenterNet. The depth

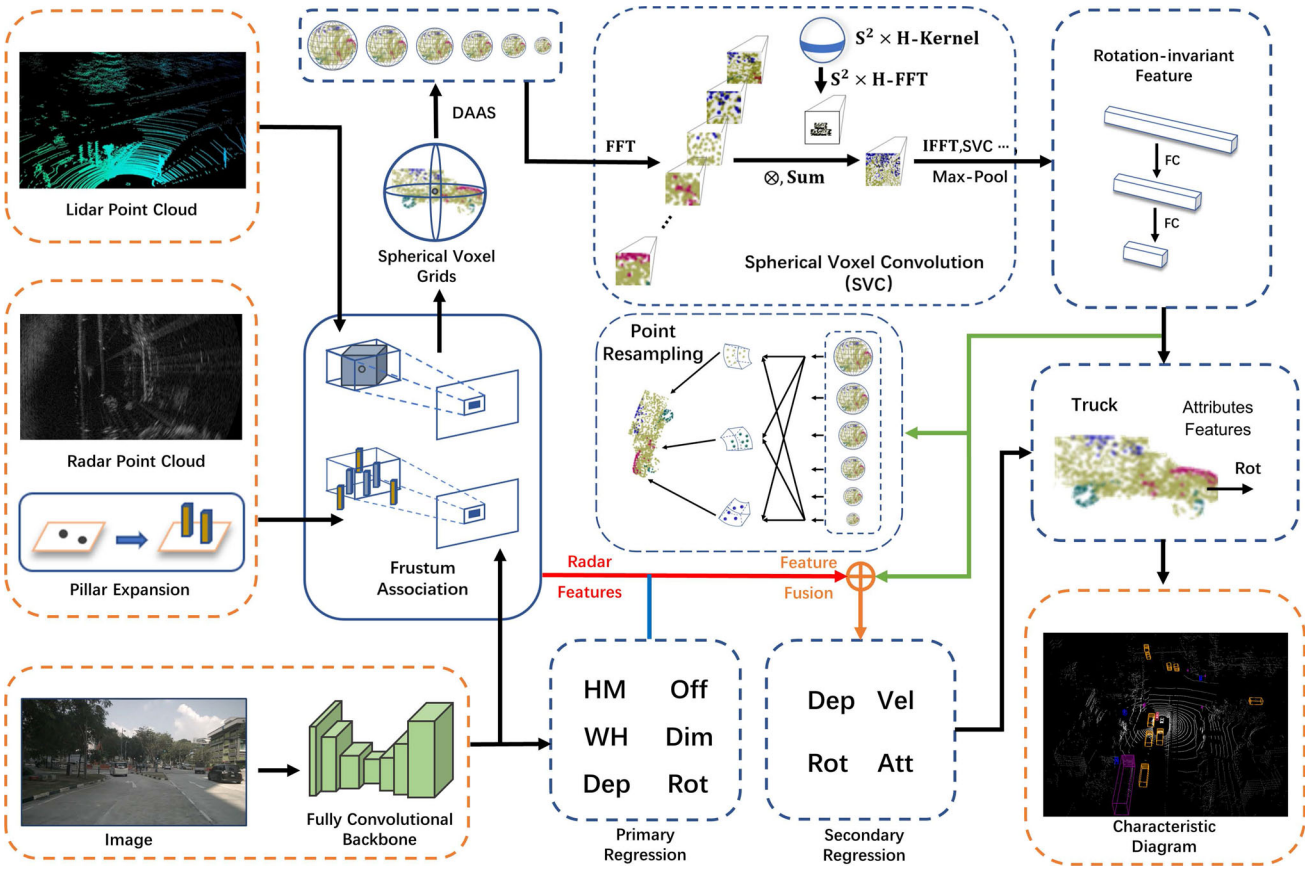


Fig. 1 Network frame diagram of the proposed method (HM, heat map; Off, offset; WH, width and height; Dim, dimension; Dep, depth; Rot, rotation; Vel, velocity; Att, attributes)

is computed and output as an additional channel $\widehat{D} \in [0,1]^{\frac{W}{R}\times\frac{H}{R}}$. The dimension contains three scalars, which are directly regressed to their absolute value via $\hat{\Gamma} \in [0,1]^{\frac{W}{R}\times\frac{H}{R}\times 3}$. Orientation is represented as two bins, and each bin contains four scalars for encoding. To avoid discretization errors in the network due to output strides, a local drift is also computed for each center point.

The training objective function is defined as follows:

$$L_k = \frac{1}{N}\sum_{xyc}\begin{cases}\left(1-\hat{Y}_{xyc}\right)^{\alpha}\log\hat{Y}_{xyc} & Y_{xyc}=1\\ \left(1-Y_{xyc}\right)^{\beta}\left(\hat{Y}_{xyc}\right)^{\alpha}\log\left(1-\hat{Y}_{xyc}\right) & \text{otherwise}\end{cases}, \tag{3}$$

where N is the number of targets and α and β are the hyper-parameters of focal loss [39].

3.2 Frustum association

The precise 2D detection frame, rough 3D detection frame of each object in a scene, and center point of the object can be obtained through CenterNet. To fully use radar information and reduce irrelevant calculations, this paper proposes a result-level fusion method of frustum association.

A ROI frustum is created for the object by using the 2D detection frame, which is obtained from the image as well as the depth and size of the estimated object. As shown in Fig. 2, all irrelevant point cloud data outside the view frustum can be filtered out by mapping the frustum to the point cloud, which effectively reduces the computational load of subsequent point cloud detection. Meanwhile, the proposed method solves the object overlapping problem in 2D image detection. As objects are separated in a 3D point cloud, separated ROI frustums can be created for 2D overlapping objects to more accurately detect the overlapping objects in the segmented 2D image.

Unlike the lidar point cloud, the radar point cloud has the problem of inaccurate Z dimension or no Z dimension at all, resulting in inaccurate height information of the object. Therefore, a preprocess method for pillar expansion of the radar point cloud is proposed. Each radar detection point is expanded into a fixed pillar, which is associated with the Z dimension in a 3D space. A portion of radar detection is considered within the ROI if the corresponding strut is located fully or partially within the ROI frustum.

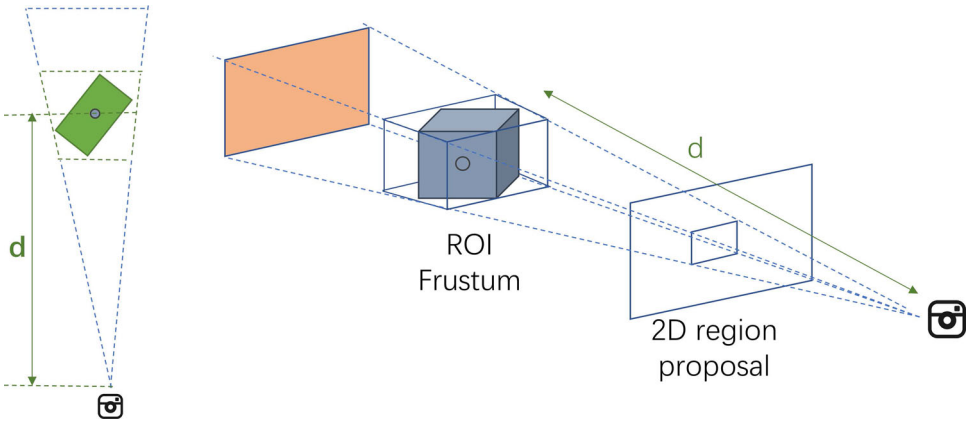


Fig. 2 Schematic BEV (left). Schematic diagram of frustum generation (right)

Different types of sensors are not synchronized temporally and spatially when acquiring data. Due to their different acquisition cycles and perspectives, aligning the camera and lidar, as well as the camera and radar, is essential before correlation, and then the fusion can be performed.

3.2.1 Camera and lidar calibration

Feature alignment of the calibration plate in this study is performed to estimate the parameters in the calibration of the camera and lidar [40]. The plane normal of the calibration plate is defined as n_L, rotation matrix as R_C^L, camera normal matrix as $N_C = \left[n_C^0, n_C^1, n_C^2\right]^T$, and lidar normal matrix as $N_L = \left[n_L^0, n_L^1, n_L^2\right]^T$. The centroid of the calibration plate and the plane normal n_L are firstly extracted [41]. The rotation matrix R_C^L can be aligned with the lidar normal matrix N_L by rotating the camera normal matrix N_C using the following equation:

$$R_C^L N_C = N_L, \tag{4}$$

where N_C and N_L are known quantities; hence, R_C^L can be found.

Although existing methods often include more samples in the computation to improve their robustness [42], they also tend to over-tune the calibration plate. Therefore, three bit-pose sets are selected to fully constrain Eq. (4). Furthermore, N_C and N_L are a formed square matrix for the analysis [40].

During solving R_C^L in Eq. (4), N_C^{-1} should be calculated. Therefore, the linear correlation is identified in the normal matrix, which is important for the accuracy of the calibration results. Moreover, lidar is subject to measurement errors. Matrix condition numbers are used to evaluate the linear correlation and quality of the rotation parameters. Meanwhile, errors in calibration plate measurements are used to evaluate the translation parameters. The variability of quality (VOQ) is defined as

$$VOQ = \kappa_{LC} + e_{be}, \tag{5}$$

where κ_{LC} denotes the most unstable result in the inverse matrix of N_L and N_C and e_{be} denotes the average plate error of the three bit poses. Thus, a lower VOQ score indicates better alignment. Figure 3 shows the actual calibration.

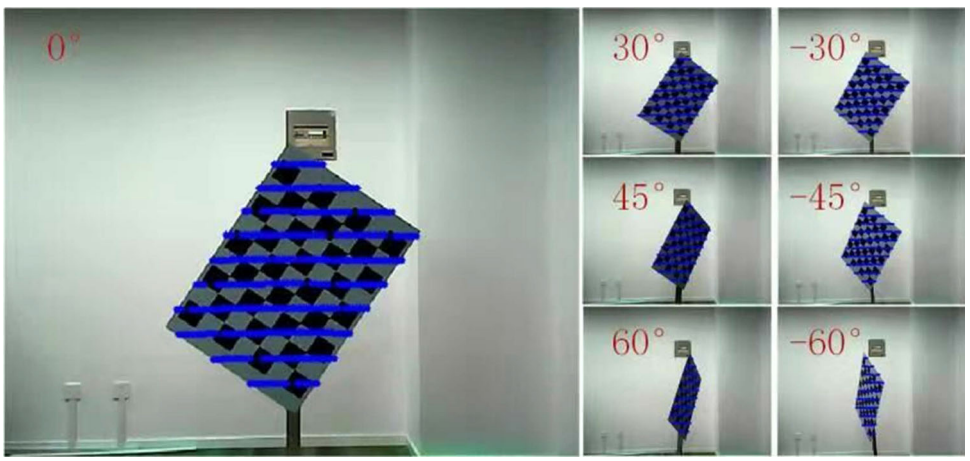


Fig. 3 Actual calibration results

3.2.2 Camera and radar calibration

The calibration of the camera and radar is relatively easy. Since the radar only obtains the X and Y-axis coordinates of the object, the coordinate system conversion between them is a conversion of a 2D X − Y coordinate system.

The camera coordinate system is defined as $O_C = [x_C, y_C, 1]^T$, the radar coordinate system as $O_R = [x_R, y_R, 1]^T$, and the rotation angle as θ. Then, the conversion relationship of the coordinate system is expressed as follows:

$$\begin{bmatrix} x_C \\ y_C \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_t \\ \sin\theta & \cos\theta & y_t \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_R \\ y_R \\ 1 \end{bmatrix}, \tag{6}$$

where x_t and y_t are the translations in the X and Y-axis directions(see Fig. 4).

3.3 Spherical voxelization of the lidar point cloud

Vehicles in driving scenes often have a certain rotation relative to their driving direction. Considering the influence of multiple factors (e.g. road slope and curves), this rotation can be arbitrary. Control of the object direction is also a key factor in predicting the movement of the objects and preventing collisions.

After the interest part of the lidar point cloud and its center point through frustum association are obtained, spherical voxelization and spherical voxel convolution [43] are used to classify the objects. Then, the rotation-invariant feature and object rotations information are extracted. To convert the point cloud into spherical voxels with a Euclidean structure, a density-aware adaptive sampling (DAAS) method is proposed, rather than uniform sampling, to solve the problem of sparse points around the poles and dense points around the equator (Fig. 5). This problem can lead to bias in the spherical signal; that is, the point cloud is not uniformly distributed, resulting in a failure to align the feature extracted from the spherical voxel convolution with the original point cloud feature. The DAAS method samples the point cloud at both poles using a wider filter to adjust for density differences, and the exacted implementation is given in Eq. (7).

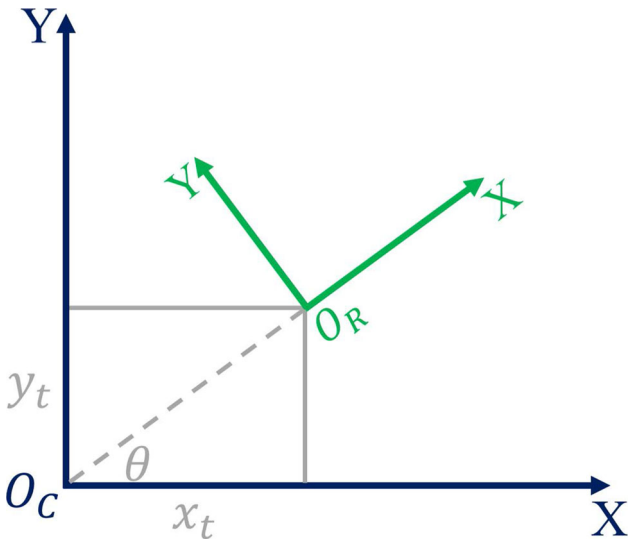


Fig. 4 Transformation diagram of the coordinate system

A unit sphere is defined as a set of points $x \in \mathbb{R}^3$, which is normalized to norm. The spherical voxel space is defined as $S^2 \times H$. Moreover, points in the space are described by (α, β, h), where $(\alpha, \beta) \in S^2$; α and β are the polar and azimuth angles, respectively; and h is the straight-line distance from the point to the center of the sphere. The position of a spherical voxel is determined by its center (a_i, b_j, c_k), where $(i, j, k) \in I \times J \times K$. $I \times J \times K$ is the spatial resolution, called bandwidth. The coordinate of the nth point in $S^2 \times H$ is (α_n, β_n, h_n). Furthermore, a total of N points exists. The calculation formula of spherical signal $f : S^2 \times H \to \mathbb{R}$ is expressed as

$$f\left(a_i, b_j, c_k\right) = \frac{\sum_{n=1}^{N} \omega_n \cdot (\delta - \|h_n - c_k\|)}{\sum_{n=1}^{N} \omega_n}, \tag{7}$$

where ω_n is the normalization factor, which is defined as

$$\omega_n = 1\left(\|\alpha_n - a_i\| < \delta\right) \cdot 1\left(\left\|\beta_n - b_j\right\| < \eta\delta\right) \cdot 1\left(\|h_n - c_k\| < \delta\right), \tag{8}$$

where η is the density-aware sampling factor, δ is the predefined filter width, and $\eta = \sin(\beta)$ is used to control f to adaptively sample point set under non-uniform density. Equation (9) is used to express information along the H-axis orthogonal to S^2, which remains unchanged under random rotations.

$$(\delta - \|h_n - c_k\|) \in [0, \delta]. \tag{9}$$

3.4 Spherical voxel convolution

Spherical signal $S^2 \times H$ serves as input of the spherical voxel convolution. The rotation group $SO(3)$ is a special orthogonal group, which can be transformed into the ZYZ-Euler angle (α, β, γ), where $\alpha \in [0, 2\pi]$, $\beta \in [0, \pi]$ and $\gamma \in [0, 2\pi]$. To conveniently extract the rotation-invariant feature, the rotation operator L_R of the spherical voxel signal is defined as

$$[L_R f](s) = f\left(R^{-1} s\right), \tag{10}$$

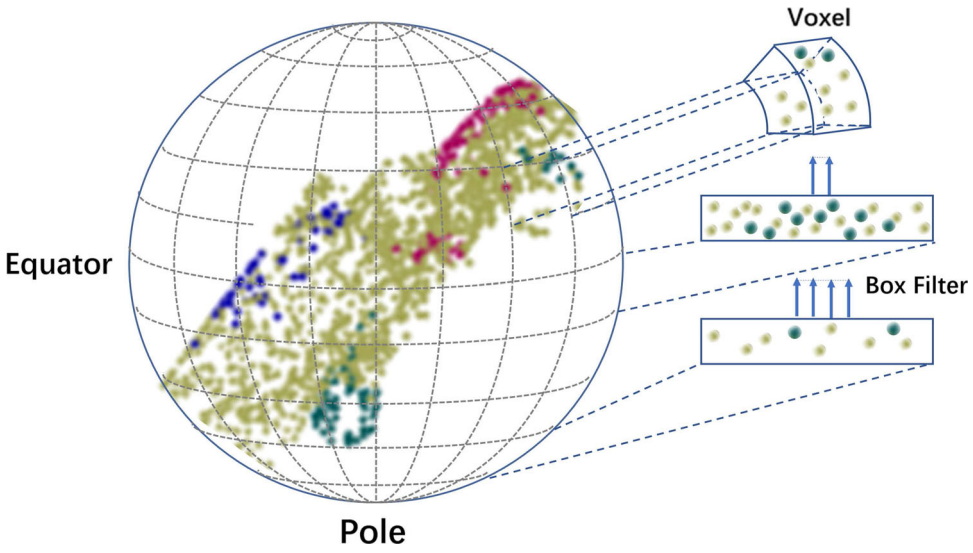


Fig. 5 Spherical voxelization of the lidar point cloud. The closer to the poles, the wider the filter, a technique to adjust for the differences in sampling density

where $R \in SO(3), s \in S^2 \times H, f : S^2 \times H \to \mathbb{R}$. The rotation only affects the spherical coordinate S^2, which has no effect on the H domain (see Fig. 6).

Equation (8) is the calculation formula for the convolution of two spherical signals

$$[\psi * f](p) = \langle L_P \psi, f \rangle = \int_{S^2 \times H} \psi \left(P^{-1} s \right) f(s) ds, \tag{11}$$

where, ψ represents the filter, $f : S^2 \times H \to \mathbb{R}$, $p \in S^2 \times H$, and P represents the element corresponding to p in $SO(3)$.

To prove the rotation invariance, the point cloud is assumed to be rotated by an arbitrary rotation matrix R. Then, for $\forall p \in S^2 \times H$, $p \to Rp$ is given. Since f is sampled from the input point cloud, the spherical signal rotation is represented by $f \to L_R f$. Subsequently, the spherical voxel convolution was applied to the rotated input signal

$$\begin{aligned} [\psi * L_R f](Rp) &= \langle L_{RP} \psi, L_R f \rangle \\ &= \int_{S^2 \times H} \psi \left(P^{-1} R^{-1} s \right) f \left(R^{-1} s \right) ds \\ &= \int_{S^2 \times H} \psi \left(P^{-1} s \right) f(s) ds \\ &= [\psi * f](p), \end{aligned} \tag{12}$$

where the simplification of the second-third step is from [43]. Therefore, the output of the spherical voxel convolution is not affected by the rotation. The rotation information of the object can be obtained by solving the transformation relationship between the corresponding feature (e.g., those of the tire and front of the car) and their counterparts in the standard orientation in the world coordinate system.

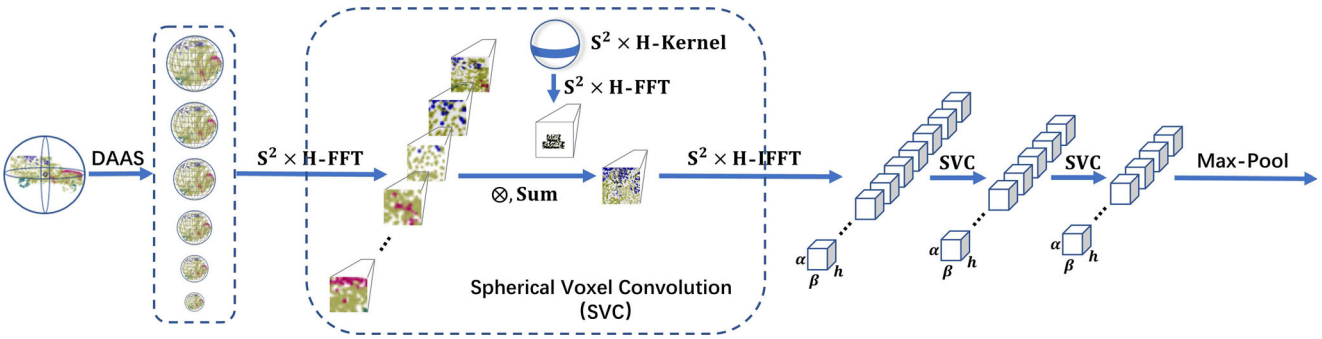


Fig. 6 Spherical voxel convolution

The spherical voxel convolution firstly converts the input and filter to a frequency spectrum via fast Fourier transform (FFT). Secondly, they are multiplied and converted to the spatial domain via inverse FFT (IFFT) [44]. Thirdly, point resampling is performed; that is, the feature is resampled at the original point positions. Trilinear interpolation is used as operator $\Lambda : \mathbb{R}^{S^2 \times H \times C} \rightarrow \mathbb{R}^{N \times C}$. The feature of each point is the weighted average values of eight nearest voxels from each point. The weight is inversely proportional to the distance between the point and each spherical voxel. A point-wise feature was obtained through fully connected layers.

3.5 Feature extraction and feature-level fusion

In practical applications, each single sensor can independently perform object detection and attribute regression. However, the advantage of the sensor often cannot be fully exerted. Three different sensors were used in this study to extract object features. An object feature was extracted through a fully convolutional network for predicting the center of the object and generating a bounding box. The extracted features include 2D size, 3D size, depth, rotation, and center offset of the object, which is used as the primary regression. The lidar point cloud extracts rotation-invariant features through a spherical voxel convolution, which is used to predict the true 3D rotation direction of an object. Furthermore, a detection frame was generated, which fits the rotation direction of the object. Radar detection can directly obtain the depth information of the object. Meanwhile, the object's speed information can be extracted according to the Doppler effect. Eigenvalues in the eigenvectors were extracted by the three sensors through the network, which has constraint and complementary relations. According to these relations, a dynamic adaptive neural network of parameters was proposed to perform feature fusion on eigenvectors.

A late fusion method with distributed feature-level was adopted in this study. As shown in Fig. 7, a feature was extracted from three types of sensor data. Moreover, the dynamic adaptive neural network of parameters was proposed for feature fusion.

The network comprises an input layer, a hidden layer with Gauss function neurons, and an output layer with linear neurons. M and Q denote the number of neurons on the hidden and output layers, respectively. The input mode is $X, X = [x_1, x_2, \ldots, x_R]^T$, and the output is $Y, Y = [y_1, y_2, \ldots, y_Q]^T$. The output of the hidden unit is expressed as

$$Z_j = \exp\left(-\left\|\frac{X - C_j}{\sigma_j}\right\|\right), \tag{13}$$

where Z_j is the output value of the jth neuron in the hidden layer, $j = 1, 2, \ldots, M$. Further, $C_j = [C_{j1}, C_{j2}, \ldots, C_{jR}]^T$ is the center of the jth neuron in the hidden layer, composed of the center components of all neurons in the output layer that corresponds to the neuron. σ_j is the width of the jth neuron in the hidden layer that corresponds to C_j.

The following expression presents the relationship between the input and output of the neurons in the output layer:

$$y_k = \sum_{j=1}^{M} w_{kj} Z_j, \tag{14}$$

where y_k is the output value of the jth neuron in the output layer, $k = 1, 2, \ldots, Q$, and w_{kj} is the weight between the kth and jth neurons in the output and hidden layers. Given that the neurons are enough in the hidden layer, the network can approximate any functions with any desired accuracy. Parameters such as the center and width of the neurons in the hidden layer and the weights of the output layer determine the network performance. If these parameters cannot be accurately determined, then network divergence may occur [45].

In this paper, a dynamic adaptive particle swarm optimization was proposed to optimize three parameters: center, base width, and weight vectors. The proposed algorithm was firstly initialized as a group of random particles. Then, the optimal solution was obtained through iteration. The particle updates itself by tracking two extrema: $pbest$ and $gbest$ [46]. After obtaining these two optimal values, the speed and position are updated by following Eq. (15)

$$\begin{cases} v_i = \omega v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i) \\ x_i(t+1) = x_i(t) + v_i(t) \end{cases}, \tag{15}$$

where $i = 1, 2, \ldots, N$, N is the total number of particles, ω is the inertia weight, v_i is the particle velocity, $rand()$ is a random number between (0, 1), x_i is the current position of the particle, and c_1 and c_2 are the learning factors.

To increase its global convergence ability and avoid falling into local optimum in the early stage, the learning factor and inertia weight were dynamically and adaptively changing in this study. The initial value of learning factor c_1 is denoted by c_{1m}, which is reduced to c'_{1m} in a nonlinear manner during iteration. The initial value of c_2 is recorded as c_{2m}, which is increased to c'_{2m} in a non-linear manner. The initial value of ω is recorded as ω_m, which is reduced to ω'_m in a non-linear manner. The following is the relevant formula:

$$\begin{cases} c_1(k) = c'_{1m} + \left(\frac{m-i}{m}\right)^{\alpha} (c_{1m} - c'_{1m}) \\ c_2(k) = c'_{2m} + \left(\frac{m-i}{m}\right)^{\beta} (c_{2m} - c'_{2m}) \\ \omega(k) = \omega'_m + \left(\frac{m-i}{m}\right)^{\beta} (\omega_m - \omega'_m) \end{cases}, \tag{16}$$

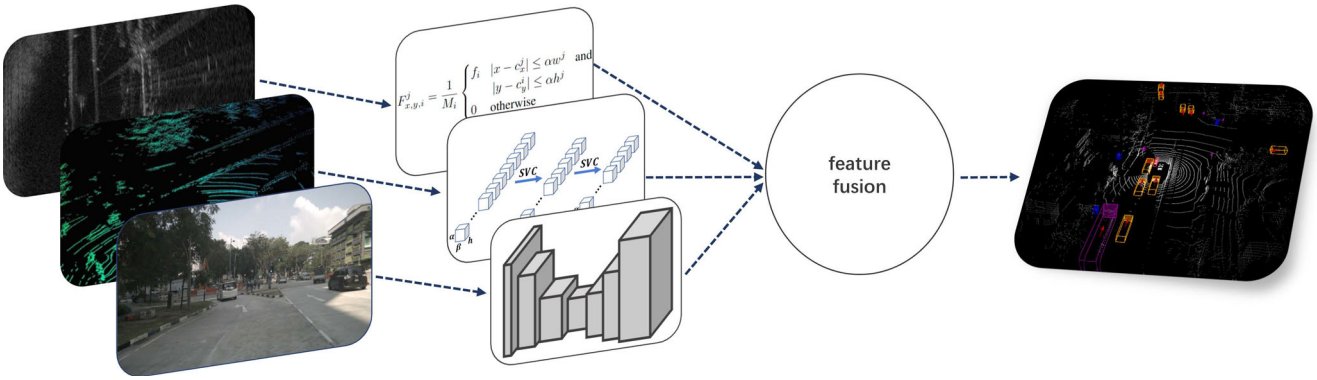


Fig. 7 Schematic diagram of the feature-level late fusion

where m is the maximum number of iterations, i is the current number of iterations, and $\alpha, \beta \in \{0.5, 1, 1.5, 2.0\}$.

Meanwhile, the mutation in the later stage of evolution divides the entire population into two parts: one part still follows the original update formula, whereas the position updated formula of the other part is changed to Eq. (17). Part of the particles move in the opposite direction of $gbest$, increasing group diversity and avoiding falling into local optimal solution.

$$x_i(t+1) = x_i(t) - v_i(t). \tag{17}$$

The values of the three parameters (i.e. center, base width, and weight vectors) were encoded as the particle parameters. Moreover, the fitness was calculated for each particle according to the fitness function defined by normalized root mean square error(NRMSE), as follows:

$$f = NRMSE = \sqrt{\frac{\sum_{k=1}^{N}(y(k) - y_m(k))^2}{N\sum_{i=1}^{N} y^2(k)}}. \tag{18}$$

Then, f is compared with the fitness of $pbest_i$ and that of $gbest$, and the relevant parameters are updated. When optimization reaches the maximum number of iterations or the ideal approximation accuracy, three parameters of the optimized neural network are output.

3.6 Loss function

The following is the definition of the overall multi-task loss function of the proposed algorithm:

$$L = \lambda_{rc} L_{rc} + \lambda_{ar} L_{ar} + \lambda_{fo} L_{fo}, \tag{19}$$

where λ represents adjustment coefficient; rc, the bounding box and center point extraction; ar, the attribute regression; and fo , the optimization module of feature fusion. The adjustment factor λ controls the weight of each task, which determines whether the model has excellent performance and training efficiency. A dynamic weight average approach was used in this study to dynamically adjust the weights of each loss [47]. Here, $w_k(\cdot)$ denotes the relative rate of loss decline in task k, i.e., the ratio of the current loss to the previous loss

$$w_k(t-1) = \frac{L_k(t-1)}{L_k(t-2)}, \tag{20}$$

where $L_k(\cdot)$ denotes the current loss in task k. The larger the ratio, the harder the current task to train, and a larger weight must be assigned. The weight λ_k of task k was updated as follows:

$$\lambda_k(t) = \frac{K \exp\left(\frac{w_k(t-1)}{T}\right)}{\sum_i \exp\left(\frac{w_k(t-1)}{T}\right)}, \tag{21}$$

where $K = \sum_i \lambda_i(t)$ ensures that all weights are active within a range and T is the modulation coefficient of the task distribution. That is, the larger the task distribution, the more uniform the task distribution. Initialization should be set up with consistent weights for each task. Further, a priori unbalanced initialization can be introduced according to an actual scenario.

The loss L_{rc} of the bounding box and center point extraction module is defined as

$$L_{rc} = L_k + \gamma_s \frac{1}{N}\sum_{k=1}^{N}|\hat{S}_k - s_k| + \gamma_o \frac{1}{N}\sum_{r}|\hat{O}_{\tilde{r}} - \left(\frac{r}{R} - \tilde{r}\right)|, \tag{22}$$

where, γ is the adjustment factor, N is the number of objects, $\hat{S}$ is the single size prediction, s is the object size, and $\hat{O}$ is the local offset.

To increase the robustness of the attribute regression and optimization modules of feature fusion, Huber loss is uni-

Table 1 Comparison of different object detection algorithms on nuScenes dataset

Method	Modality			NDS↑	mAP↑	Error↓				
	C	R	L			mATE	mASE	mAOE	mAVE	mAAE
InfoFocus [50]			√	0.395	**0.395**	0.363	0.265	1.132	1.000	0.395
MonoDIS [52]	√			0.384	0.304	0.738	0.263	0.546	1.533	0.134
CenterNet [37]	√			0.400	0.338	0.658	**0.255**	0.629	1.629	0.142
PointPillars [51]			√	0.454	0.305	0.731	0.314	0.748	1.497	0.201
CenterFusion [13]	√	√		0.449	0.326	0.631	0.261	0.516	0.614	0.115
DWD-Fusion [53]	√	√	√	0.461	0.331	0.496	0.270	0.495	0.599	0.142
Ours	√	√	√	**0.485**	0.357	**0.334**	0.259	**0.463**	**0.588**	**0.104**

formly used, which is defined as

$$L_* = \begin{cases} \frac{1}{2}(\Delta P)^2 & |\Delta P| \le \xi \\ \xi|\Delta P|^2 - \frac{1}{2}\xi^2 & |\Delta P| > \xi \end{cases}, \tag{23}$$

where, ΔP is the prediction residual and ξ is the hyperparameter determined during training.

4 Experimental verification and analysis

In this study, the proposed algorithm was evaluated on the nuScenes [48] and Radiate [49] datasets and then compared with the current popular object detection algorithms. The robustness of the proposed algorithm under different weather conditions was tested on the Radiate dataset. Ablation experiments were also performed on the nuScenes dataset. Finally, to verify the operation effect of the proposed algorithm in an actual scene, relevant experiments were conducted based on a real autonomous car platform. The proposed algorithm ran in a PyTorch framework, which was loaded on a computer with Ubuntu20.04, i7-9700k CPU, and dual 2080Ti GPU.

4.1 Tests on nuScenes dataset

The nuScenes dataset is a large-scale autonomous driving dataset, and it includes a camera and lidar and records radar data. It comprises over 1,000 scenes, including 28,130 training and 6,019 validation samples [48]. It also generally uses the NuScenes detection score (NDS) as a metric, which is a weighted sum of mAP and error metrics.

The performance of several 3D algorithms for object detection was compared on the nuScenes dataset, including lidar-based InfoFocus [50] and PointPillars [51], camera-based MonoDIS [52] and CenterNet [37], camera-radar-based CenterFusion [13], and camera-lidar-radar-based DWD-Fusion [53]. As presented in Table 1, "C," "R," and "L" represents whether a camera, radar, or lidar was used. Several indicators, including NDS, mAP, mATE, mASE, mAOE, mAVE, and mAAE, were selected for the evaluation. Meanwhile, mATE, mASE, mAOE, mAVE, and mAAE represented errors in mean translation, scale, orientation, velocity, and attributes, respectively. The up arrow " ↑ " and the down arrow " ↓ " imply that higher was better and lower was better, respectively.

In Table 1, the NDS of the proposed algorithm was higher than that of other methods. Specially, it was 21.25%, 6.83%, 8.02%, and 5.21% higher than CenterNet, PointPillars, CenterFusion, and DWD-Fusion, respectively. The indicator showed the remarkable comprehensive performance of the proposed algorithm. Moreover, lidar-based InfoFocus not only outperformed other algorithms in mAP, but also the error in feature and attribute prediction was significantly lower than that of other algorithms. Compared with CenterFusion [13], the proposed algorithm incorporated a more accurate lidar, which had a significant improvement in all performance

Table 2 Comparison of the per-class performance for 3D object detection on the nuScenes dataset

Method	Modality			mAP↑					
	C	R	L	Car	Truck	Bus	Pedestrian	Motorcycle	Bicycle
InfoFocus [50]			√	0.779	0.314	**0.448**	**0.634**	0.290	0.061
MonoDIS [52]	√			0.478	0.220	0.188	0.370	0.290	0.245
CenterNet [37]	√			0.536	0.270	0.248	0.375	0.291	0.207
PointPillars [51]			√	0.685	0.234	0.283	0.403	0.302	0.212
CenterFusion [13]	√	√		0.509	0.258	0.234	0.370	0.314	0.201
DWD-Fusion [53]	√	√	√	0.664	0.253	0.269	0.445	0.278	0.220
Ours	√	√	√	**0.785**	**0.331**	0.315	0.467	**0.329**	**0.245**

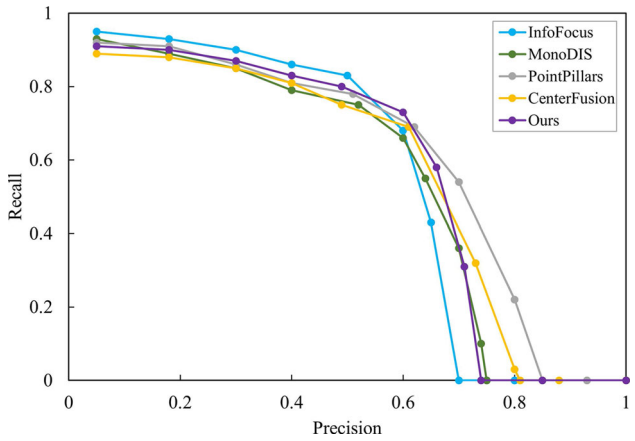


Fig. 8 Recall-Precision curves

indicators. These results demonstrated the reliability of fusing lidar data. Similar to CenterFusion [13], DWD-Fusion [53] also incorporated three types of sensor features. Compared with this method, the proposed algorithm still achieved higher scores in all performance indicators. Moreover, not only the prediction results were accurate, but also ensured the error minimization in terms of speed and direction.

Table 2 presents the detection accuracy of each algorithm on the nuScenes dataset for various objects. The average accuracy of the proposed algorithm for cars, trucks, motorcycles, and bicycles was better than that of other algorithms, which also had better detection results for medium objects. Compared with CenterFusion [13] and DWD-Fusion [53], the proposed algorithm achieved higher scores in detection accuracy for all types of targets, which indicated the superior detection performance of the proposed algorithm.

Figure 8 shows the precision and recall curves of different algorithms. The proposed algorithm considered both precision and recall, which had a better overall performance. As object detection algorithms often required a large amount of time to process complex point cloud information, a direct method to improve the efficiency of the algorithms is removing invalid information in the point cloud. The proposed frustum association method effectively filtered out almost all invalid information in the point cloud. Meanwhile, the valid point cloud information is retained in the ROI frustum. As shown in Fig. 9, only the valid information of each vehicle was retained on the road associated with the frustum, greatly reducing the time for subsequent detection and feature extraction. Furthermore, the detection accuracy was improved to a certain extent.

Figure 10 shows the impact of the presence or absence of frustum association regarding detection time and accuracy in various scenarios. The frustum association effectively reduced the detection time in various scenarios. Furthermore, the maximum reduction reached 70.95% compared with no frustum association. Moreover, the detection accuracy was effectively improved by the frustum association in most scenarios.

To verify effectiveness of each module among the proposed algorithms, an ablation experiment was performed on the nuScenes dataset. The proposed algorithm associated lidar and radar point cloud with objects through frustum association based on CenterNet. Then, features were extracted from three types of sensors through a feature-level fusion network, which was fused to obtain a final detection result. Therefore, the ablation experiment was divided into two parts. In the first part of the experiment, CenterNet was selected as a baseline to examine the effects of the frustum

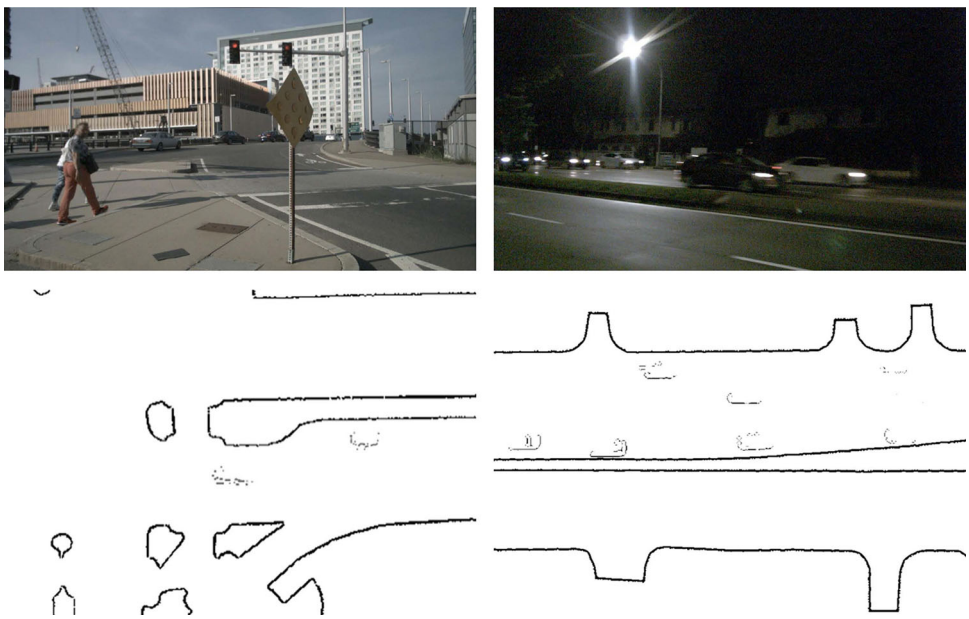


Fig. 9 The valid point cloud information retained by the view frustum association. After the frustum association, only the valid point cloud information of the object vehicle and the necessary road contour information were retained in the point cloud BEV below the corresponding image, greatly reducing the amount of point cloud processing

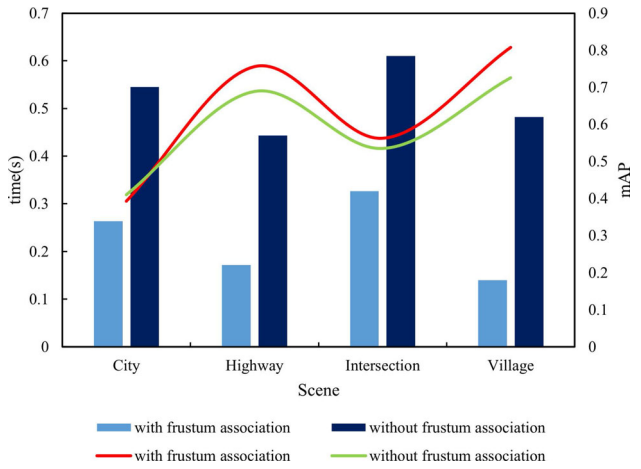


Fig. 10 Impact on the presence or absence of frustum association regarding detection time and accuracy of the detector in different scenarios. The bar graphs and curves represent time and average accuracy, respectively

association, spherical voxel convolution, and feature-level fusion network on the detection results.

Table 3 presents the results of the ablation experiments, as well as the impact of each module on the performance metrics. In the table, FA represents frustum association; SVC, spherical voxel convolution; and FFN, feature-level fusion network. The change in percentage data was compared with the benchmarking CenterNet method.

In the first experiment, only point cloud was associated with objects through the frustum. Information directly used to supplement object feature without convolution extraction and feature fusion includes such as depth, speed, and size. This simple point cloud processing method improved the NDS by 8.6% and mAP by 2.9% compared with CenterNet, which only used a camera. Various attribute errors were also reduced. In the second part of the experiment, point cloud was directly projected onto image plane. Then, an image feature and two unprocessed point cloud features were fused through a feature-level fusion network. The NDS and mAP were increased by 17.4% and 3.4%, respectively. Unlike that in the first experiment, the errors of various attributes were greatly improved. The frustum association and feature-level fusion network were further used to improve the performance of the proposed algorithm on the previous basis. Furthermore, the spherical voxel convolution greatly reduced the directional error. Finally, the NDS of the proposed algorithm was improved by 21.3% compared with that of the baseline method, and its mAP was improved by 5.6% (see Fig. 11).

For object detection algorithms, the number and type of sensors were not the more the better. A large amount of sensor data sometimes affected the judgment of the algorithm to correctly detect objects, as well as the speed and efficiency of detection. Therefore, the performance of some multimodal algorithms maybe lower than that of a single lidar-based algorithm. Comparing the role of each sensor in the fusion algorithm became the basis for judging whether the algorithm was reasonable and whether made full use of various data. The influence about three types of sensor data on the detection accuracy of the proposed algorithm was compared in the next part (see Fig. 12).

The NDS score and mAP in the four models were tested: camera-only, camera-radar, camera-lidar, and camera-radar-lidar. Figure 13 presents the results. The experimental results indicate that the multimodal method was higher than other the single-camera based method. In the case of the camera-radar-lidar model, the NDS score increased by 12.01%, and the mAP increased by 4.39% compared with that of the camera-radar model. Although radar supplies more extra features, the average accuracy still cannot be improved. However, the high accuracy of lidar significantly improves the average accuracy of the proposed algorithm. The camera-radar-lidar model is integrated in this study, which improves the accuracy and provides more object attribute feature. Therefore, the proposed method based on this model meets the need of automatic driving systems for object detection in the greatest extent.

4.2 Tests on Radiate dataset

Radiate is a severe weather dataset released by the Radiate project of Heriot-Watt University in Scotland, comprising 3h of radar images and 200,000 marked road signs, including other vehicles and pedestrians, especially for common severe weather conditions [49]. The actual effect of the algorithms validated on this dataset is helpful in examining the safety of autonomous driving in bad weather conditions.

Table 3 Ablation experiments on nuScenes dataset

Method	FA	SVC	FFN	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
Baseline				0.400	0.338	0.658	0.255	0.629	1.629	0.142
Ours	√			+8.6%	+2.9%	-15.7%	+1.9%	-4.9%	-31.6%	-11.3%
Ours			√	+17.4%	+3.4%	-37.4%	+2.2%	-8.2%	-42.0%	-17.0%
Ours	√		√	+20.0%	+4.9%	-43.1%	+1.7%	+11.9%	-52.5%	-22.3%
Ours	√	√	√	**+21.3%**	**+5.6%**	**-49.2%**	**+1.6%**	**-26.4%**	**-63.9%**	**-26.8%**

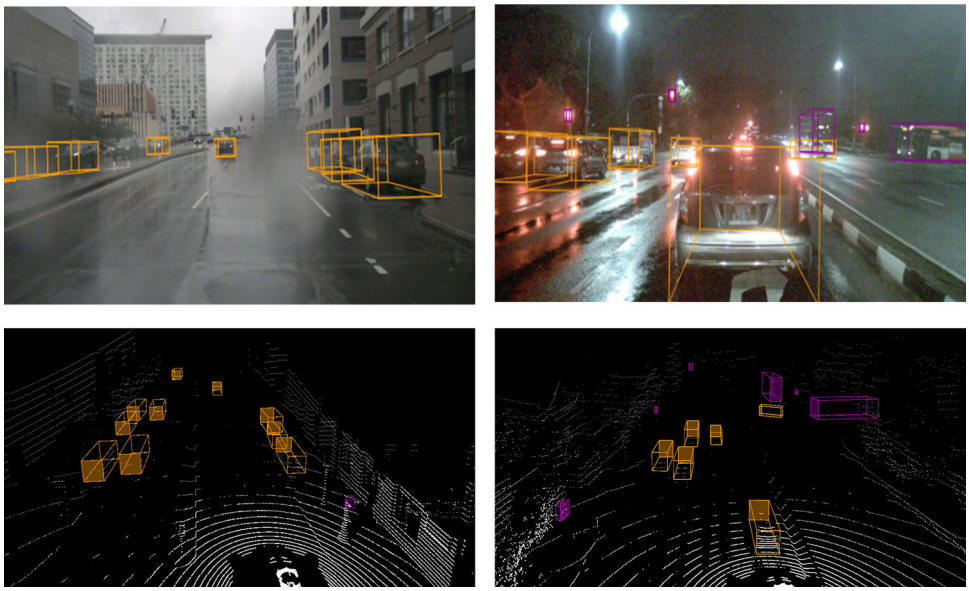


Fig. 11 Visualization in 3D maps. The top and bottom images indicate the 3D frame prediction in the images and 3D frame prediction in the point cloud, respectively

This experimental module mainly verified the robustness of the proposed algorithm under different weather conditions. In this paper, three states of the art algorithms using different sensors were selected for comparison with the proposed algorithm, and average accuracy was tested in five weather conditions: day, night, rainy, snowy, and foggy. Table 4 presents the results of the accuracy comparison among different algorithms.

The experimental results presented in Table 4 indicate that the accuracy of InfoFocus based on lidar-only was slightly higher than that of the proposed algorithm in normal weather, such as day and night. However, the proposed algorithm had clear advantages in rainy, snowy, and foggy weather conditions. Compared with that of the CenterFusion method, which achieved good results in rainy, snowy, and foggy weather conditions, the accuracy of the proposed algorithm was 8.83%, 7.02%, and 7.99% higher in these three weather conditions, respectively. The average accuracy of this algorithm was also 6.73% higher than that of DWD-Fusion, which also used three sensors.

To more intuitively reflect the performance of various algorithms, test data were drawn in a graph in Fig. 14. Generally, a camera and lidar were easily interfered in bad weather conditions and could not provide accurate scene information

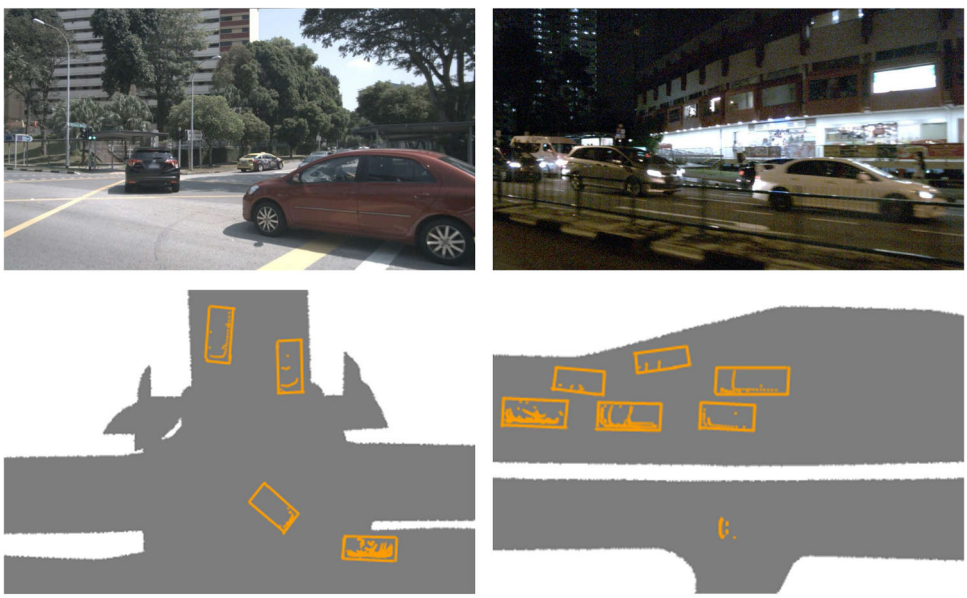


Fig. 12 BEV prediction. The top and bottom images indicate the original image and 2D box prediction in the BEV, respectively

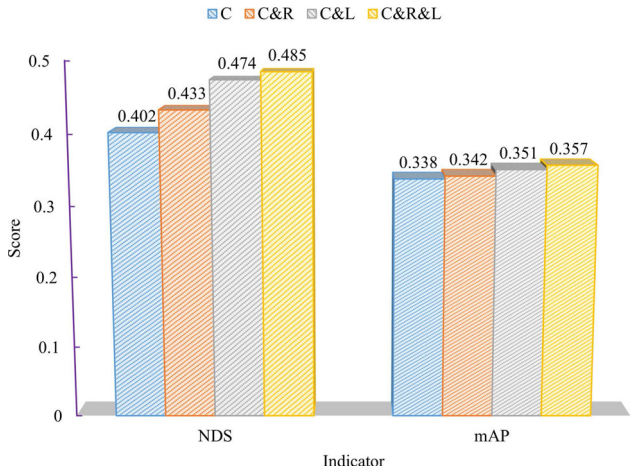


Fig. 13 Sensor ablation experiments of the proposed algorithm on the nuScenes dataset. "C," "R," and "L" represent the camera, radar, and lidar, respectively

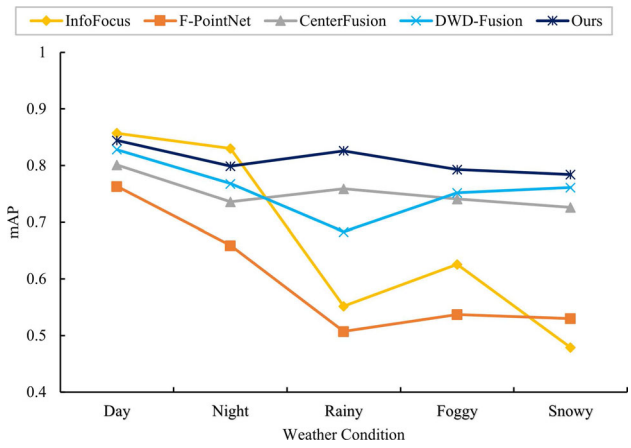


Fig. 14 Accuracy comparison curves among different algorithms on Radiate under different weather conditions

for object detection, whereas radar was well-adapted to bad weather conditions. In normal weather conditions, lidar could provide high-precision 3D point cloud information compared to a camera and radar. Figure 12 shows that the detection accuracy of InfoFocus was higher than the other algorithms during day and night, whereas multimodal algorithms were limited by different sensor fusion methods, whose accuracy was slightly lower. In rainy, snowy, and foggy weather conditions, the accuracy of InfoFocus and F-PointNet was significantly reduced, while CenterFusion, DWD-Fusion, and the proposed algorithm could still maintain a relatively stable level. Specially, the detection results were improved based on feature-level fusion by using three types of sensors. Therefore, the accuracy of the proposed algorithm was the highest in bad weather conditions.

The model was trained on the experimental platform and the loss curves were plotted in Fig. 15 for different weather conditions. In this case, the batch setting was 12, the learning rate was set to 0.01, and training was performed 100 times. Furthermore, the loss curves varied in different weather conditions. Among them, the best and second best performance were in daytime and rainy, respectively. Generally, the gradient decline was particularly evident at the beginning and then converged to a stable level at the later stage. The experimental results showed that the algorithm model had excellent prediction ability.

Based on these experiments, the proposed algorithm had better robustness in different weather conditions, and the detection performance had better stability.

Figures 16 and 17 show the visualization effect of the proposed algorithm on the Radiate dataset under normal day and that in night, rainy, snowy, and foggy weather conditions, respectively. The experimental results indicate that the proposed algorithm could achieve good detection accuracy, regardless in normal weather conditions during daytime or in extreme weather conditions (e.g., night, rainy, snowy, and foggy days).

4.3 Tests on real test site

To verify the actual operation effect of the proposed algorithm, this paper relies on an actual vehicle platform that was used to conduct road experiments in the test site of Suzhou Automotive Research Institute of Tsinghua University, where all types of advanced facilities were used to simulate various weather conditions. As shown in Fig. 18, the actual platform mainly comprises three types of sensors: camera, lidar, and radar. The camera provided clear image information, and the 64-line lidar provided rich point cloud information in a short distance. Moreover, the radar provided relatively sparse

Table 4 Accuracy comparison among different algorithms on Radiate under different weather conditions

Method	Modality			mAP↑					
	C	R	L	day	night	rain	fog	snow	average
InfoFocus [50]			√	**0.857**	**0.830**	0.552	0.626	0.479	0.669
F-PointNet [3]	√		√	0.763	0.659	0.507	0.537	0.530	0.599
CenterFusion [13]	√	√		0.801	0.736	0.759	0.741	0.726	0.753
DWD-Fusion [53]	√	√	√	0.828	0.768	0.683	0.752	0.761	0.758
Ours	√	√	√	0.844	0.799	**0.826**	**0.793**	**0.784**	**0.809**

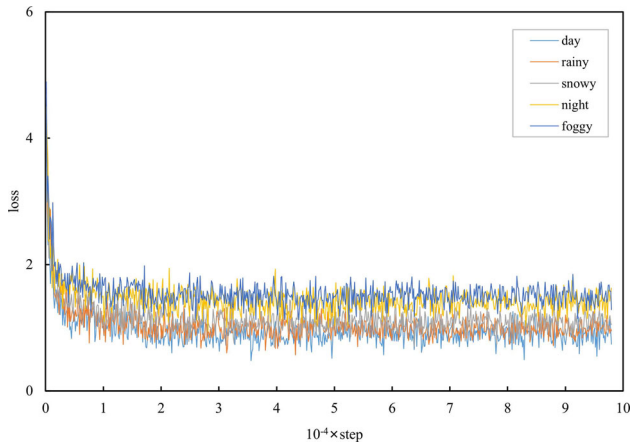


Fig. 15 Loss curves for different weather on the Radiate dataset

point cloud information in a wide range, still obtaining good results in bad weather. This algorithm mainly handled these three types of sensor data to detect surrounding vehicles in bad weather.

In the actual experiment, the detection accuracy of the proposed algorithm can meet the need of practical applications and was verified under different weather conditions. Three types of sensor data were collected from the actual platform, constructed into a dataset, and input into the proposed model. Finally, the obtained detection results were plotted to a graph in Fig. 19. The experimental results indicate that the proposed algorithm had a satisfactory detection accuracy for three types of medium and large vehicles in various weather conditions. Moreover, the mAP could reach a maximum of 0.855, which still maintained a sufficiently high accuracy even in snowy and foggy weather conditions. Experiments showed that the proposed algorithm had excellent robustness and strong generalization ability in various weather and environments.

Finally, the real scene data collected by the actual platform was placed into a trained model, and 3D object detection was performed in real time. Figure 20 shows the output feature map of point cloud. This algorithm accurately identified complex vehicles and pedestrians, which relied on the advantages of segmented fusion to make full and reasonable use of sensor data. Furthermore, the proposed method made rough predictions for distant objects, which further proved the excellent performance and mechanism of the proposed algorithm.

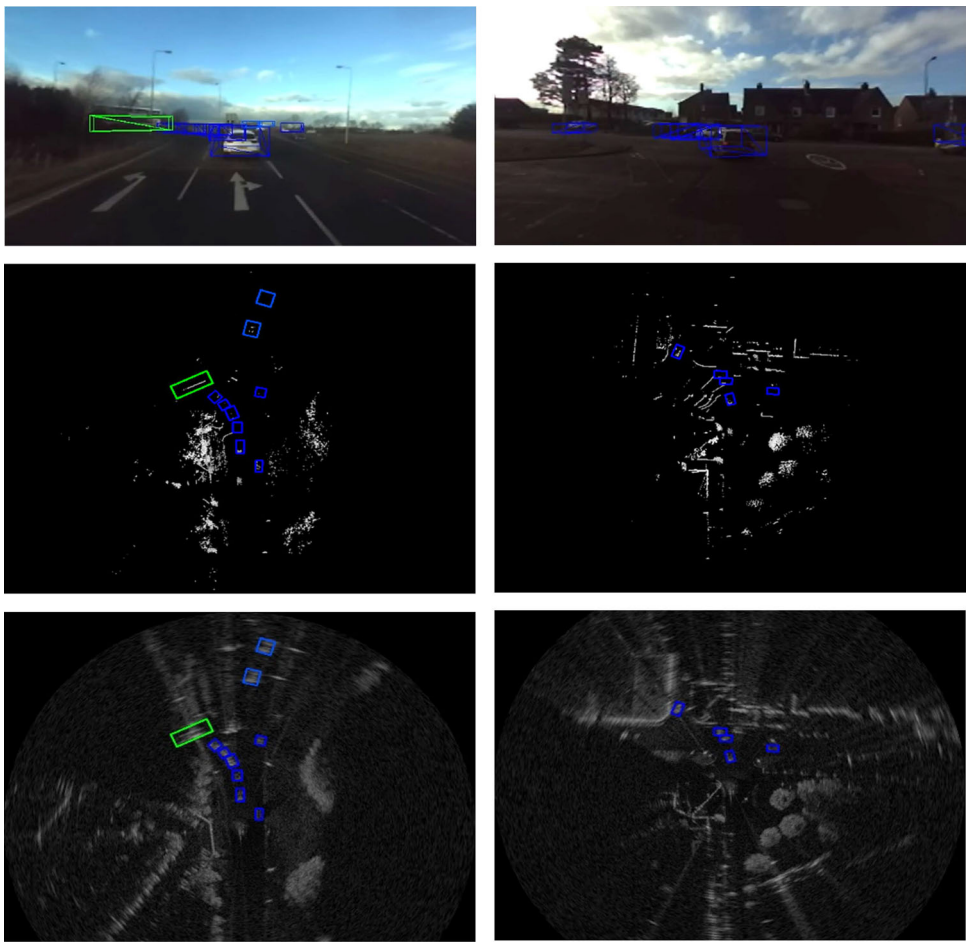


Fig. 16 Visualization under normal weather conditions during daytime. The prediction results of the camera, lidar, and radar were from top to bottom, respectively

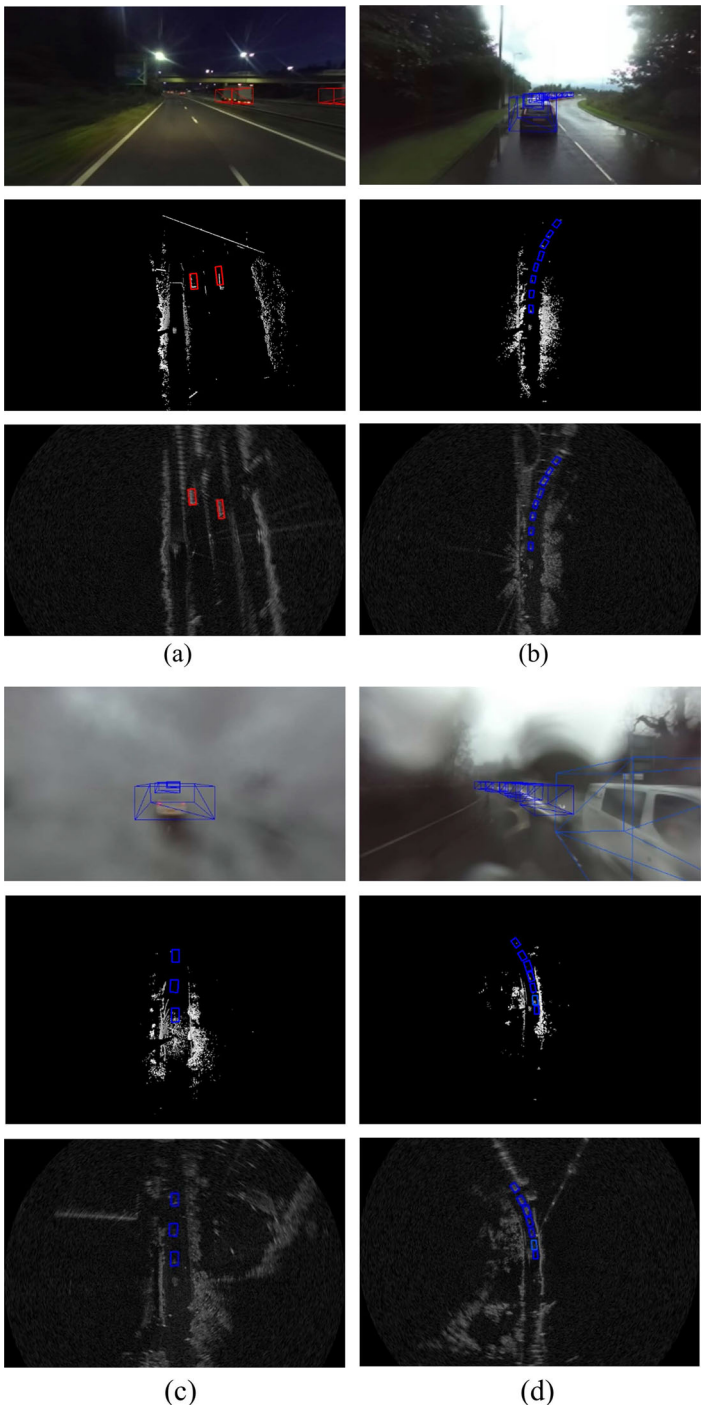


Fig. 17 Visualization effect on night, rainy, snowy and foggy days. **a** Night scene, **b** rainy scene, **c** snowy scene, **d** foggy scene

5 Conclusion

This paper proposed a multi-sensor segmental fusion of the frustum method for the 3D object detection algorithm in autonomous driving. The fusion fully exploited the advantages of each sensor and hence improved its accuracy in terms of complex weather conditions during the driving process. The frustum association method accurately associates lidar and radar detection to objects, greatly reducing the amount of point cloud detection. The spherical voxel convolution method was also exploited to extract the rotation-invariant feature of point clouds, supplementing rotation information

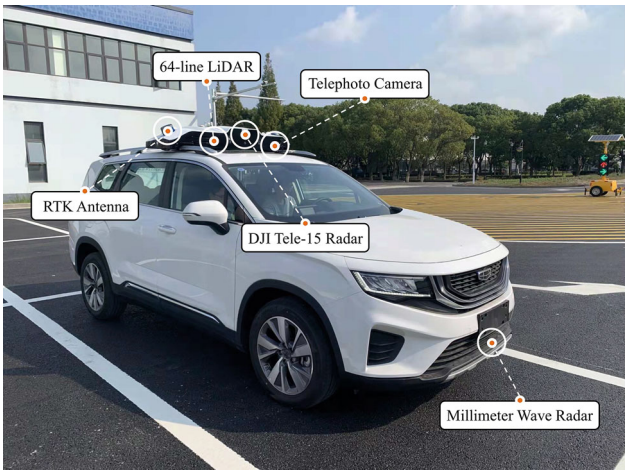


Fig. 18 Real vehicle platform in a real test site

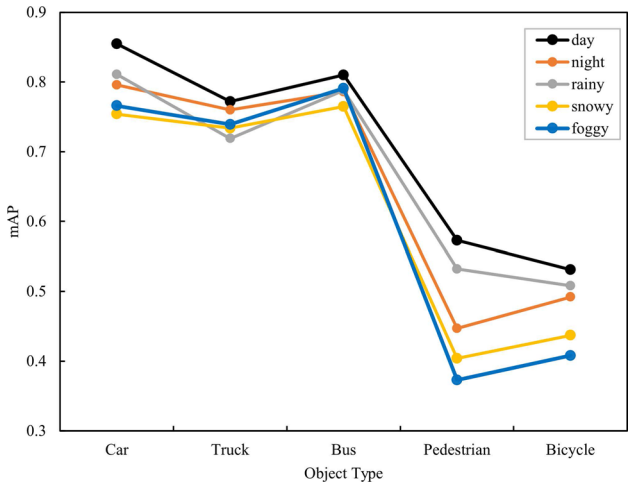
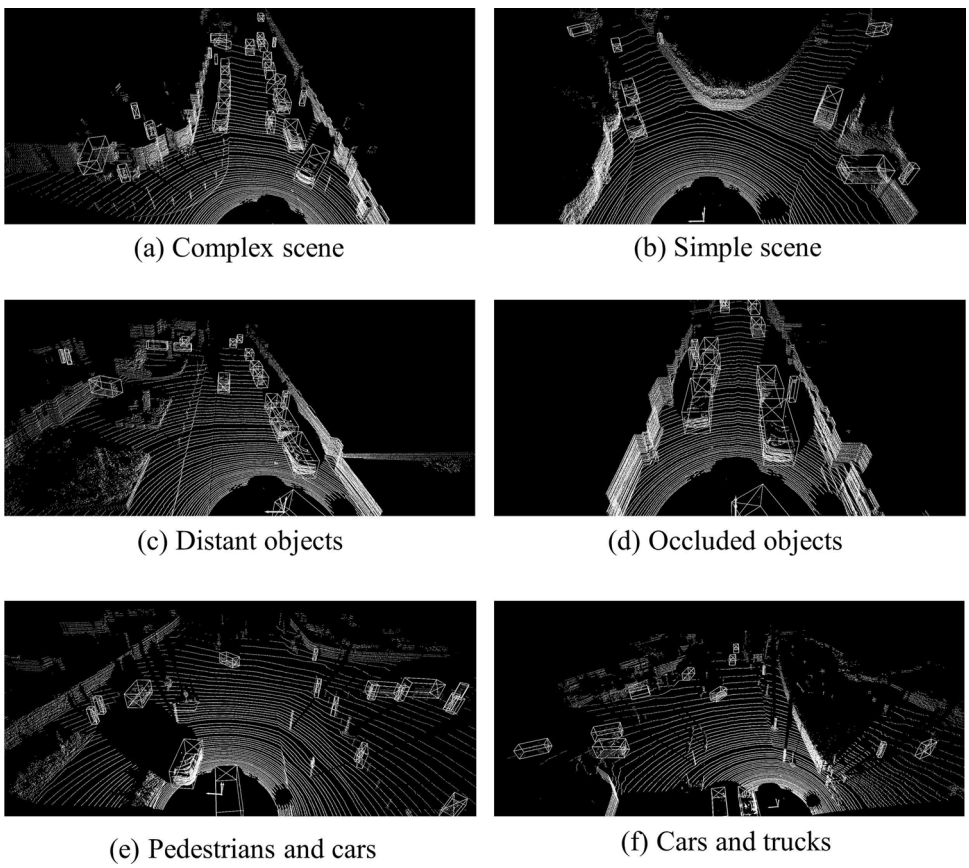


Fig. 19 Detection accuracy of various objects in different weather conditions

of the object. The dynamic adaptive feature-level fusion network both quickly and accurately obtains fusion features and improves detection results and supplements object attributes. Finally, experiments were performed on the nuScenes dataset, the Radiate dataset, and a real test site. The results indicate that the proposed algorithm has higher accuracy, richer object information, stronger generalization ability, and better robustness in complex weather conditions compared with other algorithms. Considering that the proposed algorithm relies on the 2D detection frame in the image to generate the frustum, and the image cannot provide sufficient information in extreme weather conditions (e.g., completely lightless darkness and very dense fog), further work will improve the fusion method to handle various complex situations.

Fig. 20 Visualization of the detection results from the real dataset



Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Grant No.62201375, Grant No.61972454), by the China Postdoctoral Science Foundation (2021M691848), by the Natural Science Foundation of Jiangsu Province (BK20220635,BK20201405), by the Science and Technology Projects Fund of Suzhou (Grant No. SYG202142).

Data Availability The datasets used in this study are publicly available online.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

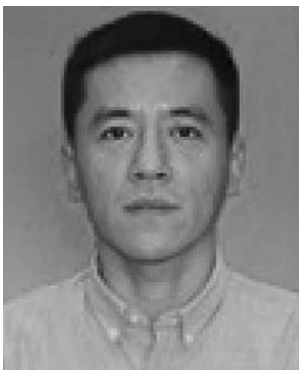
1. Chen X, Ma H, Wan J, Li B, Xia T (2017) Multi-view 3D object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 1907–1915
2. Li X, Kong, D (2022) SRIF-RCNN: sparsely represented inputs fusion of different sensors for 3D object detection. Appl Intell 1–22
3. Qi CR, Liu W, Wu C, Su H, Guibas LJ (2018) Frustum pointnets for 3D object detection from RGB-D data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 918–927
4. Yang B, Guo R, Liang M, Casas S, Urtasun R (2020) Radarnet: exploiting radar for robust perception of dynamic objects. In: European Conference on Computer Vision. Springer, pp 496–512
5. Xu C, Li Q, Zhou M, Zhou Q, Zhou Y, Ma Y (2022) RGB-T salient object detection via CNN feature and result saliency map fusion. Appl Intell 1–20
6. Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 652–660
7. Liu H, Wang X, Zhang W, Zhang Z, Li Y-F (2020) Infrared head pose estimation with multi-scales feature fusion on the irhp database for human attention recognition. Neurocomputing 411:510–520
8. Liu T, Liu H, Li Y-F, Chen Z, Zhang Z, Liu S (2019) Flexible FTIR spectral imaging enhancement for industrial robot infrared vision sensing. IEEE Trans Industr Inf 16(1):544–554
9. Liu H, Nie H, Zhang Z, Li Y-F (2021) Anisotropic angle distribution learning for head pose estimation and attention understanding in human-computer interaction. Neurocomputing 433:310–322
10. Liu H, Liu T, Zhang Z, Sangaiah AK, Yang B, Li Y (2022) ARHPE: asymmetric relation-aware representation learning for head pose estimation in industrial human-computer interaction. IEEE Trans Industr Inf 18(10):7107–7117
11. Liu H, Zheng C, Li D, Shen X, Lin K, Wang J, Zhang Z, Zhang Z, Xiong NN (2021) EDMF: efficient deep matrix factorization with review feature learning for industrial recommender system. IEEE Trans Industr Inf 18(7):4361–4371
12. Wang Z, Jia K (2019) Frustum convnet: sliding frustums to aggregate local point-wise features for Amodal 3D object detection. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 1742–1749
13. Nabati R, Qi H (2021) Centerfusion: center-based radar and camera fusion for 3D object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp 1527–1536
14. Tao C, Fu S, Wang C, Luo X, Li H, Gao Z, Zhang Z, Zheng S (2022) F-PVNET: frustum-level 3D object detection on point-voxel feature representation for autonomous driving. IEEE Internet Things J
15. Tao C, He H, Xu F, Cao J (2021) Stereo priori RCNN based car detection on point level for autonomous driving. Knowl-Based Syst 229:107346
16. Vora S, Lang AH, Helou B, Beijbom O (2020) Pointpainting: sequential fusion for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 4604–4612
17. Shi S, Wang X, Li H (2019) POINTRCNN: 3D object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 770–779
18. Yan Y, Mao Y, Li B (2018) Second: sparsely embedded convolutional detection. Sensors 18(10):3337
19. Wang Y, Chao W-L, Garg D, Hariharan B, Campbell M, Weinberger KQ (2019) Pseudo-lidar from visual depth estimation: bridging the gap in 3D object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 8445–8453
20. Chang J-R, Chen Y-S (2018) Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 5410–5418
21. Nakrani NM, Joshi MM (2022) A human-like decision intelligence for obstacle avoidance in autonomous vehicle parking. Appl Intell 52(4):3728–3747
22. Ku J, Mozifian M, Lee J, Harakeh A, Waslander SL (2018) Joint 3D proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 1–8
23. Li Y, Zhou S, Chen H (2022) Attention-based fusion factor in FPN for object detection. Appl Intell 1–10
24. Xie L, Xiang C, Yu Z, Xu G, Yang Z, Cai D, He X (2020) PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module. Proceedings of the AAAI Conference on Artificial Intelligence 34:12460–12467
25. 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). pp. 641–656
26. Pal SK, Pramanik A, Maiti J, Mitra P (2021) Deep learning in multi-object detection and tracking: state of the art. Appl Intell 51(9):6400–6429
27. Wang S, Suo S, Ma W-C, Pokrovsky A, Urtasun R (2018) Deep parametric continuous convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 2589–2597
28. Yoo JH, Kim Y, Kim J, Choi JW (2020) 3D-CVF: generating joint camera and lidar features using cross-view spatial feature fusion for 3D object detection. In: European Conference on Computer Vision. Springer, pp 720–736
29. Liang M, Yang B, Chen Y, Hu R, Urtasun R (2019) Multi-task multi-sensor fusion for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 7345–7353
30. Huang T, Liu Z, Chen X, Bai X (2020) EPNET: enhancing point features with image semantics for 3D object detection. In: European Conference on Computer Vision. Springer, pp 35–52
31. Piergiovanni A, Casser V, Ryoo MS, Angelova A (2021) 4D-net for learned multi-modal alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp 15435–15445
32. Shi S, Guo C, Jiang L, Wang Z, Shi J, Wang X, Li H (2020) PV-RCNN: point-voxel feature set abstraction for 3D object detection.

In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 10529–10538
33. Yang Z, Sun Y, Liu S, Jia, J (2020) 3DSSD: point-based 3D single stage object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 11040–11048
34. Cao J, Tao C, Zhang Z, Gao Z, Luo X, Zheng S, Zhu Y (2023) Accelerating Point-Voxel representation of 3D object detection for automatic driving. IEEE Transactions on Artificial Intelligence
35. Wang Y, Ye T, Cao L, Huang W, Sun F, He F, Tao D (2022) Bridged transformer for vision and point cloud 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 12114–12123
36. Gao Y, Liu X, Li J, Fang Z, Jiang X, Huq KMS (2022) LFT-NET: local feature transformer network for point clouds analysis. IEEE Trans Intell Transp Syst
37. Zhou X, Koltun V., Krähenbühl P (2020) Tracking objects as points. In: European Conference on Computer Vision. Springer, pp 474–490
38. Yin T, Zhou X, Krahenbuhl P (2021) Center-based 3D object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 11784–11793
39. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp 2980–2988
40. Tsai D, Worrall S, Shan M, Lohr A, Nebot E (2021) Optimising the selection of samples for robust lidar camera calibration. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, pp 2631–2638
41. Verma S, Berrio JS, Worrall S, Nebot E (2019) Automatic extrinsic calibration between a camera and a 3D lidar using 3D point and plane correspondences. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, pp 3906–3912
42. Park Y, Yun S, Won CS, Cho K, Um K, Sim S (2014) Calibration between color camera and 3D lidar instruments with a polygonal planar board. Sensors 14(3):5333–5353
43. You Y, Lou Y, Liu Q, Tai Y-W, Ma L, Lu C, Wang W (2020) Pointwise rotation-invariant network with adaptive sampling and 3D spherical Voxel convolution. Proceedings of the AAAI Conference on Artificial Intelligence 34:12717–12724
44. Esteves C, Allen-Blanchette C, Makadia A, Daniilidis K (2018) Learning so (3) equivariant representations with spherical CNNS. In: Proceedings of the European Conference on Computer Vision (ECCV). pp 52–68
45. Seghouane A-K, Shokouhi N (2019) Adaptive learning for robust radial basis function networks. IEEE Trans Cybern 51(5): 2847–2856
46. Zouari M, Baklouti N, Sanchez-Medina J, Kammoun HM, Ayed MB, Alimi AM (2020) PSO-based adaptive hierarchical interval type-2 fuzzy knowledge representation system (PSO-AHIT2FKRS) for travel route guidance. IEEE Trans Intell Transp Syst
47. Liu S, Johns E, Davison AJ (2019) End-to-end multi-task learning with attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1871–1880
48. Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O (2020) nuscenes: a multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 11621–11631
49. Barnes D, Gadd M, Murcutt P, Newman P, Posner I (2020) The Oxford Radar Robotcar Dataset: a radar extensionto the Oxford Robotcar Dataset. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 6433–6438
50. Wang J, Lan S, Gao M, Davis LS (2020) Infofocus: 3D object detection for autonomous driving with dynamic information modeling. In: European Conference on Computer Vision. Springer, pp 405–420
51. Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O (2019) Pointpillars: fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 12697–12705
52. Simonelli A, Bulo SR, Porzi L, Antequera ML, Kontschieder P (2020) Disentangling monocular 3D object detection: from single to multi-class recognition. IEEE Trans Pattern Anal Mach Intell
53. Liu Q, Zhou W, Zhang Y, Fei X (2021) Multi-target detection based on multi-sensor redundancy and dynamic weight distribution for driverless cars. In: 2021 International Conference on Communications, Information System and Computer Engineering (CISCE). IEEE, pp 229–234

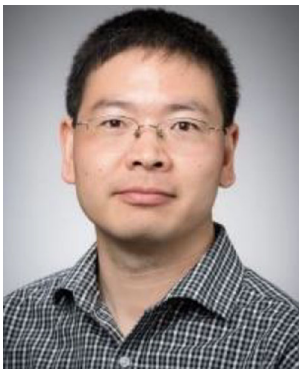
Chongben Tao (Associate professor) received the Ph.D degree from Jiangnan University, Wuxi, China,in 2014. He is the Faculty of Compute Science at Suzhou University of Science and Technology. And he also is a postdoctoral fellow of Suzhou Automobile Research Institute of Tsinghua University. His current research interests include autonomous driving, advanced robotics and automation.

Weitao Bian received the B.E. degree from Suzhou University of Science and Technology, Suzhou, China. His research interests include automatic driving and 3D object detection.

Chen Wang received the B.S. degree from the Ocean University of China, Qingdao, China, in 2015, and the Ph.D degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2021, respectively. He is currently an instructor with School of Electronic and Information Engineering, Suzhou University of Science and Technology. His research interests include SAR/remote sensing image processing and deep learning.



Huayi Li received the master's degree from Soochow University, Suzhou, China, in 2003. He is the Director of the Science and Technology Industry Department, Suzhou University of Science and Technology, Suzhou, China. His research interests include mathematics, informatics, and scientific research management.



Zhen Gao (Associate professor) received the Ph.D degree from the University of Science and Technology of China, Hefei, China, in 2010. He is the Faculty of Engineering at McMaster University. He was the Program Lead for Systems & Technology at McMaster which was focused on Cyber Physical System. His current research interests include industrial controllers, advanced robotics and automation, artificial intelligence, neural networks and pattern recognition.

Zufeng Zhang is currently pursuing the Ph.D degree with the Department of Automation, Tsinghua University. His current research interests include autonomous driving and advanced robotics.

Sifa Zheng received the B.E. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1993 and 1997, respectively. He is currently a Professor with the School of Vehicle and Mobility, and the State Key Laboratory of Automotive Safety and Energy, Tsinghua University. He is also the Deputy Director with Suzhou Automotive Research Institute, Tsinghua University, Suzhou, China. His current research interests include autonomous driving and vehicle dynamics and control.

Yuan Zhu received his double B.S. degree from Tsinghua University (China) in 1998 respectively in automotive engineering and computer application technology. He received his Ph.D. degrees from Tsinghua University (China) in 2003 in automotive engineering. He has been an assistant research scientist in department of automotive engineering at Tsinghua University during 2003–2005. During 2014–2019, he has been responsible for the Hans L. Merkle Foundation - Bosch Endowed Chair for Automotive Systems at Tongji University. Since 2017, he is the executive director for Tongji University - Vector Automotive Technology Joint Laboratory. His current research interests include simulation and control for electric drive system, embedded software system.

Authors and Affiliations

Chongben Tao[1,2,3] · Weitao Bian[1] · Chen Wang[1] · Huayi Li[1] · Zhen Gao[4] · Zufeng Zhang[5] · Sifa Zheng[5] · Yuan Zhu[6]

Chongben Tao
tom1tao@163.com

Weitao Bian
bwt0423@163.com

Huayi Li
lihuayi@usts.edu.cn

Zhen Gao
gaozhen@mcmaster.ca

Zufeng Zhang
zhangzufeng@tsari.tsinghua.edu.cn

Sifa Zheng
zsf@tsinghua.edu.cn

Yuan Zhu
yuan.zhu@tongji.edu.cn

1 Department of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China

2 Suzhou Automobile Research Institute, Tsinghua University, Suzhou 215134, China

3 College of Mechanical Engineering, Tongji University, Shanghai 200333, China

4 Faculty of Engineering, McMaster University, Hamilton ON L8S 0A, Canada

5 Tsinghua University, Beijing 100084, China

6 College of Automotive Studies, Tongji University, Shanghai 201804, China