

TractorRoadBEV: A BEV-Based Multi-View Camera Fusion Model for 3D Object Detection on Tractor Road

Lili Yang, Mengshuai Chang, Xiao Guo, and Jiale Li

Abstract—Multi-view image fusion for three-dimensional (3D) object detection in agricultural scenarios has received limited research attention. To address this issue, we propose TractorRoadBEV, a multi-stage multi-view perception framework. The proposed framework first performs two-dimensional (2D) object detection and mask enhancement to extract object-aware features from images. The multi-view features are then transformed into a unified bird's eye view (BEV) representation through a depth mapping module and a spatial unification module. Subsequently, a temporal fusion module integrates the current BEV feature map with historical BEV features to enhance temporal consistency. Finally, the fused BEV feature map is decoded to obtain 3D object detection results. Experimental results on the constructed tractor road dataset show that TractorRoadBEV achieves F1 scores of 0.8307 and 0.8166 for person detection and vehicle detection, respectively, which significantly outperforms the classical monocular detection method Deep3DBox. In addition, the backbone network of TractorRoadBEV demonstrates strong generalization capability in BEV scene segmentation, achieving an intersection over union (IoU) of 0.602, which is 0.151 higher than that of the classical lift-splat-shoot (LSS) method. These results indicate that the proposed method provides a low-cost, efficient, and scalable perception solution for autonomous agricultural machinery in tractor road environments.

Index Terms—Bird's eye view (BEV), three-dimensional (3D) object detection, multi-camera, multi-stage, tractor road

I. INTRODUCTION

WITH global agricultural production facing labor shortages, and with growing demands for higher operational efficiency and precision agriculture, autonomous driving technology is becoming a core direction for the development of modern agricultural machinery [1–3]. At the same time, the level of autonomous driving intelligence is directly related to how well the vehicle can sense its surroundings [4, 5]. In agricultural scenarios, the tractor road presents one of the most complex environments for agricultural machines, primarily due to its diverse types and low degree of structuring. In order for agricultural machinery to be able to drive autonomously, accurate three-dimensional (3D) detection of major objects on the tractor path is essential.

Manuscript received: 12 February 2026; revised: 16 March 2026; accepted: 18 March 2026. (Corresponding author: Lili Yang.)

Citation: L. Yang, M. Chang, X. Guo, and J. Li, TractorRoadBEV: A BEV-based multi-view camera fusion model for 3D object detection on tractor road, *Int. J. Intell. Control Syst.*, 2026, 31(1), 61–71.

Lili Yang, Mengshuai Chang, Xiao Guo, and Jiale Li are with College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China (e-mail: lilyang@cau.edu.cn; cmsfamily@126.com; foundnom@cau.edu.cn; jiale.li@cau.edu.cn).

Digital Object Identifier 10.62678/IJICS202603.10309

Cameras are widely used in 3D object detection in autonomous driving because of their low cost, ease of integration and deployment, and the ability to obtain rich texture information. Many camera-based solutions have been proposed. Early traditional algorithms [6–8] mainly relied on geometric knowledge and optical models as their theoretical foundation. Then, detection was completed by manually designing and extracting features, combined with the assistance of some prior information. Due to their simple model architectures, these methods are very efficient. However, their heavy dependence on handcrafted features and simplified assumptions about real-world scenes limits their generalization ability and makes them less effective in complex environments.

In recent years, many deep learning-based monocular methods [9–11] have been proposed. By training on large-scale datasets, these methods can learn feature representations and 3D structural information of objects from images, and then perform end-to-end object detection. Some methods [12–14] have also achieved good results by combining prior information, decomposing tasks, and performing multi-stage feature extraction. These methods often have good generalization, but in actual deployment, the settings of monocular camera will be limited by the field of view. The perception range of monocular camera is difficult to meet the needs of autonomous vehicles. To address this limitation, some large-scale and high-impact datasets [15, 16] were proposed. They often adopted a multi-camera, multi-perspective deployment solution to perceive a wider range. However, when a monocular method is deployed on a multi-camera platform, the inference of the algorithm needs to be performed separately for each camera view [17]. This not only needs extra post-processing of each camera's detection results based on its positional relationship, but also linearly increases the need for computing power and processing time. To solve the problem of field of view limitations of a monocular camera, deep learning-based multi-camera algorithms have emerged. The bird's eye view (BEV) paradigm [18–21] is a typical representative. These methods extract BEV features around a vehicle body by combining features from cameras in different perspectives, and then perform downstream segmentation, detection, and tracking tasks in a unified manner. Compared with the monocular camera algorithm, it not only solves the field of view limitation but also avoids extra post-processing operations. It also avoids repeated calculations for object

perception in the camera's common view area to a certain extent, as well as the misdetection problems that are easily caused by monocular camera. However, this paradigm is often applied to large-scale datasets, and an end-to-end model is often used to ensure generalization, which often requires high computation power. In addition, most current solutions are based on structured traffic scenarios, and research on unstructured or semi-structured agricultural traffic scenarios such as tractor roads is still in the exploratory stage.

Tractor roads are often unstructured and usually lack clear boundaries. Therefore, we propose a low-cost, efficient, easy-to-deploy, and scalable real-time multi-camera fusion perception model, TractorRoadBEV, for the small-scale tractor road scenario by combining the advantages and disadvantages of existing methods. We collected and constructed a tractor road multi-camera object detection dataset by using a light detection and ranging (LiDAR) to simultaneously drive three cameras from different perspectives. Then, we propose TractorRoadBEV to complete 3D object detection by using a multi-stage model. In the multi-view feature fusion stage, a mature two-dimensional (2D) object detector is first used to separately recognize 2D objects from images of different views and add image masks so as to assist the depth mapping module in quickly locating the object region. Then, the depth features of each view image output by the depth mapping module are spatially unified to obtain a preliminary BEV feature map. And then the temporal fusion stage fuses the initial BEV feature map with the historical BEV feature map to obtain the final BEV feature map. Finally, in the 3D object detection stage, the resulting BEV feature map is decoded by different detection heads to obtain 3D information about the objects in the scene.

TractorRoadBEV overcomes the field of view limitations of monocular camera through a multi-camera setup and greatly reduces the cost of extended deployment. Compared with most BEV methods, its multi-stage method causes cumulative errors to a certain extent and reduces generalization. However, it can also work in specific scenarios with small datasets. In addition, the multi-stage method is easy to debug and modify, and specific prior knowledge can be added to compensate for cumulative errors in specific scenarios. These characteristics make TractorRoadBEV a good fit for the agricultural tractor road scenario constructed in this paper.

II. RELATED WORK

In recent years, in the field of autonomous driving technology, thanks to the low cost of camera equipment and its natural fit with the human visual perception mechanism, a series of purely visual 3D object detection algorithms based on deep learning has been continuously proposed [22, 23]. While maintaining low cost, these algorithms have achieved significant performance and provide important technical support for the perception capabilities of autonomous driving systems.

A. Single Camera 3D Object Detection Algorithm

Mono3D [12] performs 3D object detection in two stages. It generates dense 3D candidate boxes, then scores them by extracting image features and combining positional prior

information and semantic context to obtain accurate 3D detection boxes for objects. This method achieved good performance in the KITTI [24] benchmark test, but its dense sampling requires a high computation power. Afterwards, Deep3DBox [13] directly learns 3D information with less variance between different objects based on a 2D object detection network and calculates the information of the 3D box. Compared with Mono3D, this algorithm has a simple architecture, which greatly reduces inference time. However, for the multi-stage approach, cumulative errors are always inevitable. Therefore, SMOKE [9] abandons 2D object detection and achieves monocular 3D detection solely through keypoint estimation and direct regression of 3D bounding boxes. Fcos3d [10], on the other hand, achieves good results by decoupling 3D attributes and then conducting multi-layer 3D predictions guided by 2D information while freeing itself from any 2D detection and the prior associations from two to three dimensions. PGD [25] further proposes a customized detection head module to improve the depth regression part of Fcos3d. However, these end-to-end methods often require a large amount of data to support training. In addition to monocular cameras, stereo cameras are also widely utilized in object detection. Stereo-RCNN [26] simultaneously detects and correlates objects in the left and right images, then combines the 2D bounding boxes from these images with predicted key points, viewpoints, and object sizes to estimate a rough 3D bounding box, and finally refines the estimate to an accurate 3D box using region-based photometric alignment. SGM3D [27] introduces a multi-granularity domain adaptation module and an alignment method based on intersection over union (IoU) matching, using the 3D features learned from stereo images to enhance the features for monocular detection. These single-camera methods each have their own strengths and weaknesses, fitting various scene requirements, but they all face challenges in scalability when it comes to practical deployment. MonoDETR [28] adopts a depth-guided transformer, in which each object adaptively estimates its 3D attributes by querying depth-guided regions in the image, rather than relying on local features. MonoDINO-DETR [29] employs a vision transformer (ViT) based design and introduces a hierarchical feature fusion module to enhance object localization capability.

B. Multi-Camera 3D Object Detection Algorithm

To address the scalability and field-of-view limitations of single cameras in autonomous driving perception, research on multi-camera configurations is continuously advancing. Among these, the BEV paradigm stands out due to its unified feature representation, offering significant advantages in data fusion, spatial understanding, and multi-task scalability. Currently, methods based on BEV are mainly divided into two categories: bottom-up approaches and top-down approaches. Bottom-up approaches first extract spatial features from each camera view and then project them into the BEV space using the camera's intrinsic and extrinsic parameters. The multi-view features are then fused to generate BEV representations for downstream tasks. Lift-splat-shoot (LSS) [18] created a bottom-up basic paradigm for BEV perception through three

steps: (1) lift: probabilistic prediction of discrete depth values for each perspective image to obtain depth features; (2) splat: pooling depth features from each perspective into voxels at corresponding positions in BEV space to obtain BEV features; and (3) shoot: using different task heads to process BEV feature maps and output perception results. BEVDet [19] follows the LSS structure and effectively acts on the subsequent network by performing additional data augmentation in the BEV space to prevent overfitting of the model while also introducing scale non-maximum suppression (NMS) to improve the robustness of the model. BEVDet4D [30] further incorporates a temporal fusion mechanism on top of BEVDet, further improving the detection accuracy.

In contrast, top-down approaches first initialize sparse or dense 3D features in the BEV space and then obtain BEV features through feature interaction and fusion with each camera. DETR3D [20] initializes multiple object queries in the BEV space and maps out the corresponding 3D reference points. These points are then projected onto 2D images from different viewpoints using intrinsic and extrinsic parameters, followed by feature interaction through the network. The results are matched one by one using the Hungarian matching algorithm, eliminating the need for the previous NMS post-processing stage. Unlike DETR3D, BEVFormer [21] does not define a query for each object. Instead, it defines a fixed-size pillar grid, where each pillar corresponds to a query. Through a self-attention mechanism, each query corresponds to a 2D reference point, and cross-attention samples height values, ultimately generating dense BEV features for 3D object detection. CorrBEV [31] focuses on the occlusion problem by introducing auxiliary prior knowledge and a random pixel dropping strategy to simulate occlusion.

Although these BEV methods have good scalability and generalizability, they require high computation power and data. The end-to-end paradigm makes it difficult to integrate prior knowledge in specific scenarios and is not friendly to small datasets.

III. MATERIALS AND METHODS

A. Dataset Production

We utilized the DF2204 tractor equipped with an industrial computer, an 80-line LiDAR, and three cameras to form a data

acquisition platform, as shown in Fig. 1. The LiDAR is installed on a horizontal bar 1.33 m above the ground. The three cameras are spaced 10.00 cm apart horizontally: the front camera faces directly ahead, while the left-front and right-front cameras are tilted at 45° towards the left-front and right-front, respectively. The maximum distance of the blind spot between adjacent cameras is 36.34 cm, with a blind spot area of 181.2 cm², which is triangular and unable to cover the space occupied by a person. This arrangement ensures a certain overlap in the field of view within the maximum visible range, and the size of the blind spots meets the actual operational requirements. The entire camera system is installed on a bar 1.03 m above the ground, maintaining the same vertical coordinate axis as the LiDAR sensor, and is triggered synchronously for data collection by the LiDAR. The industrial computer is installed in the tractor's cab and receives data collected by the LiDAR and camera sensors via an Ethernet protocol.

Detailed parameters of each device are shown in Table 1.

Data acquisition was carried out in Henanzhai Town, Miyun District, Beijing, in March 2024. The operator drove the agricultural machinery at a constant speed of no more than 20 km/h on the tractor road for approximately 1 h, resulting in a total of 765 GB of valid data. We utilized the SensorsCalibration [32] toolbox to perform extrinsic calibration of the cameras and LiDAR using the acquired data. By applying Eq. (1), we aligned the coordinate systems of the three cameras to the central reference of the camera system.

$$f(x, y, z) = T_{\text{center}}^{\text{lidar}} T_{\text{cam}}^{\text{lidar}-1} [x, y, z, 1]^T, \text{ cam} \in \{\text{left, front, right}\} \quad (1)$$

where (x, y, z) represents the coordinates of each camera coordinate system, $T_{\text{center}}^{\text{lidar}}$ represents the projection matrix from the LiDAR coordinate system to the center of the camera system, and $T_{\text{cam}}^{\text{lidar}}$ represents the projection matrix from the LiDAR coordinate system to each camera coordinate system. The above equation can be used to transform the coordinate systems of all cameras to the central coordinate system of the camera system. We then selected 11 driving sections with abundant detection objects and used the point cloud data acquired by the LiDAR as the ground truth to label the 3D bounding boxes of the two main objects in the scene, person,

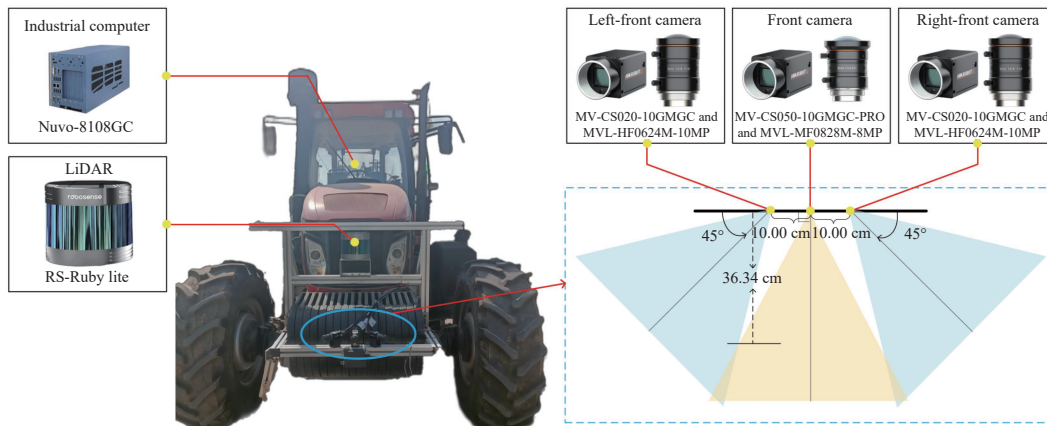


Fig. 1 Data acquisition platform.

Table 1 Device parameters.

Device name	Device model	Key parameters
Industrial computer	Nuvo-8108GC	CPU: Intel (R) Core (TM) i9-9900K@3.60 GHz, GPU: NVIDIA GeForce RTX 3060, and memory: 32 G
80-line LiDAR	RS-Ruby Lite	Acquisition frequency: 10 Hz, vertical field of view: 40° (−25°–15°), vertical resolution: 0.1°, and horizontal resolution: 0.2°
Front camera	MV-CS050-10GMGC-PRO	Acquisition frequency: 10 Hz, resolution: 1224 × 1024, pixel format: RGB 8, and sensor type: CMOS
Front lens	MVL-MF0828M-8MP	Focal length: 8 mm and field of view: D 68.46°, H 54.97°, and V 47.06°
Left/right-front camera	MV-CS020-10GMGC	Acquisition frequency: 10 Hz, resolution: 1624 × 1240, pixel format: RGB 8, and sensor type: CMOS
Left/right-front lens	MVL-HF0624M-10MP	Focal length: 6 mm and field of view: D 72.96°, H 62.46°, and V 44.05°

and cars, with reference to the labeling process in NuScenes [15]. We obtained the ground truth 2D bounding boxes for each perspective by projecting the 3D labels onto the corresponding images. In the end, we labeled a total of 2362 frames of tractor road scene data. After that, we divided the dataset into a training set, a validation set, and a test set in the ratio of 70%, 15%, and 15%, including 1654 frames of training data, 354 frames of validation set, and 354 frames of test set. Thus, a multimodal 3D object detection dataset based on tractor road scenes was constructed.

B. Methodology

In this paper, we propose TractorRoadBEV, a 3D object detection network based on three-view camera fusion.

As depicted in Fig. 2, the TractorRoadBEV network is comprised of five modules. The network is broadly divided into three stages, with modules (a), (b), and (c) carrying out multi-view image fusion, module (d) performing temporal fusion, and module (e) conducting 3D object detection. First, the multi-view image fusion stage processes synchronized images from three cameras to generate an initial BEV feature map. Second, the temporal fusion stage refines this map by incorporating historical BEV features. Finally, the 3D detection stage decodes the fused BEV features to output 3D bounding boxes for objects. The process is detailed as follows.

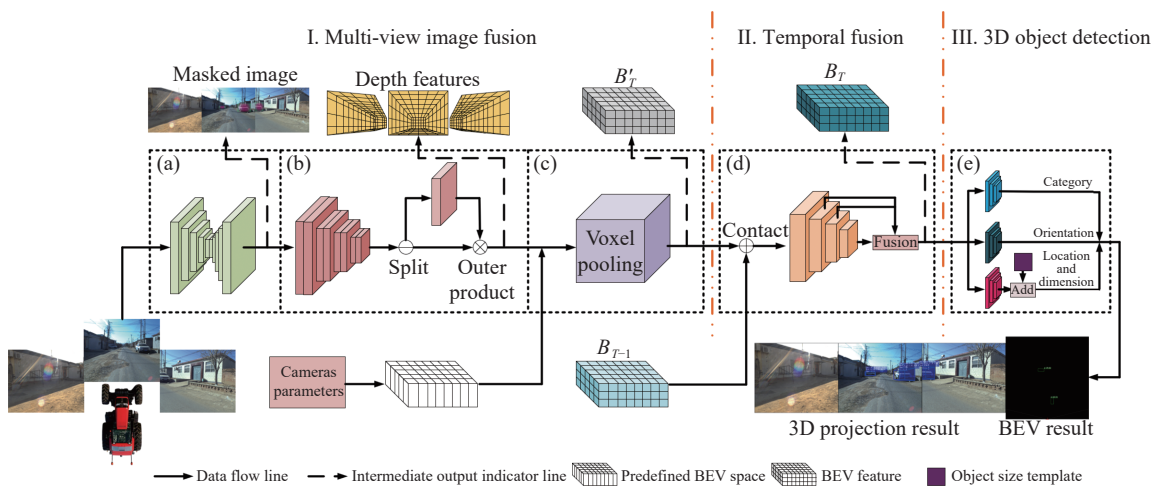
a. Multi-View Image Fusion

(1) 2D detection and mask enhancement. When estimating the depth of an image, it is necessary to estimate the depth for each pixel in the image. However, in object detection tasks, our focus is on the objects within the scene rather than

the entire scene. Therefore, we use YOLOv10 [33] to perform 2D recognition of the objects in the image and then add a center mask based on the object's 2D bounding box to enrich the features of the area where the object is located in the image. In real-world scenarios, persons, as non-rigid objects, exhibit a wide variety of poses. In addition, 2D bounding boxes of persons often contain a significant amount of background noise, while cars are prone to occlusion by foreground objects. To address these issues, we add masks with different weights to the regions where objects are located in the image based on the position of each pixel. The calculation equation for the image center mask is

$$\xi(u, v) = 1 - \frac{(u - u_0)^2 + (v - v_0)^2}{w^2 + h^2} \quad (2)$$

where u and v are the pixel coordinates within the 2D object area in the image, u_0 and v_0 are the coordinates of the object's center point, w is the width, and h is the height of the 2D object. The weights are calculated based on the distance of each pixel from the center point in the object area. If the same pixel belongs to multiple objects, the maximum value among the results is taken, and non-object areas are filled with 0. The resulting weights are concatenated as the fourth-dimensional feature to the original image. Let $X \in \mathbb{R}^{3 \times W_x \times H_x}$ be the scaled original image, where $W_x = 352$ and $H_x = 256$. After concatenating the mask, the image channel features are represented by C , with each pixel represented as (R, G, B, ξ) . Thus, $X_{\text{mask}} \in \mathbb{R}^{C \times W_x \times H_x}$ is the image with the added mask, as shown in Fig. 2(a). According to Eq. (2), pixels in the object area are assigned different weights based on their distance from the

**Fig. 2** TractorRoadBEV network structure.

center of the object image. Points closer to the center will have a higher weight.

(2) Depth estimation and feature lifting. Subsequently, the EfficientNet [34] network backbone is used to extract features from the masked image $X_{\text{mask}} \in \mathbb{R}^{C \times W_x \times H_x}$ to obtain $F \in \mathbb{R}^{(D+C_f) \times W_f \times H_f}$, where C_f represents the number of channels in the feature map, and D represents the probability of predicting D depths in the perspective direction for each pixel in the image. The probability distribution $\mathbb{R}^{D \times W_f \times H_f}$ is then outer-producted by $\mathbb{R}^{C_f \times W_f \times H_f}$ to get the depth feature $\bar{F} \in \mathbb{R}^{D \times C_f \times W_f \times H_f}$, where $D = 76$, $C_f = 64$, $W_f = 64$, and $H_f = 22$. This associates each pixel's features with every depth estimation value for that pixel, further assisting the model in depth estimation, as shown in Fig. 2(b).

(3) BEV feature pooling. Considering the driving safety of agricultural machinery, we preset the BEV space to be a rectangular area in front of the car and divide it into a voxel grid. In terms of the longitudinal range, statistics from our dataset show that images of persons beyond 76 m account for a relatively small proportion of the entire image, making 2D detection fairly difficult. For the lateral range, considering the turning areas at intersections, it needs to be wider than the typical width of a tractor road. However, dividing the grid too densely in both horizontal and vertical directions would significantly increase the computational load, while too sparsely would fail to accurately represent the object features. Therefore, we set the horizontal and vertical dimensions to be comparable to the size of a person, while the height feature is expressed uniformly without dividing the grid. The final preset BEV space range, as shown in Fig. 3, uses the camera system center O as the origin. The perception range along the direction of car travel is from 0 to 76 m, divided into 152 grids with a 0.5 m interval. The perception range perpendicular to the direction of car travel is from -24 to 24 m, divided into 96 grids with a 0.5 m interval. The height direction perception range is from -5 to 5 m, with no grid division.

Depth features estimated from images under different viewpoints are represented as $\bar{F} \in \mathbb{R}^{D \times C_f \times W_f \times H_f}$. These features are then indexed to the predefined BEV space based on their respective camera intrinsic parameters I and extrinsic parameters E . Features that fall outside the predefined range are discarded, and overlapping features within the camera's common view area are accumulated and aggregated. The final result is a BEV feature map $\bar{G} \in \mathbb{R}^{C_g \times L_g \times W_g}$ that integrates the depth features from different viewpoints, where $L_g = 152$ and $W_g = 96$. The size is consistent with the predefined BEV

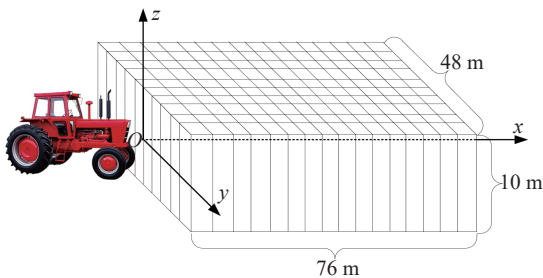


Fig. 3 Preset BEV range.

perception space, and the number of feature channels for each grid voxel is $C_g = 64$, as shown in Fig. 2(c).

b. Temporal Fusion

The movement of objects relative to the agricultural machinery in a scene is continuous. Considering that the driving speed of agricultural machinery is generally low, the change in the 3D information about objects between adjacent frames will not be significant. Therefore, in BEV perception, the BEV feature maps of historical frames theoretically have rich reference value for the perception of the current frame. Hence, the model proposed in this paper employs a temporal fusion architecture, as shown in Fig. 4. The preliminary BEV feature map of the current frame is combined with that of the previous frame and input into the temporal fusion module to fuse historical information, thereby obtaining the final BEV feature map of the current frame. The temporal fusion module is shown in Fig. 2(d).

The detailed architecture of the temporal fusion module network is shown in Fig. 5. The current preliminary feature map $\bar{G} \in \mathbb{R}^{64 \times L_g \times W_g}$ and the historical BEV feature map $G^{t-1} \in \mathbb{R}^{64 \times L_g \times W_g}$ are concatenated, and then a convolution operation is applied to obtain the feature map $\tilde{G} \in \mathbb{R}^{64 \times L_g \times W_g}$. This feature map is then fed into a feature extraction network layer composed of ResNet layers [35], resulting in feature maps $\tilde{G}_1 \in \mathbb{R}^{64 \times L_g \times W_g}$, $\tilde{G}_2 \in \mathbb{R}^{128 \times L_g/2 \times W_g/2}$, and $\tilde{G}_3 \in \mathbb{R}^{256 \times L_g/4 \times W_g/4}$ sequentially. After that, each layer's feature map is upsampled and concatenated with the feature map of the previous layer before being input into a convolutional layer. Through this multi-level feature extraction and fusion, the fusion with the historical BEV feature map is ultimately achieved, resulting in the BEV feature map G^t of the current frame $\in \mathbb{R}^{64 \times L_g \times W_g}$.

c. 3D Object Detection

The final BEV feature map is fed into the last detection heads to sequentially detect the category, location, dimension, and orientation of objects within the scene. Considering that in the specific scenario of tractor road, the sizes of both persons and cars vary within a certain range, we calculate the average size for each type of object by statistically analyzing the sizes of persons and cars in the annotated data. Based on these averages, a template 3D bounding box is created for each category and used as prior information. When the network performs regression on the objects, it only regresses the devia-

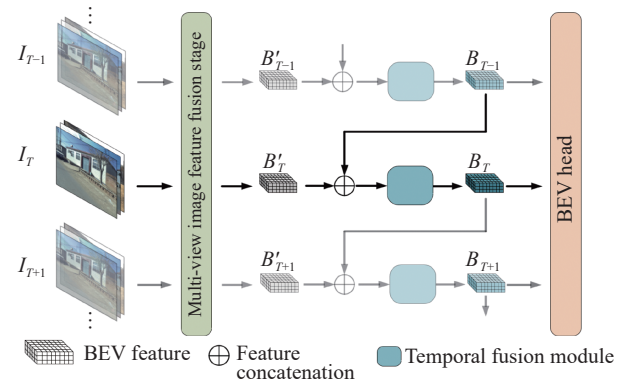


Fig. 4 Temporal architecture.

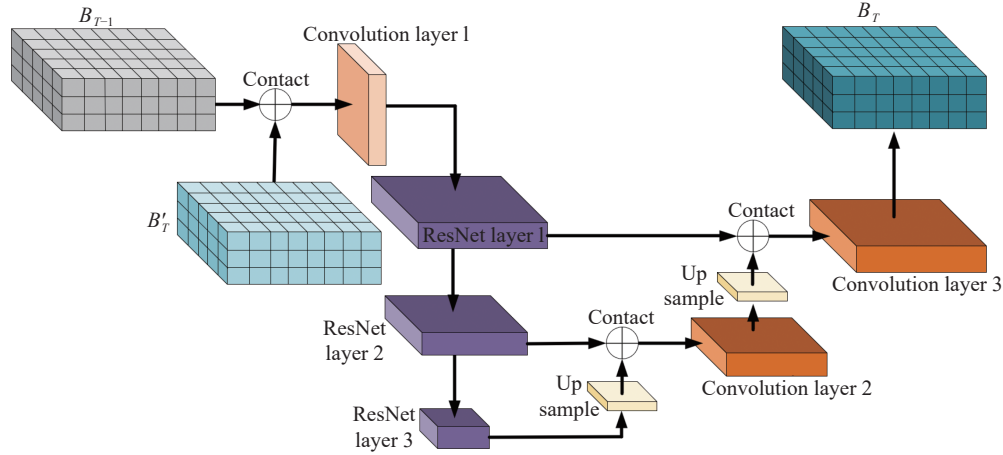


Fig. 5 Temporal fusion module.

tion from the template size. In this way, the network can identify the object category and also obtain the template size of the object based on prior information and then only fine-tune it based on this to return the actual size information. This not only reduces the difficulty of network regression but also makes the regression of size more accurate. For the detection of object orientation, with the positive direction of the y -axis of the agricultural machinery perception coordinate system shown in Fig. 3 set as 0° and the counterclockwise direction as positive, this paper divides the 360° orientation into 6 categories, each spanning 60° . Within each category, a more precise angle is regressed. On the basis of classification, further regression is carried out within the category to a more refined perspective. By adopting a mixed approach of classification on a large range and regression on a small range [13], more accurate orientation detection can be achieved. In order to save computational overhead, an anchor-free detection method is used throughout, where a fixed number of objects are detected in each frame of the scene. If the number of objects is insufficient, the scene is filled in by setting the background class.

3D object detection loss function: For category detection, we use the commonly used cross-entropy loss function, as shown in Eq. (3).

$$L_{cl}(y_i, \hat{y}_i) = \sum_{c=1}^C y_{ic} \log(\text{softmax}(\hat{y}_{ic})) \quad (3)$$

where y_i denotes the true value of the object, $\hat{y}_i$ denotes the detection value, and $C = 3$ denotes the number of object categories, including persons, cars, and background, respectively. For the loss of location and dimension is calculated as shown in Eq. (4).

$$L_{ld}(y_i, \hat{y}_i) = \sum_{l=1}^L (y_{il} - \hat{y}_{il})^2 + \sum_{d=1}^D (y_{id} - \hat{y}_{id})^2 + w_c \times L_{\text{corner}}(y_i, \hat{y}_i) \quad (4)$$

The mean squared error is used to supervise the position and size, where $L = 3$ represents the number of position dimensions and $D = 3$ represents the number of size dimensions. In addition to this, we introduce the corner loss L_{corner} [36] for overall supervision of the position and size, where $w_c = 0.5$

denotes the weight of the corner loss. Specifically, based on the detected 3D position coordinates and size information, the 8 corners of the object's 3D bounding box can be calculated. The distance deviation between the detected values and the ground truth values at the corresponding corner positions is used as the corner loss to supervise the object's position and size, as shown in Eq. (5).

$$L_{\text{corner}}(y_i, \hat{y}_i) = \min \left(\sum_{k=1}^K \|y_{ik} - y_{ik}^*\|, \sum_{k=1}^K \|y_{ik} - y_{ik}^{**}\| \right) \quad (5)$$

where $K = 8$ is the number of corners of the object's 3D bounding box, and y_i^* represents the corner information at the ground truth position of the object. In actual detection, there may be cases where the orientation recognition of the object is flipped, such as the rear and front of a car or the front and back of a person. In such situations, the detection of the object's position and size might still be accurate. Therefore, in order to avoid excessive position errors of the corners caused by the orientation-recognition flip, we flip the orientation of the real 3D bounding box to obtain the new corresponding corner positions y_i^{**} . The position errors of the corners for the two corresponding relationships are calculated, and the minimum value is taken as the final corner loss. The calculation of the orientation loss is shown in Eq. (6).

$$L_{\text{ori}}(y_i, \hat{y}_i) = \sum_{o=1}^O y_{io} \log(\text{softmax}(\hat{y}_{io})) + w_o \times (y_{icos} \times \hat{y}_{icos} + y_{isin} \times \hat{y}_{isin}) \quad (6)$$

The orientation loss is composed of classification error and angle error, where $O = 6$ represents the number of angle categories, and $w_o = 0.4$ denotes the weight of the angle error. The angle error is measured by the cosine of the angle deviation.

The anchor-free detection loss is found by first finding a set number of objects in each frame of the scene, such as persons, cars, and backgrounds. Then, Eq. (7) is used to find the best match with the ground truth label, and then the loss is calculated further.

$$\hat{\sigma} = \arg \min_{\sigma \in \theta_N} \sum_i^N \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) \quad (7)$$

where $N = 20$ is the number of detection objects in a scene frame, and the value of N is necessarily greater than the number of objects in a single frame within the dataset. For the ground truth labels, null values $\emptyset$ are used to fill in order to match the detection quantity N . The Hungarian matching algorithm is employed to obtain the lowest matching cost between the detection values y and the ground truth labels $\hat{y}$. Under the optimal matching, the total loss is calculated by the Eq. (8).

$$L_t(y, \hat{y}) = \sum_{i=1}^N [\alpha \times L_{cl}(y_i, \hat{y}_{\hat{\sigma}(i)}) + 1_{\{c_i \neq \Phi\}} (\beta \times L_{ld}(y_i, \hat{y}_{\hat{\sigma}(i)}) + L_{ori}(y_i, \hat{y}_{\hat{\sigma}(i)}))] \quad (8)$$

where $\alpha = 0.5$ and $\beta = 0.6$ denote the weights of different losses, $\hat{\sigma}(i)$ denotes the best matching relationship of each object, combined with the category loss, and the positional dimensions as well as the orientation loss are computed for non-background objects to compose the total loss.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental Setup

The experimental platform is Ubuntu 20.04.3 Long Term Support, equipped with an NVIDIA GeForce RTX 3090 and 64 GB of memory. It is installed with the CUDA 11.1 development environment. The CPU model is Intel (R) Xeon (R) CPU E5-2696 v3, with 18 high-performance cores. We use the mainstream Python 3.8.13 and PyTorch 2.4.1 as the programming language and deep learning framework, respectively. We set the model training with a batch size of 12, a learning rate of 0.001, and a weight decay of 1×10^{-7} . After 100 iterations of training on the platform, convergence is achieved. We conduct performance evaluation experiments on the TractorRoadBEV using a self-built dataset of tractor roads.

B. Ablation Experiments

We conducted ablation studies on three key modules in the model: the center mask, temporal fusion, and corner loss. Referencing the KITTI standard, an IoU > 0.7 for cars and an IoU > 0.5 for pedestrians are considered the correct identification. Table 2 displays the results on the validation set.

Table 2 demonstrates how each strategy contributes to the model's enhanced performance in 3D object detection. With the incorporation of each strategy, there is a notable increase in the F1 score. The center mask strategy improved the model's precision and recall by 1.14% and 1.37%, respectively. The addition of the center mask helped the model to initially pinpoint the object area and made a rough distinction between

noises, which in turn better supported the subsequent detection tasks. The temporal fusion module further enhanced the model's precision and recall by 0.69% and 0.68%, respectively, indicating that the temporal fusion module also effectively established the relationship between consecutive frames to assist in detection. The introduction of the corner loss increased the model's precision and recall by 0.41% and 2.46%, respectively. Since unmatched boxes cannot compute the corner loss, the supervision of the corner loss does not significantly improve the detection of false positives, and the precision improvement is limited. However, for matched objects, the corner loss can better supervise the 3D bounding boxes to align with real-world conditions, which significantly improves the recall rate.

It is noteworthy that, as observed in Table 2, the temporal fusion module offers a very limited improvement to the model's performance. We believe that the temporal fusion module is responsible for establishing the relationship between consecutive frames, using the results identified in the previous frame to aid the detection in the current frame. Our analysis of the dataset used in this paper indicates that if an object is detected in the previous frame, it is highly probable that it will also be detected in the next frame; conversely, if it is not detected in the previous frame, it is likely to remain undetected in the subsequent frame. During the temporal process, the temporal fusion module can correct some instances where objects are detected in the previous frame but not in the next; however, the proportion of such data is not high. Therefore, the limited enhancement in precision and recall by this module may be constrained by the nature of the data itself. We compare the reasoning results to the validation set by ablating the temporal fusion module, and the visualization of these results is presented in Fig. 6.

Fig. 6 shows the model's detection of a car and a person in consecutive frames T and $T + 1$. For the person in the scene, without the temporal fusion module, the model has detected the person in frame T but failed to detect them in frame $T + 1$, possibly due to the similarity in color between the person's clothing and the background. The inclusion of the temporal fusion module corrected this issue. In addition to accuracy, the task of 3D detection should also focus on the stability of object detection; that is, the changes in the 3D bounding box detection results for the same object across a sequence of continuous frames should exhibit continuity. The temporal fusion module, which establishes the relationship between consecutive frames, should help to enhance this stability. To validate this hypothesis, we selected a sequence of frames over a period of time and conducted a stability test on the size

Table 2 Comparison of validation set ablation (IoU_{car} > 0.7 and IoU_{person} > 0.5).

Model	Center mask	Temporal fusion	Corner loss	Precision (%)	Recall (%)	F1 score
TractorRoadBEV	—	—	—	85.81	85.11	0.8546
	√	—	—	86.95	86.48	0.8671
	√	√	—	87.64	87.16	0.8740
	√	√	√	88.05	89.62	0.8883

Note: √ means that the relevant policy is used, and — means that the relevant policy is not used.

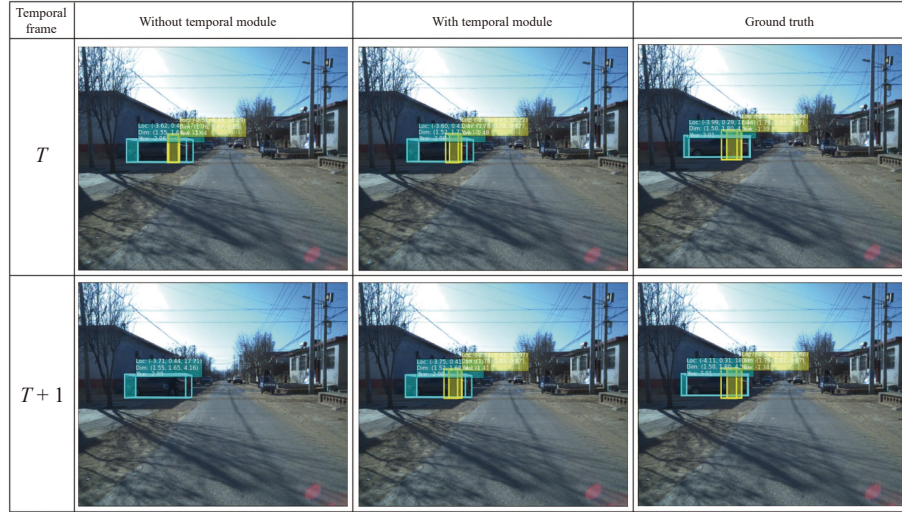


Fig. 6 Temporal fusion module visualization comparison chart.

of a detected car. Unlike persons, cars, as rigid objects, have a fixed size that does not change over time in real-world scenarios. Therefore, the detected size of the car in the model's detections should not vary significantly. We calculated the average absolute deviation in the size of the detection results for this object over the selected time period. The experimental results are shown in Table 3.

Table 3 Comparison of dimensional average absolute deviation.

Temporal fusion module	Length (m)	Width (m)	Height (m)
Without	0.122152	0.023828	0.02084
With	0.072693	0.003064	0.00847

The average absolute deviation in the size of the 3D bounding box is a good measure of the stability of object detection size. As observed in Table 3, the temporal fusion module significantly improves the size stability of object detection. The average absolute deviations in length, width, and height are all significantly lower than those of the model without the temporal fusion module, with reductions of 0.049459, 0.020764, and 0.01237 m, respectively. This further demonstrates the effectiveness of the temporal fusion module in establishing relationships between consecutive frames to assist the model. In this paper, we extracted 10 frames at fixed time intervals, and the visualization results are shown in Fig. 7. It can be intuitively observed that the addition of the temporal fusion module improves the stability of the detected object size.

C. Comparative Experiments

We compare our multi-view fusion approach against Deep3DBox, a classic two-stage monocular detector designed

for single forward-facing cameras. It first detects 2D objects then regresses 3D attributes, a paradigm that performs well on our dataset. We select it for two reasons: (1) its multi-stage philosophy aligns with ours, enabling a meaningful comparison; and (2) its modular design allows us to integrate the same 2D detector YOLOv10 and object priors, and we adopt its best-performing backbone MobileNetV3 [34, 35] for a fair benchmark. To handle multi-view input, we apply Deep3DBox independently to each view and fuse results in 3D space using camera extrinsics, a standard extension that exposes its limitations in cross-view truncation and fusion. In contrast, TractorRoadBEV fuses features within the network, avoiding redundant computation and reducing false positives in overlapping views—a key improvement. The comparison results of the two models on the test set are shown in Table 4.

As shown in Table 4, the TractorRoadBEV proposed in this paper achieves a significantly higher F1 score for object detection on the tractor road test dataset compared to the classic algorithm Deep3DBox. TractorRoadBEV also exhibits overall better performance than Deep3DBox in the detection of both persons and cars, with F1 scores that are 0.0148 and 0.0134 higher for the two categories, respectively. We also note that, whether considering the overall object detection or the individual categories of persons and cars, TractorRoadBEV shows a slightly lower recall rate compared to Deep3DBox, while significantly outperforming Deep3DBox in precision. The recall rates of TractorRoadBEV for persons and cars are 0.47% and 0.64% lower than those of Deep3DBox, respectively, while the precision rates are 3.46% and 3.39% higher. We analyzed that for the monocular detection algorithm Deep3DBox to detect multi-view images, our strategy was to test each camera separately and then integrate

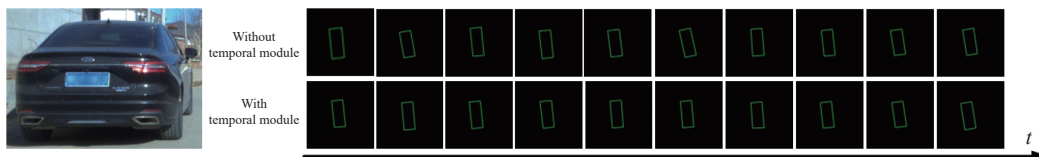
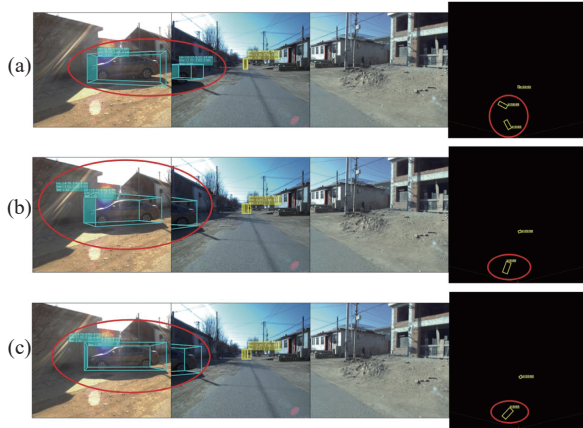


Fig. 7 Dimensional stability comparison.

Table 4 Comparison of the results of the 3D frame detection of the object on the test set.

Model	Persons (IoU > 0.5)			Cars (IoU > 0.7)			Average		
	Precision (%)	Recall (%)	F1 score	Precision (%)	Recall (%)	F1 score	Precision (%)	Recall (%)	F1 score
Deep3DBox	80.93	82.27	0.8159	80.06	80.58	0.8032	80.57	81.56	0.8106
TractorRoadBEV	84.39	81.80	0.8307	83.45	79.94	0.8166	83.99	81.01	0.8247

the results into the same space using the camera's extrinsic parameters. Monocular detection of objects at the image edges, due to truncation and other factors, inherently has some instability, and when combined with systematic errors of the camera's extrinsic parameters, Deep3DBox easily misidentifies these objects, leading to many false positives. In contrast, TractorRoadBEV effectively fuses image features from different viewpoints within the network, greatly reducing the number of false positives. Additionally, the time required for multi-view detection with a monocular algorithm increases linearly with the number of images. In our comparison, the average processing time for TractorRoadBEV is 84.9 ms, while Deep3DBox takes 264.7 ms, which is significantly longer than the method proposed in this paper. We have selected some visualized comparison results, as shown in Fig. 8.

**Fig. 8** Comparison of 3D inspection visualization of test set models.

Figs. 8(a)–8(c) display the detection results of Deep3DBox, show the detection results of our proposed model, TractorRoadBEV, and provide the annotated ground truth. It can be seen that for the car in the scene, which is in the overlapping field of view of the left front and front cameras, Deep3DBox results in a false detection. In contrast, TractorRoadBEV effectively integrates the object across the overlapping view-points. For the distance and orientation errors of the car detection in the image, Deep3DBox has errors of 1.05 m and 1.244 rad, respectively, while TractorRoadBEV has errors of only 0.05 m and 0.400 rad, respectively.

For the orientation error of the person in the front view, Deep3DBox has an error of 0.460 rad, whereas TractorRoadBEV has an error of 0.200 rad, clearly illustrating the superiority of TractorRoadBEV.

D. Extended Experiments

Leveraging its unified BEV representation, our model readily extends to other tasks. We demonstrate this by adapting it

for BEV segmentation and comparing against LSS, a foundational bottom-up BEV method that serves as the *de facto* baseline for multi-camera BEV segmentation. LSS excels on massive datasets but its frame-by-frame lifting makes it noise-sensitive in small-data scenarios. By replacing only the detection head, TractorRoadBEV (with mask-enhanced and temporal fusion) achieves superior segmentation, validating its effectiveness beyond object detection and its robustness to limited agricultural data. TractorRoadBEV uses the same detection head and loss function as LSS. The comparison results on the test set are shown in Table 5.

Table 5 Comparison of test set scenario segmentation results.

Model	IoU
LSS	0.461
TractorRoadBEV	0.602

Table 5 shows that the IoU of the TractorRoadBEV proposed in this paper is 0.151 higher than that of LSS in the scene segmentation of the tractor road test set. This further demonstrates the superiority of the TractorRoadBEV network backbone in recognizing object features, as well as its good expandability. Some visualized comparison results are shown in Fig. 9.

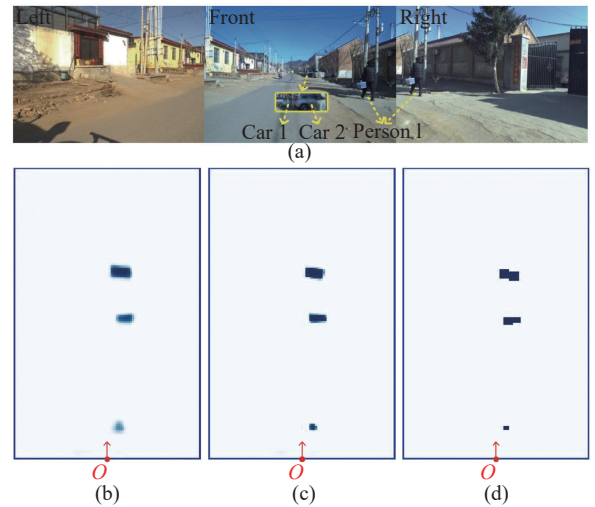
**Fig. 9** Comparison of test set model scenario segmentation visualization.

Fig. 9(a) shows the multi-view images input into the model, while Figs. 9(b)–9(d) represent the segmentation results of LSS, TractorRoadBEV, and the ground truth annotations, respectively. In the segmentation maps, the origin point O denotes the location of the agricultural machinery, the arrow indicates the direction of car travel, and the blue area represents the object's location. From the segmentation results, it

can be seen that TractorRoadBEV is more accurate than LSS in segmenting both a nearby person and distant cars, and its results are closer to the ground truth.

V. CONCLUSIONS

To address the limited research on multi-view camera fusion for 3D object detection in agricultural tractor road scenes, we propose a lightweight multi-view perception framework named TractorRoadBEV. The main contributions are summarized as follows.

(1) We propose a multi-view BEV perception model for small-scale agricultural datasets. By introducing a central mask strategy, the model focuses more effectively on object regions and suppresses background noise. We further design a temporal fusion module to capture correlations between consecutive frames, improving both detection stability and overall performance. Additionally, we design a loss function incorporating corner loss, which effectively mitigates the issue of missed detections. Experimental results show that TractorRoadBEV achieves precision rates of 84.39% and 83.45% for person and vehicle detections, with F1 scores of 0.8307 and 0.8166, respectively, on the tractor road test set.

(2) The proposed framework demonstrates good extensibility for different perception tasks. By replacing the detection head, TractorRoadBEV can be adapted to scene segmentation tasks, achieving an IoU of 0.602, which is significantly higher than that of the classical LSS method.

(3) We construct a multi-view tractor road dataset containing diverse scene information to support future research in agricultural autonomous driving. In future work, the dataset will be further expanded to include additional environmental conditions such as lighting variations and seasonal changes. Moreover, we will incorporate inertial measurement unit (IMU) information to spatially align historical feature maps before temporal fusion, which may further improve the performance of the temporal fusion module.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (No. 2023YFE0208200).

REFERENCES

- [1] L. Zhang, G. Liu, Y. Qi, T. Yang, and C. Jin, Research progress on key technologies of agricultural machinery unmanned driving system, *J. Intell. Agric. Mech.*, 2022, 3(1&2), 27–36.
- [2] P. Wang and Y. Yang, Development status and future trends of agricultural machinery automatic driving technology, *Agric. Mach. Usage Maint.*, 2024(11), 64–67.
- [3] M. S. Hossain, M. Rahman, A. Rahman, M. M. Kabir, M. F. Mridha, J. Huang, and J. Shin, Automatic navigation and self-driving technology in agricultural machinery: A state-of-the-art systematic review, *IEEE Access*, 2025, 13, 94370–94401.
- [4] W. Xu and W. Li, Environment perception and location technology of driverless vehicles, *Auto Sci-tech.*, 2021, 6, 52–60, (in Chinese).
- [5] H. Zhao, X. Li, C. Xu, B. Xu, and H. Liu, A survey of automatic driving environment perception, in *Proc. 2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion*, Cambridge, UK, 2024, 1038–1047.
- [6] G. P. Stein, O. Mano, and A. Shashua, Vision-based ACC with a single camera: Bounds on range and range rate accuracy, in *Proc. IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)*, Columbus, OH, USA, 2003, 120–125.
- [7] S. Tuohy, D. O’Cualain, E. Jones, and M. Glavin, Distance determination for an automobile environment using inverse perspective mapping in OpenCV, in *Proc. IET Irish Signals and Systems Conference (ISSC 2010)*, Cork, Ireland, 2010, 100–105.
- [8] K.-Y. Park and S.-Y. Hwang, Robust range estimation with a monocular camera for vision-based forward collision warning system, *Sci. World J.*, 2014, 2014, 923632.
- [9] Z. Liu, Z. Wu, and R. Tóth, SMOKE: Single-stage monocular 3D object detection via keypoint estimation, in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, WA, USA, 2020, 4289–4298.
- [10] T. Wang, X. Zhu, J. Pang, and D. Lin, FCOS3D: Fully convolutional one-stage monocular 3D object detection, in *Proc. IEEE/CVF International Conference on Computer Vision Workshops*, Montreal, Canada, 2021, 913–922.
- [11] P. Liao, F. Yang, D. Wu, and L. Bo, MonoDETRNext: Next-generation accurate and efficient monocular 3D object detection method, arXiv preprint arXiv: 2405.15176, 2024.
- [12] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, Monocular 3D object detection for autonomous driving, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, 2147–2156.
- [13] A. Mousavian, D. Anguelov, J. Flynn, and J. Koščeká, 3D bounding box estimation using deep learning and geometry, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, 5632–5640.
- [14] H. Yu and J. Oh, Anchor distance for 3D multi-object distance estimation from 2D single shot, *IEEE Robot. Autom. Lett.*, 2021, 6(2), 3405–3412.
- [15] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, NuScenes: A multimodal dataset for autonomous driving, in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, 11618–11628.
- [16] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine et al., Scalability in perception for autonomous driving: Waymo open dataset, in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, 2443–2451.
- [17] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu et al., MMDetection: Open MMLab detection toolbox and benchmark, arXiv preprint arXiv: 1906.07155, 2019.
- [18] J. Philion and S. Fidler, Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D, in *Proc. 16th European Conference on Computer Vision—ECCV 2020*, Glasgow, UK, 2020, 194–210.
- [19] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, BEVDet: High-performance multi-camera 3D object detection in bird-eye-view, arXiv preprint arXiv: 2112.11790, 2021.
- [20] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, DETR3D: 3D object detection from multi-view images via 3D-to-2D queries, in *Proc. 5th Conference on Robot Learning*, London, UK, 2022, 180–191.
- [21] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, BEVFormer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers, in *Proc. 17th European Conference on Computer Vision—ECCV 2022*, Tel Aviv, Israel, 2022, 1–18.
- [22] R. Qian, X. Lai, and X. Li, 3D object detection for autonomous driving: A survey, *Pattern Recognit.*, 2022, 130, 108796.
- [23] Y. Ma, T. Wang, X. Bai, H. Yang, Y. Hou, Y. Wang, Y. Qiao, R. Yang, and X. Zhu, Vision-centric BEV perception: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024, 46(12), 10978–10997.
- [24] A. Geiger, P. Lenz, and R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in *Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012, 3354–3361.

- [25] T. Wang, X. Zhu, J. Pang, and D. Lin, Probabilistic and geometric depth: Detecting objects in perspective, in *Proc. 5th Conference on Robot Learning*, London, UK, 2022, 1475–1485.
- [26] P. Li, X. Chen, and S. Shen, Stereo R-CNN based 3D object detection for autonomous driving, in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, 7636–7644.
- [27] Z. Zhou, L. Du, X. Ye, Z. Zou, X. Tan, L. Zhang, X. Xue, and J. Feng, SGM3D: Stereo guided monocular 3D object detection, *IEEE Robot. Autom. Lett.*, 2022, 7(4), 10478–10485.
- [28] R. Zhang, H. Qiu, T. Wang, Z. Guo, Z. Cui, Y. Qiao, H. Li, and P. Gao, MonoDETR: Depth-guided transformer for monocular 3D object detection, in *Proc. 2023 IEEE/CVF International Conference on Computer Vision*, Paris, France, 2023, 9121–9132.
- [29] J. Kim, S. Moon, S. Nahl, and D. H. Shim, MonoDINO-DETR: Depth-enhanced monocular 3D object detection using a vision foundation model, in *Proc. 2025 IEEE Intelligent Vehicles Symposium*, Cluj-Napoca, Romania, 2025, 1386–1393.
- [30] J. Huang and G. Huang, BEVDet4D: Exploit temporal cues in multi-camera 3D object detection, arXiv preprint arXiv: 2203.17054, 2022.
- [31] Z. Xue, M. Guo, H. Fan, S. Zhang, and Z. Zhang, CorrBEV: Multi-view 3D object detection by correlation learning with multi-modal prototypes, in *Proc. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2025, 27413–27423.
- [32] G. Yan, Z. Liu, C. Wang, C. Shi, P. Wei, X. Cai, T. Ma, Z. Liu, Z. Zhong, Y. Liu et al., OpenCalib: A multi-sensor calibration toolbox for autonomous driving, *Softw. Impacts*, 2022, 14, 100393.
- [33] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, YOLOv10: Real-time end-to-end object detection, in *Proc. 38th International Conference on Neural Information Processing Systems*, Vancouver, Canada, 2024, 3429.
- [34] M. Tan and Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in *Proc. 36th International Conference on Machine Learning*, Long Beach, CA, USA, 2019, 6105–6114.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, 770–778.
- [36] C. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, Frustum PointNets for 3D object detection from RGB-D data, in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, 918–927.



Lili Yang received the PhD degree from China Agricultural University, Beijing, China, in 2009. She currently is an associate professor at College of Information and Electrical Engineering, China Agricultural University, China. Her research interests include intelligent environment perception for unmanned driving agricultural machinery and big data analysis of agricultural operations.



Mengshuai Chang received the BE degree from Liaoning Technical University, Fuxin, China, in 2022. He is currently pursuing the ME degree in computer science and technology at College of Information and Electrical Engineering, China Agricultural University, China. His research interests include 3D object detection and tracking in computer vision.



Xiao Guo received the BE degree from China Agricultural University, Beijing, China, in 2022. He is currently pursuing the ME degree at College of Information and Electrical Engineering, China Agricultural University, China. His research interests include deep learning and the perception system of agricultural machinery autonomous driving in agricultural scenarios.



Jiale Li received the ME degree from China Agricultural University, Beijing, China, in 2024. He is currently pursuing the PhD degree in computer science and technology at College of Information and Electrical Engineering, China Agricultural University, China. His research interests include intelligent environment perception for unmanned driving agricultural machinery.