Chapter 10
Bird's Eye View Perception for Autonomous Driving

Jiayuan Du, Shuai Su, Rui Fan, and Qijun Chen

Abstract Bird's eye view (BEV) perception refers to transforming a perspective view into a bird's eye view and performing perception tasks such as 3D detection, map segmentation, and motion prediction. Due to its inherent advantages in representing 3D space, fusing multi-modal data, facilitating decision making, and aiding in path planning, BEV perception has garnered significant attention from both academia and industry. In this chapter, we present an overview of BEV perception, covering its definition, categorization, benefits, and mathematical foundations. We then provide a comprehensive review of the current state-of-the-art research, datasets, evaluation metrics, and industrial applications. In the end, we outline several existing challenges and present our own conclusions regarding BEV perception.

10.1 Introduction

Bird's eye view (BEV) perception involves learning representations in the bird's eye view space and performing perception tasks, such as 3D detection [1], map segmentation [2], and motion prediction [3]. With the sensor settings of autonomous vehicles becoming increasingly diverse and complex, there is a great demand for a unified representation. Thanks to its inherent advantages in 3D space representation,

J. Du · S. Su · R. Fan · Q. Chen (✉)
The Robotics and Artificial Intelligence Laboratory (RAIL), The Department of Control Science and Engineering, Frontiers Science Center for Intelligent Autonomous Systems, and State Key Laboratory of Intelligent Autonomous Systems, Tongji University, Shanghai 201804, People's Republic of China
e-mail: qjchen@tongji.edu.cn

J. Du
e-mail: dujiayuan@tongji.edu.cn

S. Su
e-mail: sushuai@tongji.edu.cn

R. Fan
e-mail: rfan@tongji.edu.cn

R. Fan et al. (eds.), *Autonomous Driving Perception*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-981-99-4287-9_10

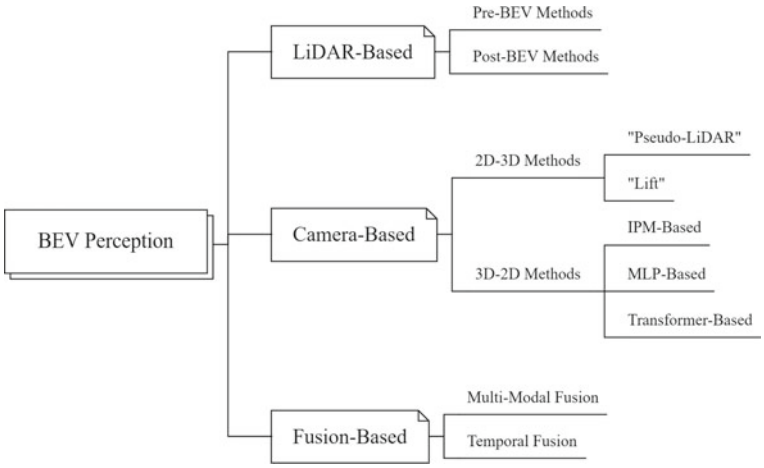

Fig. 10.1 Taxonomy of BEV perception methods

multi-modal fusion, decision making, and path planning, BEV perception has been attracting more and more attention from both academia and industry.

According to the different modalities of input data, a taxonomy of BEV perception methods can be yielded as illustrated in Fig. 10.1: LiDAR-based methods, camera-based methods, and fusion-based methods. Conventional LiDAR-based methods with their BEV-like point clouds have shown great success in 3D detection and semantic segmentation tasks [4]. However, perception using only perspective view or multi-view (vision-centric methods) remains a significant challenge.

In modern autonomous driving, decision-making and motion planning modules depend on multiple perception and prediction modules to gather sufficient environmental information. The perception task not only detects dynamic objects but also identifies static elements such as road boundaries, crosswalks, lane lines, and road signs. Meanwhile, the prediction task requires the system to deduce the motion trend of other dynamic objects, providing a basis for decision-making and path planning to avoid collisions.

Currently, in the industry, research on perception and prediction algorithms based solely on vision typically only focuses on a single sub-problem in the overall process. For example, researchers may focus on 3D object detection, map segmentation, object tracking, or motion prediction, and then fuse the perception results of different networks through pre-fusion or post-fusion methods. While this approach enables problem decomposition and facilitates independent academic research, it results in multiple submodules stacked in a linear structure when building the overall system. However, this serial architecture has several significant drawbacks:

1. The errors of upstream modules are continuously transmitted downstream, which can significantly affect the performance of downstream tasks when sub-problems

are studied independently. In these cases, the ground truth is typically used as input, which can result in cumulative errors impacting downstream performance.
2. In a serial architecture, repeated operations such as feature extraction and dimension conversion in Convolution Neural Networks (CNNs) can lead to redundant calculations, which are not conducive to improving the overall efficiency of the system.
3. The temporal information cannot be fully utilized. On the one hand, the temporal information can be leveraged as a supplement to the spatial information to better detect the blocked object at the current moment and provide more reference information for the location of the object. On the other hand, temporal information can help judge the motion state of the object. In the absence of temporal information, the method based on pure vision can hardly effectively judge the motion speed of the object.

Different from traditional serial architecture, the BEV scheme utilizes multiple sensors (such as camera, LiDAR, RADAR, IMU, GPS) and convert multi-modal information to a unified BEV space for various downstream perception tasks. Such a scheme can provide a unified representation space for autonomous driving perception [5] and can accomplish multiple perception tasks in parallel. Specifically, the advantages of BEV perception are reflected in the following aspects:

1. Intuitive and friendly for subsequent modules. BEV perception provides a holistic view of the surrounding environment, enabling the system to detect obstacles and road elements that might not be visible from the vehicle's current position, which is beneficial to downstream prediction and planning modules.
2. Fusion-friendly.
    a. BEV perception provides a consistent coordinate system for all sensors, simplifying the fusion of data from multiple sources and enabling better object association and tracking.
    b. Temporal fusion is easier to realize. In the BEV space, temporal information can be easily fused via ego-motion of the autonomous vehicle.
3. Easier end-to-end optimization. In traditional perception tasks, recognition, tracking, and prediction operate more like a "serial system", where errors in the upstream of the system are propagated downstream, resulting in error accumulation. However, in the BEV space, perception and prediction occur in a unified space, allowing for direct end-to-end optimization through neural networks and generating "parallel" results, thereby avoiding error accumulation. Additionally, this approach can significantly reduce the impact of algorithm logic, enabling the perceptual network to learn from data in a self-driven manner and achieve better functional iterations.

This chapter is organized as follows: Sect. 10.1 is an overall introduction of BEV perception, including conception, taxonomy and advantages. Section 10.2 introduces mathematical fundamentals of inverse perspective mapping, which is of vital importance in BEV perception. Sections 10.3, 10.4 and 10.5 introduces LiDAR-

based, camera-based and fusion-based methodologies of BEV perception, respectively. Section 10.8 shows nowadays industrial applications of BEV perception. In Sects. 10.9 and 10.10 we discuss the existing challenges and Future Development of BEV perception.

10.2 BEV Fundamentals

Generally, cameras mounted on a vehicle suffer from a significant perspective effect [6–8], as illustrated in Fig. 10.2a. This perspective effect can result in the driver's inability to accurately perceive distance, making advanced image processing or analysis challenging. As a result, it is necessary to generate a bird's eye view using perspective transformation., as shown in Fig. 10.2 [9].

Perspective mapping can transform points in 3D space to the image space, while the inverse problem of projecting image pixels back to 3D space is considered an ill-posed problem. To address this challenge, the pioneer work of Inverse Perspective Mapping (IPM) [10] introduced a constraint that the inversely mapped points should lie on the ground plane. This constraint allows the mapping to be described via a homography matrix, enabling the solution of the ill-posed problem. The geometry of perspective mapping and inverse transformation is shown in Fig. 10.3. Mathematically, IPM is a linear mapping in homogeneous coordinates.

10.2.1 Perspective Mapping

For an ideal pinhole camera model, we state two definitions for camera coordinate system $\boldsymbol{C} = \{\mathbf{x_C}, \mathbf{y_C}, \mathbf{z_C}\}$ and an image plane $\boldsymbol{I} = \{\mathbf{u}, \mathbf{v}\}$ at distance f (focal length) from the origin parallel to $\mathbf{x_C}$ and $\mathbf{y_C}$. A perspective mapping is described by $\mathscr{P}_B : \mathbb{R}^3 \mapsto \mathbb{R}^2$:

$$\mathbf{p} \mapsto \mathbf{p}' = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{-f}{(\mathbf{p} \cdot \mathbf{z_C})} \cdot \begin{bmatrix} (\mathbf{p} \cdot \mathbf{x_C}) \\ (\mathbf{p} \cdot \mathbf{y_C}) \end{bmatrix}, \tag{10.1}$$

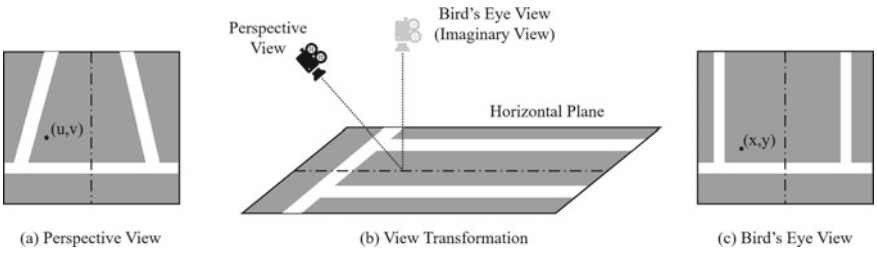

Fig. 10.2 Illustration of perspective transformation in a parking lot scene [9]

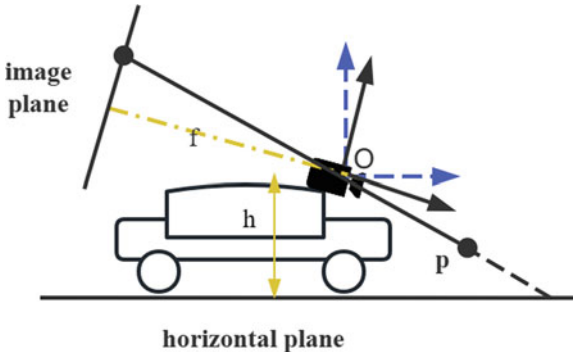

Fig. 10.3 Geometry of inverse perspective mapping [10]. Nodal point O is the center of projection and the common origin of the camera frame and the world frame. Solid black arrow $\mathbf{z_C}$, $\mathbf{y_C}$ axes of the camera coordinate system, and the Z-axis is in the same direction as the camera. h is the height of the nodal point O and f is the focal length of the camera. Dotted blue arrow $\mathbf{y_W}$, $\mathbf{z_W}$ axes of the world coordinate system, the Y-axis is in the same direction as the vehicle. The horizontal axes $\mathbf{x_C}$ and $\mathbf{x_W}$ are perpendicular to the paper plane. $\mathbf{p}$ is a point in 3D space; $\mathbf{p'_I}$, $\mathbf{p'_H}$ denote its projections into the image and the horizontal plane; $\tilde{\mathbf{p}}$ means homogeneous representation of $\mathbf{p'}$

where $(\cdot)$ denotes the inner product. All 3D points on the ray projected to $p' = [u, v]^\top$ is given by $\tilde{\mathbf{p}}$:

$$\mathbf{p'} \mapsto \tilde{\mathbf{p}} = \begin{bmatrix} \lambda u \\ \lambda v \\ -\lambda f \end{bmatrix}, \quad for \quad \lambda \in \mathbb{R}, \tag{10.2}$$

where $\tilde{\mathbf{p}}$ is the homogeneous representation of $\mathbf{p'}$ in the coordinate frame $\boldsymbol{C}$. Substitute (10.2) into (10.1), we have $\mathscr{P}(\tilde{\mathbf{p}}) = \mathbf{p'}$ for all $\lambda \neq 0$.

10.2.2 Inverse Perspective Mapping

Denote a world coordinate system as $\boldsymbol{W} = \{\mathbf{x_W}, \mathbf{y_W}, \mathbf{z_W}\}$. The horizontal plane is spanned by $\mathbf{x_W}$ and $\mathbf{y_W}$ and the upward direction ia pointed by $\mathbf{z_W}$. Suppose there is a center of projection O at height h from the horizontal plane and its focal length is f from the image plane. By assuming camera frame and world frame share the common origin O, then the coordinate transformation from the camera to the world is described by an orthogonal matrix $\boldsymbol{Q}$. In (10.3), $\boldsymbol{Q}$ is represented by the column vectors $\mathbf{x'}$, $\mathbf{y'}$ and $\mathbf{z'}$, the matrix $\boldsymbol{Q}$ is also known as the extrinsic matrix.

Under the assumption that the back-projected points fall on the ground, IPM is actually finding the correspondence between a point $\mathbf{p'_I}$ in the image plane and a point $\mathbf{p'_H}$ in the horizontal plane. Here, the horizontal ground plane is spanned by $\mathbf{x_W}$, $\mathbf{y_W}$ in the world frame. In homogeneous coordinates, this is a linear mapping of a point $\tilde{\mathbf{p}}_\mathbf{I}$ to a point $\tilde{\mathbf{p}}_\mathbf{H}$, characterized by the orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$:

$$\tilde{\mathcal{Q}} : \mathbb{R}^3 \mapsto \mathbb{R}^3; \quad \tilde{\mathbf{p}}_{\mathbf{H}} = Q \cdot \tilde{\mathbf{p}}_{\mathbf{I}}$$
$$\begin{bmatrix} \mu X \\ \mu Y \\ -\mu h \end{bmatrix} = \begin{bmatrix} x_1' & y_1' & z_1' \\ x_2' & y_2' & z_2' \\ x_3' & y_3' & z_3' \end{bmatrix} \cdot \begin{bmatrix} \lambda u \\ \lambda v \\ -\lambda f \end{bmatrix}. \tag{10.3}$$

Here, the expression of camera frame axes $\mathbf{x_C}, \mathbf{y_C}, \mathbf{z_C}$ in the world frame is denoted by the matrix $\boldsymbol{Q}$, which is composed out of $x_1', ..., z_3'$. Then it projects $\tilde{\mathbf{p}}_{\mathbf{H}}$ onto the horizontal plane using (10.1) and yields the inverse perspective mapping $\mathcal{Q}$:

$$\mathcal{Q} : \mathbb{R}^2 \mapsto \mathbb{R}^2; \quad \mathbf{p}_{\mathbf{H}}' = \mathcal{Q}(\mathbf{p}_{\mathbf{I}})$$
$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{-h}{x_3' u + y_3' v - z_3' f} \cdot \begin{bmatrix} x_1' u + y_1' v - z_1' f \\ x_2' u + y_2' v - z_2' f \end{bmatrix}. \tag{10.4}$$

10.3 LiDAR-Based BEV Perception

LiDAR sensors provide depth information, which can lead to better object recognition performance compared to visual-based sensors. However, LiDAR lacks semantic information such as texture or color, and the point clouds may be sparse in distant areas where information is missing. The general pipeline of LiDAR-based BEV perception is illustrated in Fig. 10.4. It takes point cloud as input and performs feature extraction and view transformation to construct BEV feature maps. Common detection heads are used to generate 3D prediction results. LiDAR-based methods can be classified into pre-BEV methods and post-BEV methods based on the order of feature extraction and BEV transformation, as shown in Fig. 10.4.

10.3.1 Pre-BEV Methods

Pre-BEV methods extract features before the BEV transformation using either point or voxel representations. Point-based methods process the raw LiDAR point cloud, which has significant consumption of computing and storage resources. In contrast, voxel-based methods voxelize the point cloud into discrete grids by discretizing the continuous 3D coordinate, providing a more efficient representation. To extract

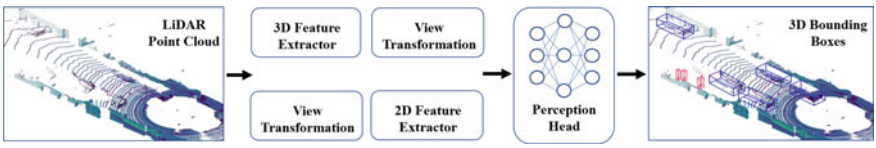

Fig. 10.4 General pipeline of LiDAR-based BEV perception [11]

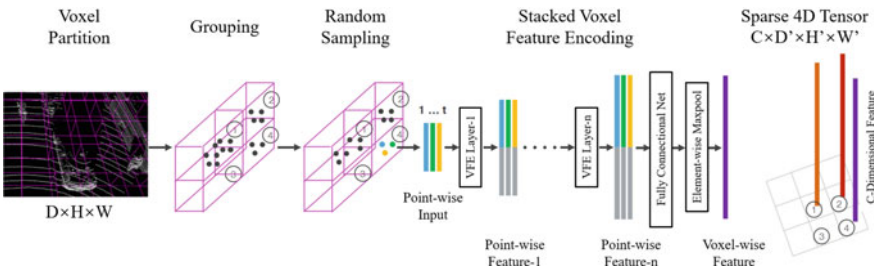

Fig. 10.5 Feature extraction of VoxelNet architecture [14]

point cloud features from the voxel, 3D convolution or 3D sparse convolution [12, 13] are commonly utilized. The majority of cutting-edge techniques typically undertake feature extraction using 3D sparse convolution. The height axis is then densified and compressed to form at the 3D voxel characteristics as a 2D tensor in BEV space.

As shown in Fig. 10.5. VoxelNet [14] firstly partitions the 3D space into voxels, generates voxel-wise features by maxpooling of all point-wise features, and represents the space as a sparse 4D tensor. Then, a 3D convolution is applied to aggregate spatial context in the tensor. Finally, it transforms the tensor into BEV implicitly, and generates the 3D bounding boxes with a region proposal network (RPN). SECOND [15] introduces sparse convolution in the processing of voxel representation, which greatly accelerate the training and inference.

Comparing with anchor-based bounding box, center-based representation has several advantages:

- Points have no intrinsic direction. Therefore, the search space of the target detector is reduced, which allowing backbone to learn the rotation invariance of objects and their variance with respect to rotation.
- It is also helpful to simplify downstream tasks such as tracking.A point object's trajectory is its path across space and time. The network can then integrate the continuous frames and forecast the relative offset between them with ease.
- Point-based feature extraction facilitates researchers to design a faster and more effective two-stage refinement module.

As the representative work, CenterPoint [16] becomes a baseline method for 3D detection due to its excellent design of two-stage center-based detector (Fig. 10.6).

To learn more discriminative features from LiDAR point cloud, PV-RCNN [17] combines voxel and point branches to generate 3D proposals and refinement respectively. To help the backbone network learn structure-aware features, SA-SSD [18] designs an auxiliary network that combines the foreground segmentation and center estimation tasks. Voxel Region of Interest (RoI) Pooling is adopted by Voxel R-CNN [19] to aggregate construction data from voxel-wise features, and it achieves comparable accuracy to point-based approaches. With the concept of dynamic graph, Object DGCNN [20] remodels the detection problem as information transmission process in the BEV space. To aggregate large 3D context, VoTr [21] introduces an attention

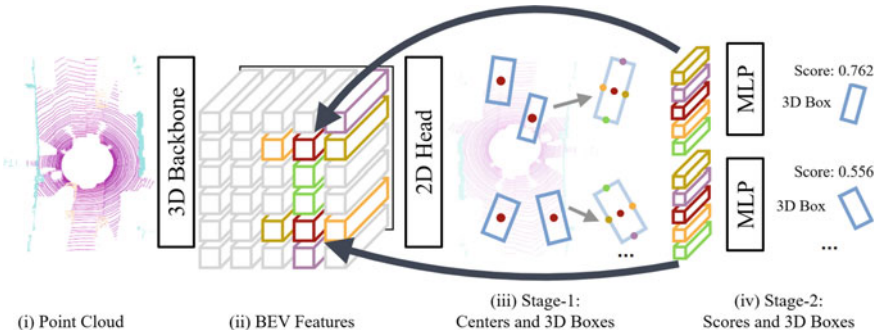

Fig. 10.6 Overview of CenterPoint framework [16]

mechanism on numerous voxels. SST [22] follows the idea of shifting windows in Swin Transformer [23], and divides the voxelized point cloud space into windows. It applies sparse regional attention and window shifting to avoid information loss by downsampling. AFDetV2 [24] introduces a keypoint supervision as auxiliary task and adopts a multi-task head to build a single-stage anchor-free network.

10.3.2 Post-BEV Methods

Another technical route of LiDAR-based BEV perception is to transform the LiDAR point cloud into BEV space first, and then carry out 2D feature extraction.

In order to fuse the front image with LiDAR point cloud and its front view, MV3D [25] first transforms LiDAR point cloud into BEV space (Fig. 10.7). It generates 3D object proposals in BEV space then project them back to three views. For each view,

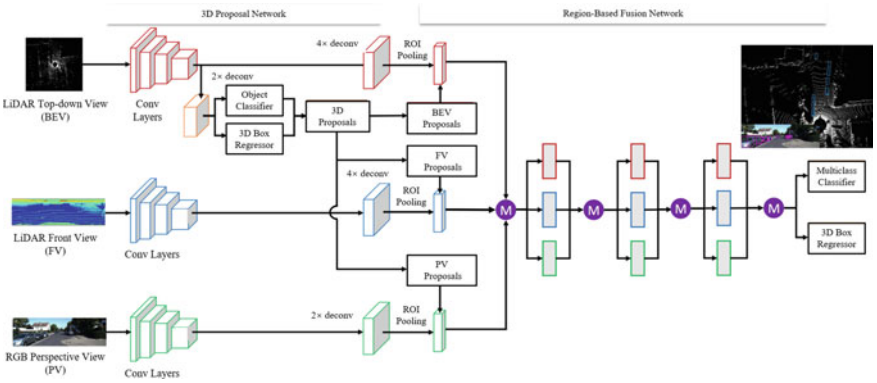

Fig. 10.7 Multi-view 3D object detection network (MV3D) [25]

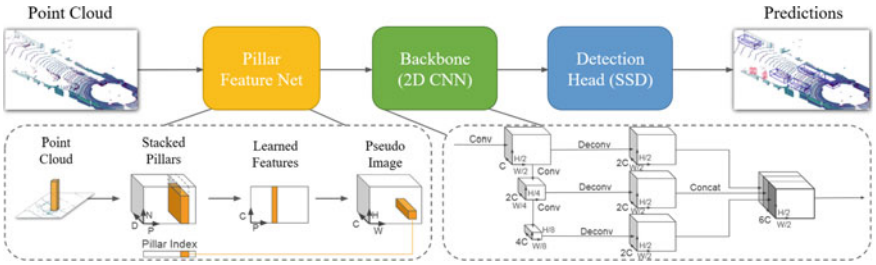

Fig. 10.8 PointPillar overview [32]

ROI pooling is used to extract region-wise features and these features are fused by A deep fusion network. The fused features are then used to predict 3D bounding boxes and object class jointly. The BEV representation of LiDAR point cloud is widely adopted in other works [26–31] as well. The term "pillar" refers to a unique kind of voxel with limitless height, which is first introduced in PointPillars [32]. To learn a representation of points in pillars, PointPillars utilizes a condense version of the PointNet [33]. Following that, standard 2D convolutional networks and detection heads are applied to process the encoded features. Although the PointPillar does not perform as well as other state-of-the-art (SOTA) methods, it and its variants are highly efficient, making them appropriate for industrial applications (Fig. 10.8).

10.4 Camera-Based BEV Perception

Comparing with LiDAR point clouds, 2D images do not naturally preserve accurate range information. Hence the core issue of camera-based BEV perception is to learn the depth from images explicitly or implicitly. According to the methods for transformation from a perspective view (PV) into a BEV, the camera-based BEV perception can be classified into two categories: 2D-3D methods and 3D-2D methods. The former "lifts" 2D features to 3D space via reconstructing depth information from 2D features, and the latter utilizes 3D-2D projection mapping to encode 2D features with 3D information (Fig. 10.9).

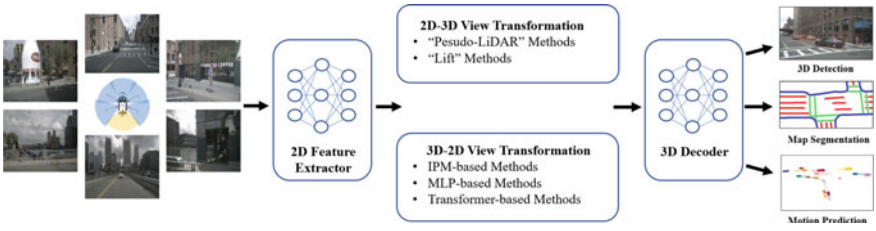

Fig. 10.9 General pipeline of camera-based BEV perception

10.4.1 2D-3D Methods

Since depth information is inaccessible for monocular cameras, 2D-3D methods construct the view transformation by predicting the depth. Currently, there are two sets of mainstream methods. One set is the "pseudo-LiDAR" method, which predicts a dense depth map, and is represented by the Pseudo-LiDAR family [34, 35]. The other set is the "lift" method, which predicts depth distribution and is represented by LSS (Lift, Splat, Shoot) [36] and its variants.

10.4.1.1 "Pseudo-LiDAR" Methods

As the name implies, Pseudo-LiDAR [34] estimates pixel-wise depth maps from stereo or monocular images and converts depth maps into pseudo-LiDAR point clouds, which can be direct input for 3D detectors designed for LiDARs. The pipeline of Pseudo-LiDAR is shown in Fig. 10.10, which is a straightforward way that can easily integrates mature experience of SOTA monocular depth estimation and LiDAR-based 3D detection methods. The accuracy of monocular depth estimation network is insufficient, while stereo network can provide better depth estimation and already has excellent performance in perception for autonomous driving. Therefore, Pseudo-LiDAR++ [35] improves the accuracy of depth estimation of faraway objects by adopting the stereo network. In AM3D [37], the pseudo-LiDAR point clouds is enhanced by corresponding RGB features. PatchNet [38] owes the effectiveness of the "pseudo-LiDAR" method to the coordinate system transformation rather than the data representation itself. However, the "Pseudo-LiDAR" methods have several problems. The performance of detection is heavily rely on the accuracy of depth estimation [39]. And the pixel-wise ground truth in outside scenes are difficult to obtain. Moreover, such methods suffer from data leakage and generalization problems. Researches show that the data leakage of KITTI depth benchmark causes the performance gap between validation and testing sets [34, 40]. To allow the pipeline can be trained in an end-to-end manner, E2E Pseudo-LiDAR [41], as illustrated in Fig. 10.11, introduces a Change-of-Representation module. The module improves generalization performance by alleviating the gradient cut-off between 3D object detection stage and depth estimation stage.

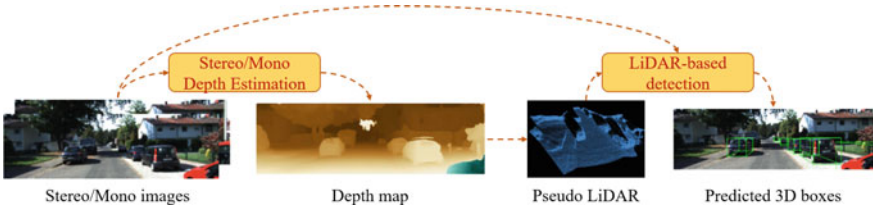

Fig. 10.10 Pipeline of Pseudo-LiDAR [34]

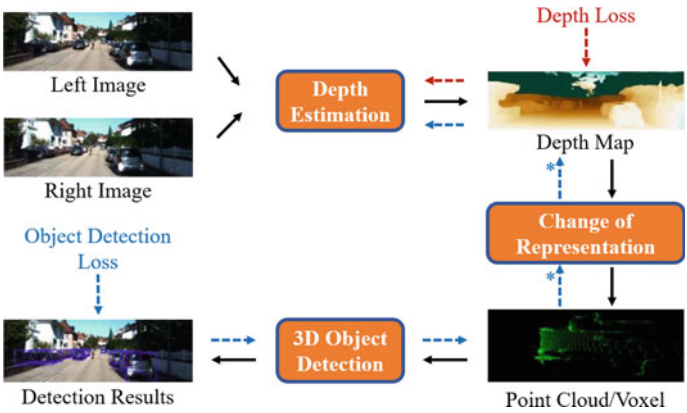

Fig. 10.11 The architecture of E2E Pseudo-LiDAR [41]

10.4.1.2 "Lift" Methods

Rather than predicting dense depth map, "Lift" paradigm predicts the depth distribution instead. LSS [36] is representative of this approach (Fig. 10.12). For each pixel, it predicts a context vector $\mathbf{c} \in \mathbb{R}^C$ and a categorical distribution over depth $\alpha \in \Delta^{D-1}$, and the corresponding 3D feature is determined by their outer product $\alpha \cdot \mathbf{c}$. In other words, the 2D features are "lifted" via depth distribution to 3D space, then the downstream perception task can be performed following SOTA LiDAR-based approaches. However, there are several drawbacks of such methods: (1) the depth distribution estimated is discrete and sparse; (2) the boundaries of objects are difficult to handle. A lot of works [42–47] follow this LSS paradigm as well. To improve the accuracy of depth distribution, CaDDN [42] uses depth ground truth derived from LiDAR point cloud as supervision (Fig. 10.13). BEVDepth [47] also uses LiDAR points as depth supervision. In addition, the intrinsic and extrinsic parameters of camera are introduced as the depth estimation prior, and the input features are adjusted in a SE-like manner. In BEVDepth, the predictions of depth and context are separated, and additional Resnet blocks are used to increase discrimination. FIERY [43] using EfficientNet [48] to extract features and predict depth distribution in the same way with LSS [36]. BEVFusion [46] adds LiDAR branch (VoxelNet [14]) onto LSS, and fuses camera features and LiDAR features in BEV space following Transfusion [49] paradigm. BEVDet [44] and its temporal enhenced version [45] follow the similar paradigm to learn an implicit view transformation (Figs. 10.14 and 10.15).

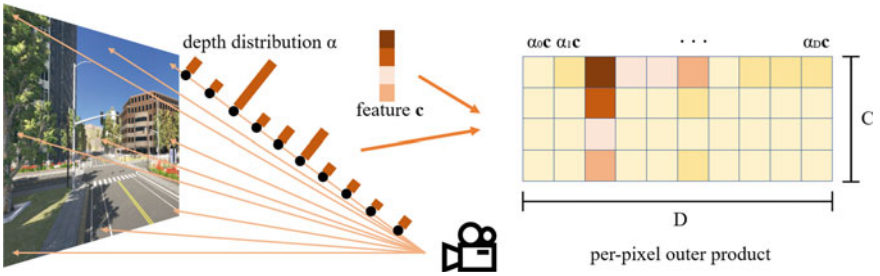

Fig. 10.12 The "lift" step of LSS [36]

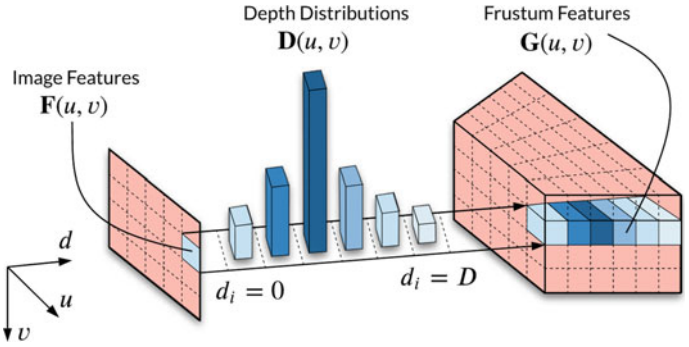

Fig. 10.13 Construction of frustum features in CaDDN [42]

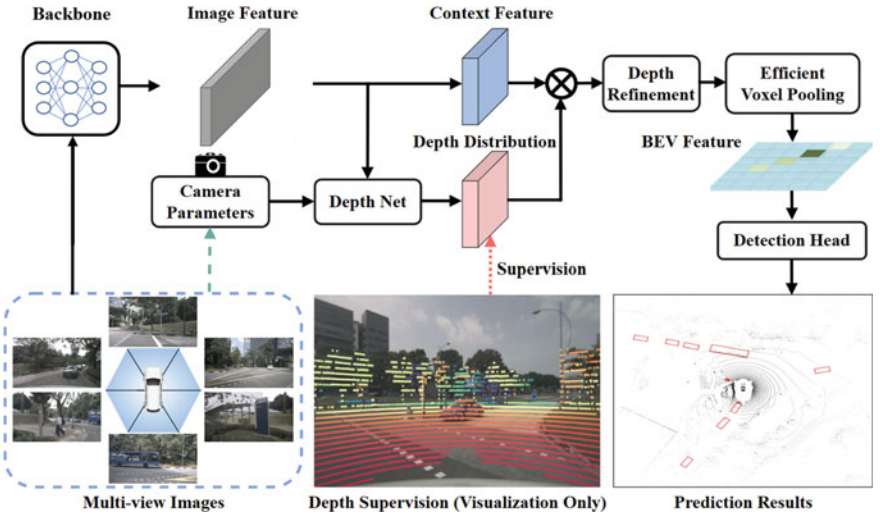

Fig. 10.14 Framework of BEVDepth [47]

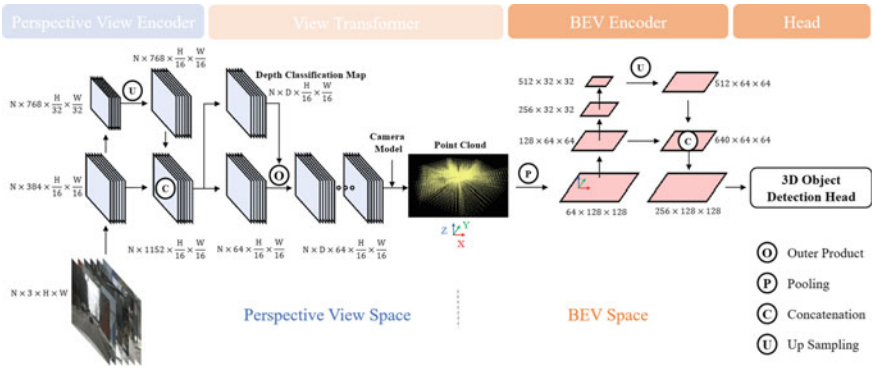

Fig. 10.15 The framework of the BEVDet [44] paradigm

10.4.2 3D-2D Methods

For 3D-2D methods, the 3D information is reconstructed by encoding 2D features to 3D space via 3D prior such as 3D-2D projection mapping. Geometry-based methods are introduced by Inverse Perspective Mapping. For pure network-based methods, MLP and transformer are two main branches, Some works index local features according to 3D-2D projection such as explicit BEV feature modeling [50, 51], or DETR3D [1] and its derivants [52]. Other methods utilize implicit 3D positional encoding [53] to avoid direct detph estimation.

10.4.2.1 IPM-Based Methods

Inverse Perspective Mapping (IPM) is the pioneer work of 3D-2D methods. Due to the lack of depth information caused by perspective mapping, projecting the pixels on image back to 3D space is ill-posed. Mallot et al. present IPM [10] to solve this ill-posed problem by adding constraint that the inversely mapped points lie on the ground plane, which means that the mapping can be described via a homography matrix. The detailed foundation of mathematics is introduced in Sect. 10.2. In practical applications, the homography matrix can be mathematically derived from the calibration parameters of camera settings, such as intrinsic and extrinsic matrices.

Abbas and Zisserman [54] use CNNs to extract 2D features and estimate the vertical vanishing points and horizontal vanishing lines to determine the homography matrix. After the PV-BEV transformation by IPM, many downstream perception tasks can be done in the BEV space, such as optical flow estimation, object detection, map segmentation, object tracking, path planning, etc. VPOE [55] adopts YOLOv3 [56] as the detection head to estimate the object pose on BEV. Palazzi et al. [57] maps detections from PV to BEV occupancy grid map based on synthetic dataset. In practice, the intrinsic and extrinsic parameters generally unknown or the extrinsic

parameters may change due to the bumpy road. To overcome this issue, TrafCam3D [58] introduces a dual-view network architecture to perform a more robust homography mapping.

IPM-based methods have good interpretability of the PV-BEV transformation. However, there are distorted areas where the objects are above the ground plane. To reduce the distortion, a series of subsequent works [50, 58–66] explore the semantic information or aid the domain gap between PV and BEV using GANs [67].

10.4.2.2 MLP-Based Methods

IPM-based methods are explicitly based on physical reality and strong mathematical assumptions. On the contrary, MLP-based methods leverage the geometry of camera. In MLP-based methods, the multilayer perceptron (MLP) constructs a universal approximate mapping function of view transformation in a data-driven manner.

Adopting MLP to perform 3D-2D feature projection is initially introduced by OFT [68], which projects 2D features to 3D voxel space (Fig. 10.16). Rather than predict the depth distribution, OFT scatters the same image feature along the ray from nodal point to a specific 3D point, in other word, it assumes that the depth distribution is uniform. Such assumption works in flat roads but fails in undulating roads. VED [69] introduces a variational encoder-decoder (VED) architecture with an MLP bottleneck to convert PV image to semantic BEV occupancy grid map end-to-end. VPN [70] utilizes a two-layer MLP to perform the view transformation. FishingNet [71] follows the same pattern with VPN and fuses LiDAR and RADAR data in a late fusion manner for downstream tasks. To overcome the difficulties including occlusion, lack of depth information and small objects, STA-ST [72] and PON [73] make use of Feature Pyramid Networks (FPNs) [74] to extract multi-scale image features, and compress the features along height axis and expand along dpeth aixs to leverage the vertical context, as shown in Fig. 10.17. Leveraging the vertical context via column-wise compression makes the network model more robust to occlusion and inaccessible

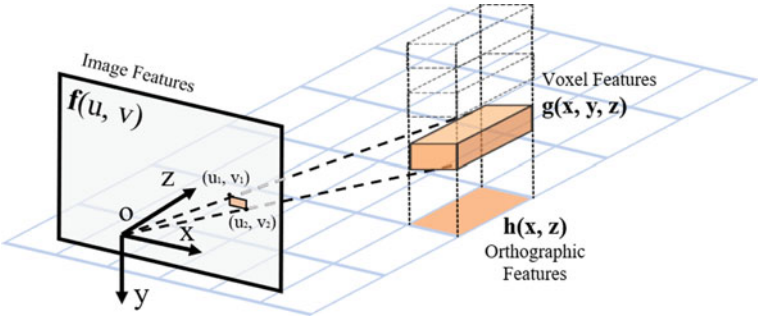

Fig. 10.16 Orthographic Feature Transform (OFT) [68]. Orthographic features are generated by feature accumulation and compression

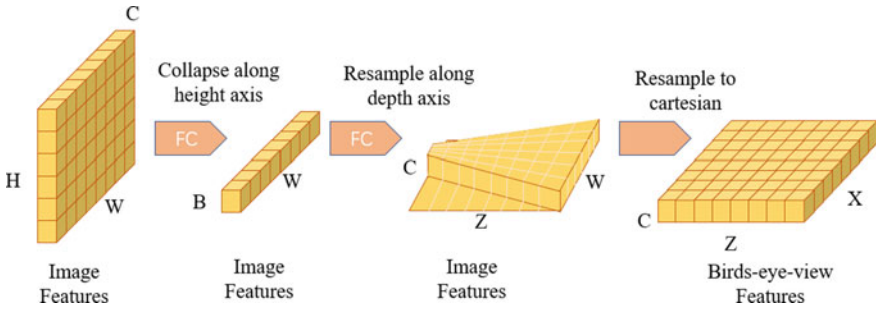

Fig. 10.17 PON [73]: features extracted by FPN are collapsed along height axis and expanded along depth axis

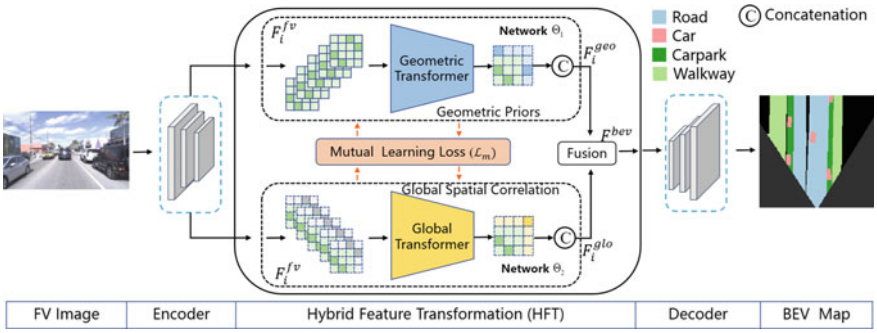

Fig. 10.18 An overview architecture of HFT [77]

depth information. It is also widely adopted and promoted in the transformer-based methods, which is discussed in Sect. 10.4.2.3.

To improve the accuracy of vectorized HD map, HDMapNet [75] uses an extra MLP to build bidirectional projection between PV and BEV to check whether the features are correctly mapped. Inspired by the back projection, PYVA [76] puts forward a bidirectional self-supervision scheme with stronger attention-based BEV feature. HFT [77] combines a camera model-based branch with a camera model-free branch via a hybrid feature transformation to utilize geometric prior and capture global context respectively (Fig. 10.18).

10.4.2.3 Transformer-Based Methods

With the development of attention mechanism and vision transformer (ViT), a lot of BEV perception works utilize transformer architecture to model the 3D-2D view transformation in an implicit way. On the strength of the cross attention, transformer is more data-dependent, thus has more powerful spatio-temporal representation but

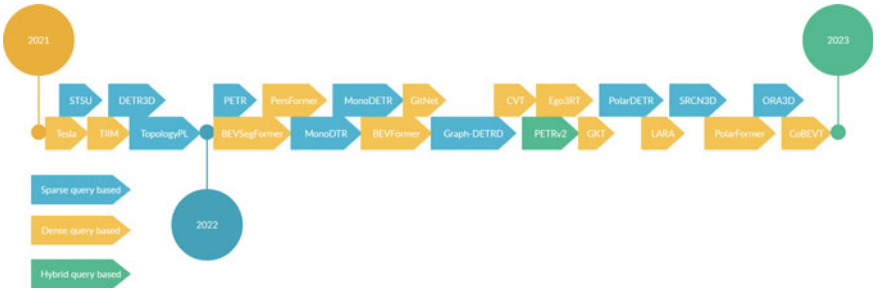

Fig. 10.19 Chronological overview of transformer-based methods [78]

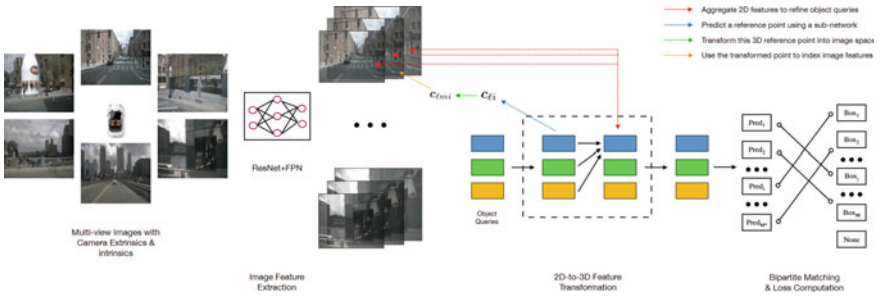

Fig. 10.20 Overview of DETR3D [1]

is difficult to train. Designing effective positional encoding and constructing appropriate queries are still challenging.

Based on the granularity of learnable queries in the transformer decoder, the transformer-based methods can be divided into three categories: sparse query-based, dense query-based and hybrid query-based [78]. A chronological overview of transformer-based methods is illustrated in Fig. 10.19. In terms of specific BEV perception downstream tasks, sparse queries are commonly used for detection tasks and dense queries are commonly used for segmentation tasks.

Sparse query-based methods are represented by the "DETR" family [1, 41, 53, 79–84]. As shown in Fig. 10.20, DETR3D [1] takes multi-view images as input and adopts ResNet and FPN to extract 2D features. After defining a set of sparse object queries, each one is translated into a 3D reference point. By reprojecting the 3D reference point into the image space, 2D features are sampled in the meantime to refine the object queries. In the end, it employs a set-to-set loss and produces predictions for each query. In order for sparse queries to interact with 2D features directly in the vanill cross attention, PETR [53] encodes 3D positional embedding derived from camera settings into 2D multi-view features, which considerably streamlines the feature sampling procedure in DETR3D (Fig. 10.21). The follow-up study PETRv2 [81] makes use of the temporal information by extending the 3D positional embedding from spatial domain to temporal domain. Graph-DETR3D [82] makes use of graph

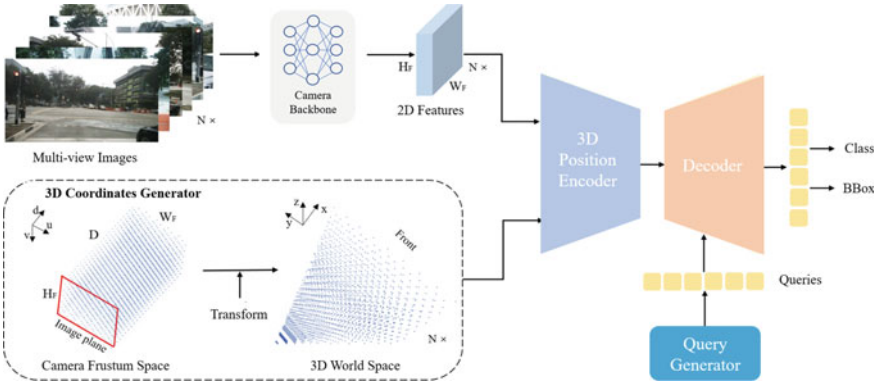

Fig. 10.21 The architecture of the proposed PETR paradigm [53]

structure learning, which enhances the inadequate feature aggregation of DETR3D and improves the performance in the overlap regions. ORA3D [83] utilizes stereo disparity supervision and adversarial training. PolarDETR [84] reformulates the 3D detection task in the polar coordinate system, which takes full advantage of the symmetry of multiple views. SRCN3D [85] designs the first two-stage 3D-2D surround-view camera 3D detection approach in a fully-convolutional architecture based on SparseRCNN [86]. It replaces global attention with local dynamic instance interaction head to get lower computation cost.

Dense queries are pre-allocated with corresponding positions in 3D space or BEV space for dense query-based approaches. The quantity of dense queries (spatial resolution) is typically larger than the quantity of sparse queries in sparse query-based algorithms. For a variety of downstream tasks, including 3D detection, motion prediction and map segmentation, the interaction between image features and dense queries can lead to the dense BEV representation.

Tesla [87] takes advantage of positional encoding and context summary to generate dense queries in the BEV space, and performs the view transformation using the vanilla cross attention between image features in multiple views and dense BEV queries, without considering the camera parameters. CVT [2] constructs a camera-aware positional embedding derived from calibrated parameters of surround-view cameras. Then through a succession of cross attention layers, it learns a BEV positional embedding that aggregate cross-view information (Fig. 10.22).

To balance the memory consumption and model scalability, Persformer [51] and BEVSegFormer [88] adopt deformable attention [89] in the view transformation module for 3D lane detection and BEV segmentation, respectively. BEVFormer [5] takes advantage of the deformable attention to interact dense BEV queries with multi-view features. Current queries and history queries are also interacted via deformable attention to leverage temporal information. Inspired by the ray tracing perspective, Ego3RT [90], as shown in Fig. 10.23, learns ego 3D representation from 2D image features making use of deformable attention and executes multiple downstream tasks.

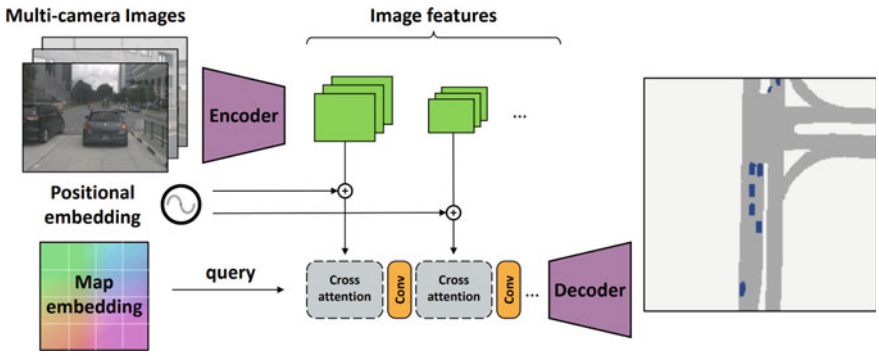

Fig. 10.22 The pipeline of CVT [2]

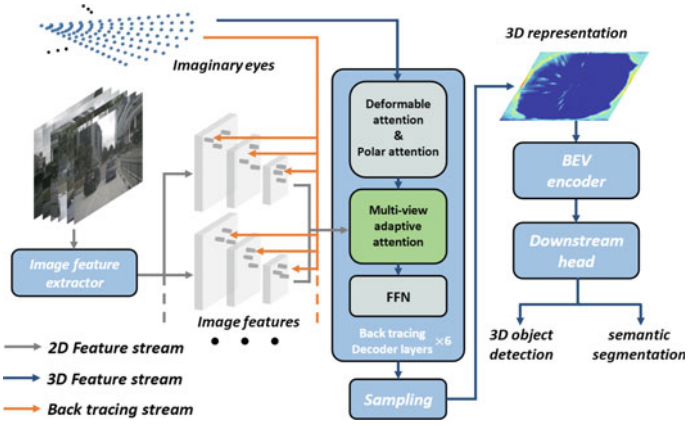

Fig. 10.23 Ego3RT pipeline [90]

CoBEVT [91] introduces fused axial attention (FAX), a novel attention variant, which can effectively aggregate features in both local and global agent or camera views.

The geometric priors are used by GKT [93] to guide the transformer to concentrate on discriminative regions and unfolds kernel features to produce BEV representation. TIIM [92] treats 1–1 correspondence between each image column and BEV polar ray as a sequence-to-sequence translation, as shown in Fig. 10.24. Such geometric constraint avoids the dense cross attention between 2D image features and BEV queries, and makes it more suitable for the transformer-style architecture. GitNet [94] obtains coarse pre-aligned BEV features by a geometry-guided pre-alignment module and refines the features via a ray-based transformer like TIIM. PolarFormer [95] extends such ray-based or column-wise transformer from a single camera to multiple surrounding cameras. Instead of learning a quadratic "all-to-all" correspondence between features in multi-camera space and BEV space, LaRa [96] uses a moderately fixed size latent space to control computation and memory consumption of the view transformation.

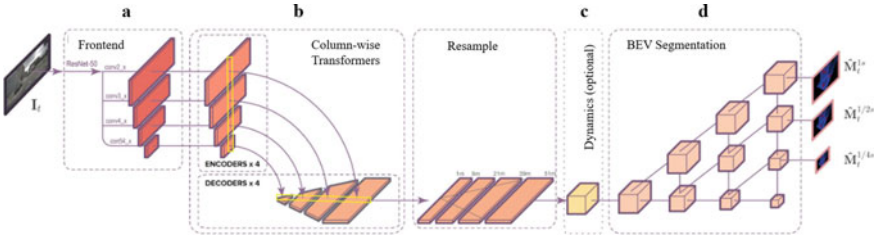

Fig. 10.24 Architecture of TIIM framework [92]. **a** Construct spatial representations in the image-plane. **b** Transform image-plane representations to BEV. **c** Construct spatiotemporal representation in BEV-plane (optional) . **d** Semantically segmentation BEV representation

10.5 Fusion-Based BEV Perception

Sensor fusion in autonomous driving is mainly about the fusion of images and point clouds (either LiDAR or RADAR). Previous methods conduct fusion in data-level [97, 98] or feature-level [25, 99–102], which rely either on calibration or directly using high-dimension features. As we have introduced in previous sections, BEV space provides a unified and convenient representation for LiDAR, camera and RADAR. Meanwhile, temporal information can be easily fused in the BEV space via ego-motion of vehicle. With the concern of robustness and consistency, a lot of works conduct BEV perception in a fusion-based way.

10.5.1 Multi-modal Fusion

Some methods operate feature fusion in 3D space [33, 89, 103–106]. As Fig. 10.25 shows, UVTR converts image and point cloud to modality-specific voxel space and fuses the features with a voxel encoder. Some other methods perform feature fusion in BEV space [46, 71, 107]. As a representative work, BEVFusion [46] extracts multi-modal features from LiDAR and multi-view cameras, the features are transformed into a shared BEV space efficiently and fused with a fully-convolutional BEV encoder. Then the encoded features are decoded by task-specific heads to support different perceptual tasks (Fig. 10.26). Besides fusion in 3D space or BEV space, there is a kind of methods conduct fusion via general queries [49, 52]. As shown in Fig. 10.27, FUTR3D [52] designs a query-based Modality-Agnostic Feature Sampler (MAFS) to extract features from all available modalities according to the 3D reference point of each query, which is easily adaptable to all sensor configurations and combinations.

The majority of recent research relies on single-agent systems, which struggle to handle occlusions and recognize far-off objects in complicated traffic situations. This problem can be addressed because to the advancement of Vehicle-to-Vehicle (V2V) communication technologies, which broadcast sensor data to other adjacent

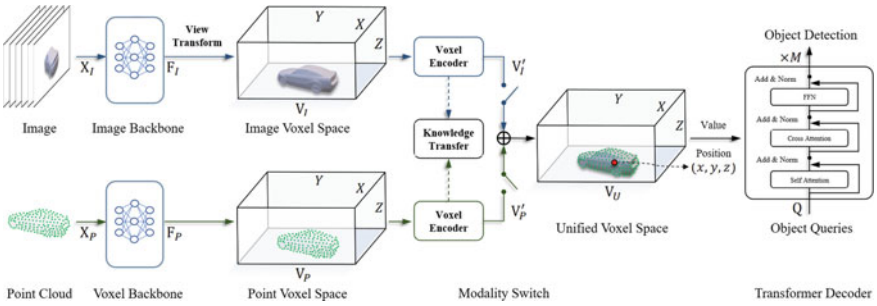

Fig. 10.25 The framework of UVTR with multi-modality inputs [103]

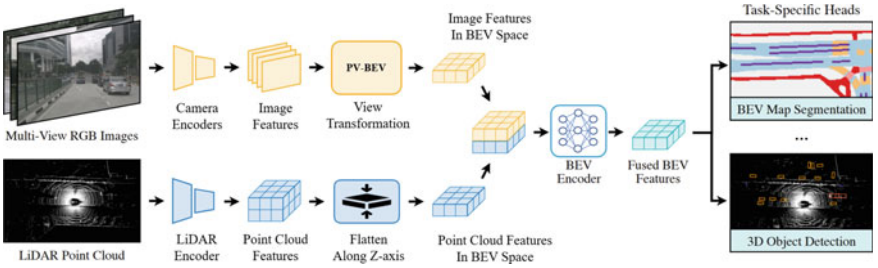

Fig. 10.26 The architecture of BEVFusion [46]

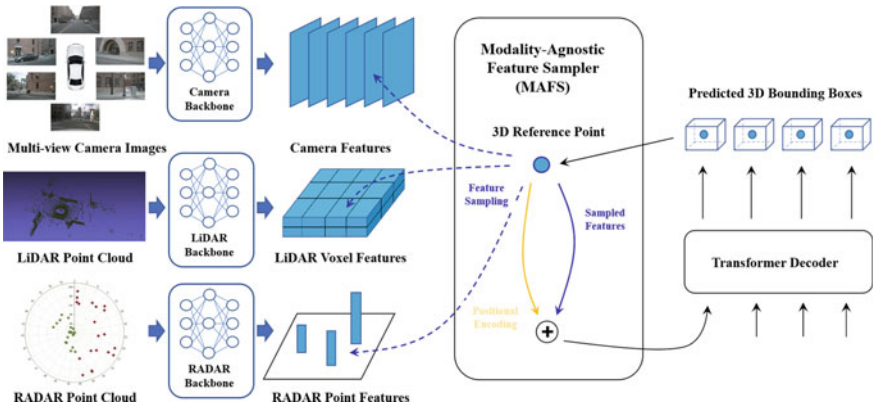

Fig. 10.27 Overview of FUTR3D [52]

autonomous cars to provide alternative perspectives of the same scene. Inspired by v2v communication technology, CoBEVT [91] develops a multi-agent multi-camera perception architecture that can cooperatively produce BEV map. However, it has only been validated on simulated datasets [108]. Most methods rely on object-ground intersections (for example, shadows) as context for depth reasoning [109]. However, these methods become unreliable for distant objects. Besides vehicle-to-

vehicle communication, object-to-object relationship is also studied. Reference [110] proposes to localize object depth by comparing objects to each other and perform message-passing across a graph of the objects via a graph neural network.

10.5.2 Temporal Fusion

To alleviate the occlusion and estimate the motion of objects, temporal fusion is an effective way for robust BEV perception system. BEVDet4D [45] fuses the feature maps with sparial alignment and concatenation. A similar concatenation-based approach has also been used in other works, including TIIM [92], FIERY [43], and PolarFormer [95]. Conerning the object motion and ego-motion, BEVFormer [5], PETRv2 [81] and UniFormer [111] utilize attention module to fuse temporal information from either previous BEV feature maps or previous frames, to more effectively create associations of the same objects in different timestamps (Fig. 10.28).

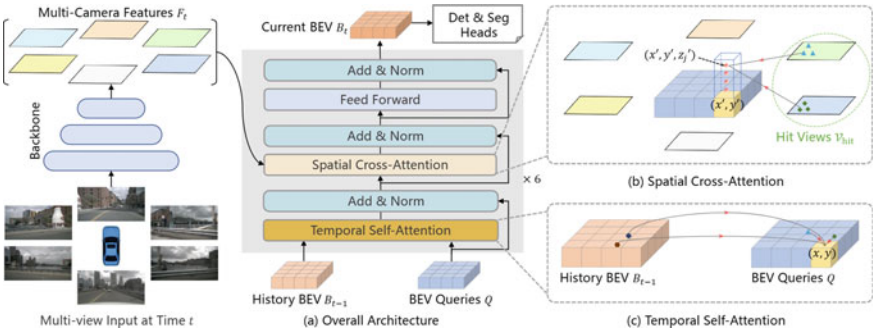

Fig. 10.28 Overall architecture of BEVFormer [5]

10.6 Datasets

KITTI [112], nuScenes [113] and Waymo Open Dataset (WOD) [114] are the most influential benchmarks for BEV perception. KITTI is a well-known benchmark for multiple autonomous driving tasks, such as stereo, optical flow, visual odometry, object detection and 3D tracking [112]. It contains 3712, 3769 and 7518 images for training, validation, and testing, respectively, also with corresponding LiDAR point clouds. The evaluation consists of 3D object detection and BEV evaluation. And there are three levels of difficulty according to objects' size, occlusion and truncation. The nuScenes dataset is a public large-scale dataset for autonomous driving developed by the team at Motional [113]. There are 1000 driving scenes in Boston

and Singapore, two cities that are known for their dense traffic and highly challenging driving situations. Among them, 850 scenes are used for training and validation, 150 scenes are used for testing, with duration of 20 seconds each. The sensors include 6 surrounding cameras, 1 LiDAR and 5 RADARs. The resolution of image is 1600 × 900 pixels. In the meantime, corresponding CAN-bus data and high-definition map (HD Map) are available to investigate the help of numerous inputs. In the training, validation, and testing sets of the Waymo Open Dataset [114], there are 798, 202, and 80 video sequences, respectively. Each sequence consists of 5 LiDARs and 5 views (side left, front left, front, front right and side right), the image resolution is 1920 × 1280 pixels or 1920 × 886 pixels.

Besides the datasets mentioned above, more benchmarks such as Argoverse, Lyft, KITTI-360, H3D, are also applicable to for BEV perception. Table 10.1 provides a summary of the detailed information for these benchmarks.

Table 10.1 Datasets for BEV Perception

Dataset	Views	Scenes	Scans	Images	Det.	Seg.	Classes	Night/rain	Stereo
KITTI 3D [112]	1	–	15 K	15 K	80 K	–	8 (3)	✗/✗	✓
nuScenes [113]	6	1,000	390 K	1.4 M	1.4 M	40 K	23 (10)	✓/✓	✗
Waymo Open Dataset [114]	5	1,150	230 K	12 M	12 M	50 K	4 (3)	✓/✓	✗
Argoverse V1 [115]	7	113	22 K	490 K	993 K	–	15	✓/✓	✓
Argoverse V2 [116]	7	1000	150 K	2.7 M	–	–	26	✓/✓	✓
Lyft L5 [117]	6	366	46 K	260 K	1.3 M	–	9	✗/✗	✗
H3D [118]	3	160	27 K	83 K	1.1 M	–	8	✗/✗	✗
Cityscapes 3D [119]	1	–	–	5 K	40 K	–	8 (6)	✓/✓	✓
KITTI 360 [120]	4	11	80 K	320 K	68 K	80 K	19	✗/✗	✓

KITTI 3D: https://www.cvlibs.net/datasets/kitti/
nuScenes: https://www.nuscenes.org/nuscenes
Waymo Open Dataset: https://waymo.com/open/
Argoverse v1: https://www.argoverse.org/av1.html
Argoverse v2: https://www.argoverse.org/av2.html
Lyft L5: https://www.woven-planet.global/en/data/perception-dataset
H3D: https://usa.honda-ri.com/H3D
Cityscapes 3D: https://www.cityscapes-dataset.com
KITTI 360: https://www.cvlibs.net/datasets/kitti-360/

10.7 Evaluation Metrics

The most commonly used evaluation metric for BEV detection is average precision (AP), which refers to the area under the precision-recall curve. In order to calculate AP, the Intersection-over-Union (IoU) is used to measure the difference between predictions and ground truth (annotations, labels). The definition of IoU between prediction bounding box and ground truth is given by (10.5):

$$IoU = \frac{area \quad of \quad overlap}{area \quad of \quad union}. \tag{10.5}$$

If the IoU is beyond the threshold, the prediction is seen as True Positive (TP). Otherwise, the result is treated as False Positive. False Negtive (FN) and True Negtive (TN) can be defined in the same way. Thus, Precision and Recall can be calculated as follows:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Predictions}, \tag{10.6}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{GroundTruth}. \tag{10.7}$$

Using the interpolated values, AP can be calculated by (10.8):

$$AP = \frac{1}{|\mathbb{R}|} \sum_{r \in \mathbb{R}} p_{interp}(r), \tag{10.8}$$

where $\mathbb{R}$ denotes the set of all recall positions, the interpolation function $p_{interp}(\cdot)$ is defined as: $p_{interp} = max_{r':r' \geq r} p(r')$. Mean average precision (mAP) is the average of APs of different classes or difficulty levels.

As for evaluation metrics for BEV segmentation, the IoU for each class and mIoU over all classes are the most frequently adopted.

Besides the common metrics, there are dataset-specific metrics.

- Average Orientation Similarity (AOS):

$$AOS = \frac{1}{|\mathbb{R}|} \sum_{r \in \mathbb{R}} \max_{r':r' \geq r} c(r'), \tag{10.9}$$

  where the c(r) denotes orientation similarity, which is a normalized variant of the cosine similarity.
- Average Translation Error (ATE) calculates the Euclidean distance between predicted object center and ground truth on the 2D ground plane.
- Average Scale Error (ASE) calculates the 3D IoU error ($1 - IoU$) after performing the alignment of translation and orientation.

- Average Orientation Error (AOE) refers to the smallest yaw angle bias between the predictions and labels.
- Average Velocity Error (AVE) is defined as the L2 norm of 2D velocity differences.
- Average Attribute Error (AAE) indicates one minus attribute classification accuracy.
- nuScenes Detection Score (NDS) is the combination of the mean AP and the mean TP over all catergories:

$$NDS = \frac{1}{10}\left[5 \cdot mAP + \sum_{k=1}^{5}(1 - \min(1, mTP_k))\right]. \tag{10.10}$$

- Average Precision weighted by Heading (APH) takes account of heading angel while calculating the AP metric.
- Longitudinal Error Tolerant 3D Average Precision (LET-3D-AP) is designed to be more tolerant with respect to depth estimation errors.

$$LET - 3D - AP = \int_0^1 p(r)dr, \tag{10.11}$$

where $p(r)$ is the precision value at recall r.

- Longitudinal Affinity Weighted LET-3D-APL (LET-3D-APL) penalizes the predictions that do not overlap with any ground truth.

$$LET - 3D - APL = \int_0^1 p_L(r)dr = \int_0^1 \bar{a_l} \cdot p(r)dr, \tag{10.12}$$

where $p_L(r)$ indicates the longitudinal affinity weighted precision, the precision value at recall r is denoted by $p(r)$, and the factor $\bar{a_l}$ means the average longitudinal affinity of all matched predictions treated as TP.

10.8 Industrial Applications

In the industry, BEV perception has been on the rise in recent years. The system-level architecture design for BEV perception is discussed in this section. The various BEV perception architectures put forth by companies worldwide are summarized in Fig. 10.29. The industrial BEV perception architectures usually consist of data preprocessing, feature extractor, PV-BEV transformation, fusion module and task-specific prediction head. Each module is elaborated in details below.

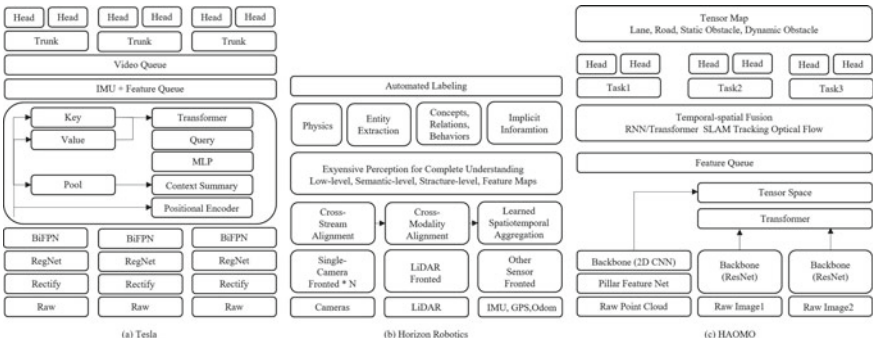

Fig. 10.29 BEV architecture comparison across industrial corporations [87, 121, 122]

10.8.1 Data Preprocessing

Several data modalities, including camera, LiDAR, RADAR, IMU, and GPS, are supported by BEV-based perception algorithms at present. The primary perception sensors for autonomous driving are cameras and LiDAR. Several products, such as those made by Tesla [87], PhiGent [123], and Mobileye [124], only use cameras as input sensors. The others use a variety of camera and LiDAR combinations, for example, Horizon [121] and HAOMO [122]. Be aware that sensor fusion designs frequently use IMU and GPS signals [87, 121, 122]. For examples, see Tesla, Horizon, and HAOMO. In the data preprocessing, numerous sensors are usually calibrated and synchronized.

10.8.2 Feature Extraction

A backbone network and a neck network frequently make up the feature extractor module, which transforms raw data into useful feature representations. Several feature extractor combinations exist for the backbone network and neck network. As an illustration, the image backbone network can be RegNet [125] in Tesla and ResNet [126] in HAOMO. The neck network could be FPN [74] adopted by HAOMO, BiFPN [127] utilized by Tesla, etc. The voxel-based option from Mobileye or the pillar-based choice from HAOMO are both ideal candidates for the backbone in terms of point cloud input.

10.8.3 PV-BEV Transformation and Fusion

In industry, there are mainly four approaches to perform PV-BEV transformation:

1. Fixed IPM. Projecting PV features into BEV space can be achieved by a fixed view transformation based on the assumption that the ground is flat. The ground plane is handled well by fixed IPM projection. Nonetheless, it is sensitive to road flatness and vehicle jolting.
2. Adaptive IPM. The extrinsic parameters of self-driving vehicles are used by adaptive IPM, which projects features to BEV in accordance with these parameters. Adaptive IPM still makes the flat ground assumption despite being robust to vehicle pose.
3. Transformer-based view transformation. Transforming PV features into BEV space using a dense transformer is known as transformer-based view transformation. Tesla, Horizon, and HAOMO have all broadly adopted this data-driven transformation since it works without making any assumptions first.
4. ViDAR (Pseudo-LiDAR). This term is first proposed by Waymo and Mobileye concurrently at different venues [124, 128] in early 2018, to describe the method of projecting PV features from visual inputs into BEV space, resembling the representation form of LiDAR point cloud. Therefore, point cloud-based techniques can be applied to obtain BEV features. Recently, there have been numerous ViDAR applications, including those from Tesla, Mobileye, Waymo, Toyota, [87, 124, 128–130], etc.

In the industrial world, ViDAR and transformer choices predominate. And the multi-modal fusion or spatiotemporal fusion are often adopted as a auxiliary of the view transformation module.

10.8.4 Perception Heads

The multi-head design is widely adopted in industrial BEV perception architecture as well. All prediction results such as motion prediction, map segmentation and 3D bounding boxes are decoded from BEV feature space since BEV feature is a unified representation that aggregates multi-modal information. In some designs, prediction results in PV are also decoded from the associated PV features. According to Horizon Robotics [121], the prediction results can be classified into four categories: (a) Low-level results are related to physics constrains, such as optical flow, depth, normal vector, etc. (b) Semantic-level results include entity extractions, for example, vehicle bounding boxes, laneline segmentation, etc. (c) Structure-level results represent relationship between objects, including object tracking, motion prediction, etc. (d) Implicit information such as feature map of auxiliary task.

10.9 Existing Challenges

The processing of raw point cloud in LiDAR-based methods requires high memory consumption and computational cost. However, converted representations such as voxel and pillar lose information to varying degrees. Generating a balanced trade-off between performance and efficiency becomes a vital challenge for LiDAR-based BEV perception.

For camera-based methods, the core issue is to reconstruct 3D information from 2D images. In 2D-3D methods, the effort is devoted to estimating depth or depth distribution. However, "Pseudo-LiDAR" methods can not estimate accurate depth as real LiDAR, while "Lift" methods do not perform well at object boundaries and far away distance. Thus a better depth estimation approach is in urgent need to be explored. In 3D-2D methods, recent studies devote to complementing geometric into network approaches. IPM-based methods have good interpretability but can not avoid distortions above the ground plane. MLP-based methods are theoretically a universal approximator [131], but not fully explored in the depth, occlusion and multi-view. Transformer-based methods have best performance at present but still need to further investigate better queries, spatiotemporal fusion, network lightweight and polar parameterization. For network training, effective ground truth annotation for the BEV grid is still pending.

In general, the existing challenges of BEV perception are:

- Proposing a better representation of point cloud;
- Designing a better depth estimator for 2D images;
- Achieving better generalization ability and robustness;
- Obtaining the truth value annotation in the BEV grid conveniently;
- Trading-off between performance and efficiency in applications.

10.10 Conclusions

In the traditional approach, various perception tasks such as 3D object detection, obstacle instance segmentation, lane line segmentation, and trajectory prediction are separated from each other. This separation makes the autonomous driving algorithm need to use multiple sub-modules in series, which greatly increases the development and maintenance cost of the whole system. BEV perception allows these perception tasks to be implemented within a single algorithmic framework, significantly reducing the need for manpower. Considering the advantages of BEV mentioned above, many research institutions and major car companies are promoting the implementation of BEV solutions. Corresponding BEV algorithms have been proposed based on different combinations of sensor input layers, basic tasks, and product scenarios. While it is certain that BEV perception algorithm can integrate the features of multiple sensors and improve the accuracy of perception and prediction, its potential to

be the ultimate solution for autonomous driving is uncertain and depends on various factors.

Acknowledgements This work was supported by the National Key R&D Program of China under Grant 2020AAA0108100, the National Natural Science Foundation of China under Grant 62233013, and the Science and Technology Commission of Shanghai Municipal under Grant 22511104500.

References

1. Wang Y et al (2022) DETR3D: 3D object detection from multi-view images via 3D-to-2D queries. In: Conference on robot learning (CoRL). PMLR, pp 180–191
2. Zhou B, Krähenbühl P (2022) Cross-view transformers for real-time map-view semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 13760–13769
3. Zhang Y et al (2022) Beverse: unified perception and prediction in birds-eye-view for vision-centric autonomous driving. Comput Res Repos (CoRR). arXiv:2205.09743
4. Yulan G et al (2020) Deep learning for 3D point clouds: a survey. IEEE Trans Pattern Anal Mach Intell 43(12):4338–4364
5. Li Z et al (2022) BEVFormer: learning Bird's-eye-view representation from multi-camera images via spatiotemporal transformers. Comput Res Repos (CoRR). arXiv:2203.17270
6. Rui F et al (2018) Road surface 3D reconstruction based on dense subpixel disparity map estimation. IEEE Trans Image Process 27(6):3025–3035
7. Rui F et al (2021) Learning collision-free space detection from stereo images: homography matrix brings better data augmentation. IEEE/ASME Trans Mechatron 27(1):225–233
8. Rui F et al (2021) Rethinking road surface 3-d reconstruction and pothole detection: from perspective transformation to disparity map segmentation. IEEE Trans Cybern 52(7):5799–5808
9. Luo L-B et al (2010) Low-cost implementation of bird's-eye view system for camera-on-vehicle. In: ICCE 2010 - 2010 digest of technical papers international conference on consumer electronics (ICCE), pp 311–312
10. Mallot HA et al (1991) Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biol Cybern 64(3):177–185
11. Li H et al (2022) Delving into the devils of bird's-eye-view perception: a review, evaluation and recipe. Comput Res Repos (CoRR). arXiv:2209.05324
12. Graham B (2014) Spatially-sparse convolutional neural networks. Comput Res Repos (CoRR). arXiv:1409.6070
13. Choy C et al (2019) 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 3075–3084
14. Zhou Y, Tuzel O (2018) VoxelNet: end-to-end learning for point cloud based 3D object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4490–4499
15. Yan Y et al (2018) SECOND: Sparsely embedded convolutional detection. Sensors 18(10):3337
16. Yin T et al (2021) Center-based 3D object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11784–11793
17. Shi S et al (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 (CVPR), pp 10529–10538

18. Chenhang He et al (2020) Structure aware single-stage 3D object detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11873–11882
19. Jiajun D et al (2021) Voxel R-CNN: towards high performance voxel-based 3D object detection. Proceedings of the AAAI conference on artificial intelligence (AAAI) 35:1201–1209
20. Wang Y, Solomon JM (2021) Object DGCNN: 3D object detection using dynamic graphs. Adv Neural Inf Process Syst (NeurIPS) 34:20745–20758
21. Mao J et al (2021) Voxel transformer for 3D object detection. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 3164–3173
22. Fan L et al (2022) Embracing single stride 3D object detector with sparse transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 8458–8468
23. Liu Z et al (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 10012–10022
24. Hu Y et al (2022) AFDetV2: rethinking the necessity of the second stage for object detection from point clouds. Proceedings of the AAAI conference on artificial intelligence (AAAI) 36:969–979
25. Chen X et al (2017) Multi-view 3D object detection network for autonomous driving. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1907–1915
26. Yang B et al (2018) Pixor: real-time 3D object detection from point clouds. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 7652–7660
27. Yang B et al (2018) HDNet: exploiting HD maps for 3D object detection. In: Conference on robot learning (CoRL). PMLR, pp 146–155
28. Beltrán J et al (2018) BirdNet: a 3D object detection framework from LiDAR information. In: 2018 21st international conference on intelligent transportation systems (ITSC). IEEE, pp 3517–3523
29. Yiming Z et al (2018) RT3D: real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving. IEEE Robot Autom Lett (RAL) 3(4):3434–3440
30. Ali W et al (2018) YOLO3D: end-to-end real-time 3D oriented object bounding box detection from LiDAR point cloud. In: Proceedings of the European conference on computer vision (ECCV) workshops
31. Simony M et al (2018) Complex-YOLO: an Euler-region-proposal for real-time 3D object detection on point clouds. In: Proceedings of the European conference on computer vision (ECCV) workshops
32. Lang AH et al (2019) PointPillars: fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 12697–12705
33. Qi CR et al (2017) PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 652–660
34. Wang Y et al (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 (CVPR), pp 8445–8453
35. You Y et al (2019) Pseudo-LiDAR++: accurate depth for 3D object detection in autonomous driving. Comput Res Repos (CoRR). arXiv:1906.06310
36. Philion J, Fidler S (2020) Lift, splat, shoot: encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In: European conference on computer vision (ECCV). Springer, pp 194–210
37. Ma X et al (2019) Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 6851–6860

38. Ma X et al (2020) Rethinking Pseudo-LiDAR representation. In: European conference on computer vision (ECCV). Springer, pp 311–327
39. Lian Q et al (2022) Exploring geometric consistency for monocular 3D object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1685–1694
40. Simonelli A et al (2021) Are we missing confidence in Pseudo-LiDAR methods for monocular 3D object detection? In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 3225–3233
41. Qian R et al (2020) End-to-end Pseudo-LiDAR for image-based 3D object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 5881–5890
42. Reading C et al (2021) Categorical depth distribution network for monocular 3D object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 8555–8564
43. Hu A et al (2021) FIERY: future instance prediction in Bird's-Eye view from surround monocular cameras. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 15273–15282
44. Huang J et al (2021) BEVDet: high-performance multi-camera 3D object detection in bird-eye-view. Comput Res Repos (CoRR). arXiv:2112.11790
45. Huang J, Huang G (2022) BEVDet4D: exploit temporal cues in multi-camera 3D object detection. Comput Res Repos (CoRR). arXiv:2203.17054
46. Liu Z et al (2022) BEVFusion: multi-task multi-sensor fusion with unified Bird's-eye view representation. Comput Res Repos (CoRR). arXiv:2205.13542
47. Li Y et al (2022) BEVDepth: acquisition of reliable depth for multi-view 3D object detection. Comput Res Repos (CoRR). arXiv:2206.10092
48. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning (ICML). PMLR, pp 6105–6114
49. Bai X et al (2022) TransFusion: robust LiDAR-camera fusion for 3D object detection with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1090–1099
50. Garnett N et al (2019) 3D-LaneNet: end-to-end 3D multiple lane detection. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 2921–2930
51. Chen L et al (2022) PersFormer: 3D lane detection via perspective transformer and the OpenLane benchmark. Comput Res Repos (CoRR). arXiv:2203.11089
52. Chen X et al (2022) FUTR3D: a unified sensor fusion framework for 3D detection. Comput Res Repos (CoRR). arXiv:2203.10642
53. Liu Y et al (2022) PETR: position embedding transformation for multi-view 3D object detection. Comput Res Repos (CoRR). arXiv:2203.05625
54. Abbas SA, Zisserman A (2019) A geometric approach to obtain a bird's eye view from an image. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV) workshops
55. Kim Y, Kum D (2019) Deep learning based vehicle position and orientation estimation via inverse perspective mapping image. In: 2019 IEEE intelligent vehicles symposium (IV). IEEE, pp 317–323
56. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. Comput Res Repos (CoRR). arXiv:1804.02767
57. Palazzi A et al (2017) Learning to map vehicles into bird's eye view. In: International conference on image analysis and processing (ICIAP). Springer, pp 233–243
58. Loukkal A et al (2021) Driving among flatmobiles: Bird-eye-view occupancy grids from a monocular camera for holistic trajectory planning. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 51–60
59. Can YB et al (2022) Understanding bird's-eye view of road semantics using an onboard camera. IEEE Robot Autom Lett (RAL) 7(2):3302–3309

60. Reiher L et al (2020) A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In: 2020 IEEE 23rd international conference on intelligent transportation systems (ITSC). IEEE, pp 1–7
61. Hou Y et al (2020) Multiview detection with feature perspective transformation. In: European conference on computer vision (ECCV). Springer, pp 1–18
62. Gu JA (2020) Homography loss for monocular 3D object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1080–1089
63. Zhu X et al (2018) Generative adversarial frontal view to bird view synthesis. In: 2018 International conference on 3d vision (3DV). IEEE
64. Srivastava S et al (2020) Learning 2D to 3D lifting for object detection in 3D for autonomous vehicles. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 4504–4511
65. Mani K et al (2020) Monolayout: amodal scene layout from a single image. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 1689–1697
66. Bruls T et al (2019) The right (angled) perspective: improving the understanding of road scenes using boosted inverse perspective mapping. In: 2019 IEEE intelligent vehicles symposium (IV). IEEE, pp 302–309
67. Antonia C et al (2018) Generative adversarial networks: an overview. IEEE Signal Process Mag 35(1):53–65
68. Roddick T et al (2018) Orthographic feature transform for monocular 3d object detection. Comput Res Repos (CoRR). arXiv:1811.08188
69. Lu C et al (2019) Monocular semantic occupancy grid mapping with convolutional variational encoder-decoder networks. IEEE Robot Autom Lett (RAL) 4(2):445–452
70. Bowen P et al (2020) Cross-view semantic segmentation for sensing surroundings. IEEE Robot Autom Lett (RAL) 5(3):4867–4873
71. Hendy N et al (2020) Fishing net: future inference of semantic heatmaps in grids. Comput Res Repos (CoRR). arXiv:2006.09917
72. Saha A et al (2021) Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE, pp 5133–5139
73. Roddick T, Cipolla R (2020) Predicting semantic map representations from images using pyramid occupancy networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11138–11147
74. Lin T-Y et al (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2117–2125
75. Li Q et al (2022) HDMapNet: an online HD map construction and evaluation framework. In: 2022 international conference on robotics and automation (ICRA). IEEE, pp 4628–4634
76. Yang W et al (2021) Projecting your view attentively: monocular road scene layout estimation via cross-view transformation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 15536–15545
77. Zou J et al (2022) HFT: lifting perspective representations via hybrid feature transformation. Comput Res Repos (CoRR). arXiv:2204.05068, 2022
78. Ma Y et al (2022) Vision-centric BEV perception: a survey. Comput Res Repos (CoRR). arXiv:2208.02797
79. Can YB et al (2021) Structured bird's-eye-view traffic scene understanding from onboard images. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 15661–15670
80. Can YB et al (2022) Topology preserving local road network estimation from single onboard camera image. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 17263–17272
81. Liu Y et al (2022) PETRv2: a unified framework for 3D perception from multi-camera images. Comput Res Repos (CoRR). arXiv:2206.01256

82. Chen Z et al (2022) Graph-DETR3D: rethinking overlapping regions for multi-view 3D object detection. Comput Res Repos (CoRR). arXiv:2204.11582
83. Roh W et al (2022) ORA3D: overlap region aware multi-view 3D object detection. Comput Res Repos (CoRR). arXiv:2207.00865
84. Chen S et al (2022) Polar parametrization for vision-based surround-view 3D detection. Comput Res Repos (CoRR). arXiv:2206.10965
85. Shi Y et al (2022) SRCN3D: Sparse R-CNN 3D surround-view camera object detection and tracking for autonomous driving. Comput Res Repos (CoRR). arXiv:2206.14451
86. Sun P et al (2021) Sparse R-CNN: end-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 14454–14463
87. Tesla. tesla ai day 2022. https://www.youtube.com/watch?v=j0z4FweCy4M Accessed August, 2021
88. Peng L et al (2023) BEVSegFormer: Bird's eye view semantic segmentation from arbitrary camera rigs. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 5935–5943
89. Zhu X et al (2020) Deformable DETR: Deformable transformers for end-to-end object detection. Comput Res Repos (CoRR). arXiv:2010.04159
90. Lu J et al (2020) Learning ego 3D Representation as Ray Tracing. Comput Res Repos (CoRR). arXiv:2206.04042, 2022
91. Xu R et al (2022) CoBEVT: cooperative bird's eye view semantic segmentation with sparse transformers. Comput Res Repos (CoRR). arXiv:2207.02202
92. Saha A et al (2022) Translating images into maps. In: 2022 international conference on robotics and automation (ICRA). IEEE, pp 9200–9206
93. Chen S et al (2022) Efficient and robust 2D-to-BEV representation learning via geometry-guided Kernel transformer. Comput Res Repos (CoRR). arXiv:2206.04584
94. Gong S et al (2022) GitNet: geometric prior-based transformation for birds-eye-view segmentation. In: 17th European conference computer vision-ECCV 2022, Part I, Tel Aviv, Israel, October 23–27, 2022, Proceedings. Springer, pp 396–411
95. Jiang Y et al (2022) PolarFormer: multi-camera 3D object detection with polar transformers. Comput Res Repos (CoRR). arXiv:2206.15398
96. Bartoccioni F et al (2022) LaRa: latents and rays for multi-camera bird's-eye-view semantic segmentation. Comput Res Repos (CoRR). arXiv:2206.13294
97. Vora S et al (2020) PointPainting: sequential fusion for 3D object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 4604–4612
98. Wang C et al (2021) PointAugmenting: cross-modal augmentation for 3D object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11794–11803
99. Piergiovanni AJ et al (2021) 4D-Net for learned multi-modal alignment. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 15435–15445
100. Xu D et al (2018) PointFusion: deep sensor fusion for 3D bounding box estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 244–253
101. Liang M et al (2018) Deep continuous fusion for multi-sensor 3D object detection. In: Proceedings of the European conference on computer vision (ECCV), pp 641–656
102. Ku J et al (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
103. Li Y et al (2022) Unifying voxel-based representation with transformer for 3D object detection. Comput Res Repos (CoRR). arXiv:2206.00630
104. Chen Z et al (2022) AutoAlign: pixel-instance feature aggregation for multi-modal 3D object detection. Comput Res Repos (CoRR). arXiv:2201.06493
105. Chen Z et al (2022) AutoAlignV2: deformable feature aggregation for dynamic multi-modal 3d object detection. Comput Res Repos (CoRR). arXiv:2207.10316

106. 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 (WACV), pp 1527–1536
107. Harley AW et al (2022) Simple-BEV: what really matters for multi-sensor BEV perception? . Comput Res Repos (CoRR). arXiv:2206.07959
108. Dosovitskiy A et al (2017) CARLA: an open urban driving simulator. In: Conference on robot learning (CoRL). PMLR, pp 1–16
109. van Dijk T, de Croon G (2019) How do neural networks see depth in single images? In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 2183–2191
110. Saha A et al (2022) "The pedestrian next to the lamppost" adaptive object graphs for better instantaneous mapping. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 19528–19537
111. Qin Z et al (2022) UniFormer: unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view. Comput Res Repos (CoRR). arXiv:2207.08536
112. Andreas G et al (2013) Vision meets robotics: the kitti dataset. Int J Robot Res 32(11):1231–1237
113. Caesar H et al (2020) nuScenes: a multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11621–11631
114. Sun P et al (2020) Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2446–2454
115. Chang M-F et al (2019) Argoverse: 3D tracking and forecasting with rich maps. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 8748–8757
116. Wilson B et al (2021) Argoverse 2: next generation datasets for self-driving perception and forecasting. Comput Res Repos (CoRR). arXiv:2301.00493
117. Houston J et al (2021) One thousand and one hours: self-driving motion prediction dataset. In: Conference on robot learning (CoRL). PMLR, pp 409–418
118. Patil A et al (2019) The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes. In: 2019 international conference on robotics and automation (ICRA). IEEE, pp 9552–9557
119. Gählert N et al (2020) Cityscapes 3D: dataset and benchmark for 9 dof vehicle detection. Comput Res Repos (CoRR). arXiv:2006.07864
120. Liao Y et al (2022) Kitti-360: a novel dataset and benchmarks for urban scene understanding in 2D and 3D. IEEE Trans Pattern Anal Mach Intell
121. Robotics H (2020) Horizon algorithms. https://en.horizon.ai/products/applications-algorithms/ Accessed 2022
122. HAOMO. haomo ai day. https://www.bilibili.com/video/BV1Wr4y1H7W7 Accessed 2022
123. PhiGent. Phigent: Technical roadmap. https://43.132.128.84/coreTechnology Accessed 2022
124. Mobileye (2020) Ces 2020 by mobileye. https://youtu.be/HPWGFzqd7pI Accessed 2020
125. Radosavovic I et al (2020) Designing network design spaces. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 10428–10436
126. He K et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
127. Tan M et al (2020) EfficientDet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 10781–10790
128. Waymo (2020) Drago anguelov - machine learning for autonomous driving at scale. https://youtu.be/BV4EXwlb3yo Accessed 2020
129. Zhou T et al (2017) Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1851–1858

130. Gordon A et al (2019) Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 8977–8986
131. Kim T, Adalı T (2003) Approximation by fully complex multilayer perceptrons. Neural Comput 15(7):1641–1666