PolarFormer: Multi-Camera 3D Object Detection with Polar Transformer

Yanqin Jiang¹,4*, Li Zhang²†, Zhenwei Miao⁵, Xiatian Zhu⁶, Jin Gao¹,4, Weimin Hu¹,4,7, Yu-Gang Jiang³

¹NLPR, Institute of Automation, Chinese Academy of Sciences
²School of Data Science, Fudan University
³School of Computer Science, Fudan University
⁴School of Artificial Intelligence, University of Chinese Academy of Sciences
⁵Alibaba DAMO Academy
⁶Surrey Institute for People-Centred Artificial Intelligence, CVSSP, University of Surrey
⁷School of Information Science and Technology, ShanghaiTech University

Abstract

3D object detection in autonomous driving aims to reason “what” and “where” the objects of interest present in a 3D world. Following the conventional wisdom of previous 2D object detection, existing methods often adopt the canonical Cartesian coordinate system with perpendicular axis. However, we conjugate that this does not fit the nature of the ego car’s perspective, as each onboard camera perceives the world in shape of wedge intrinsic to the imaging geometry with radical (non-perpendicular) axis. Hence, in this paper we advocate the exploitation of the Polar coordinate system and propose a new Polar Transformer (PolarFormer) for more accurate 3D object detection in the bird’s-eye-view (BEV) taking as input only multi-camera 2D images. Specifically, we design a cross-attention based Polar detection head without restriction to the shape of input structure to deal with irregular Polar grids. For tackling the unconstrained object scale variations along Polar’s distance dimension, we further introduce a multi-scale Polar representation learning strategy. As a result, our model can make best use of the Polar representation rasterized via attending to the corresponding image observation in a sequence-to-sequence fashion subject to the geometric constraints. Through experiments on the nuScenes dataset demonstrate that our PolarFormer outperforms significantly state-of-the-art 3D object detection alternatives.

Introduction

3D object detection is an enabling capability of autonomous driving in unconstrained real-world scenes (Wang et al. 2022b, 2021). It aims to predict the location, dimension and orientation of individual objects of interest in a 3D world. Despite favourable cost advantages, multi-camera based 3D object detection (Wang et al. 2021, 2022a,b; Zhou, Wang, and Krähenbühl 2019) remains particularly challenging. To obtain 3D representation, dense depth estimation is often leveraged (Philion and Fidler 2020), which however is not only expensive in computation but error prone. To bypass depth inference, more recent methods (Wang et al. 2022b; Li et al. 2022b) exploit query-based 2D detection (Carion et al. 2020) to learn a set of sparse and virtual embedding for multi-camera 3D object detection, yet incapable of effectively modeling the geometry structure among objects. Typically, the canonical Cartesian coordinate system with perpendicular axis is adopted in either 2D (Zhou, Wang, and Krähenbühl 2019; Wang et al. 2021) or 3D (Wang et al. 2022b; Li et al. 2022b) space. This is largely restricted by convolution based models used. In contrast, the physical world perceived under each camera in the ego car’s perspective is in shape of wedge intrinsic to the camera imaging geometry with radical non-perpendicular axis (Figure 5). Bearing this imaging property in mind, we conjugate that the Polar coordinate system should be more appropriate and natural than the often adopted Cartesian counterpart for 3D object detection. Indeed, the Polar coordinate has been exploited in a few LiDAR-based 3D perception methods (Zhang et al. 2020; Bewley et al. 2020; Rapoport-Lavie and Raviv 2021; Zhu et al. 2021). However, they are limited algorithmically due to the adoption of convolutional networks restricted to rectangular grid structure and local receptive fields.

Motivated by the aforementioned insights, in this work a novel Polar Transformer (PolarFormer) model for multi-camera 3D object detection in a Polar coordinate system is introduced (Figure 3). Specifically, we first learn the representation of polar rays corresponding to image regions in a sequence-to-sequence cross-attention formulation. Then we rasterize a BEV Polar representation consisting of a set of Polar rays evenly distributed around 360 degrees. To deal with irregular Polar grids as suffered by conventional LiDAR based solutions (Zhang et al. 2020; Bewley et al. 2020; Rapoport-Lavie and Raviv 2021; Zhu et al. 2021), we propose a cross-attention based decoder head design without restriction to the shape of input structure. For tackling the challenge of unconstrained object scale variation along Polar’s
distance dimension, we resort to a multi-scale Polar BEV representation learning strategy.

The contributions of this work are summarized as follows: (I) We propose a new Polar Transformer (PolarFormer) model for multi-camera 3D object detection in the Polar coordinate system. (II) This is achieved based on two Polar-tailored designs: A cross-attention based decoder design for dealing with the irregular Polar grids, and a multi-scale Polar representation learning strategy for handling the unconstrained object scale variations over Polar’s distance dimension. (III) Extensive experiments on the nuScenes dataset show that our PolarFormer achieves leading performance for camera-based 3D object detection (Figure 1).

Related Work

Monocular/multi-camera 3D object detection Image-based 3D object detection aims to estimate the object location, dimension and orientation in the 3D space alongside its category given only image input. To solve this ill-posed problem, (Zhou, Wang, and Krähenbühl 2019; Wang et al. 2021) naively build their detection pipelines by augmenting 2D detectors (Zhou, Wang, and Krähenbühl 2019; Tian et al. 2019) in a data driven fashion. PGD (Wang et al. 2022a) further captures the uncertainty and models the relationship of different objects by utilizing geometric prior. Contrast to the above image-only methods, depth-based methods (Xu and Chen 2018; Ding et al. 2020; Wang et al. 2019; You et al. 2019; Ma et al. 2019; Reading et al. 2021) use depth cues as 3D information to mitigate the naturally ill-posed problem. Recently, multi-camera-based 3D object detection emerges. DETR3D (Wang et al. 2022b) considers detecting objects across all cameras collectively. It learns a set of sparse and virtual query embedding, without explicitly building the geometry structure among objects/queries. (Li et al. 2022b) considers detecting objects in BEV, performing end-to-end object detection via object queries. Note that multi-camera setting uses the same amount of training data as the monocular pipelines. Both multi-camera and monocular paradigms share the same evaluation metrics.

Bird’s-eye-view (BEV) representation Recently there is a surge of interest in transforming the monocular or multi-view images from ego car cameras into the bird’s-eye-view coordinate (Roddick, Kendall, and Cipolla 2019; Philion and Fidler 2020; Li et al. 2022a; Roddick and Cipolla 2020; Reading et al. 2021; Saha et al. 2022) followed by specific optimization tasks (e.g., 3D object detection, semantic segmentation). A natural solution (Philip and Fidler 2020; Hu et al. 2021) is to learn the BEV representation by leveraging the pixel-level dense depth estimation. This however is error-prone due to lacking ground-truth supervision. Another line of research aims to bypass the depth prediction and directly leverage a Transformer (Chitta, Prakash, and Geiger 2021; Can et al. 2021; Saha et al. 2022) or a FC layer (Li et al. 2022a; Roddick and Cipolla 2020; Yang et al. 2021) to learn the transformation from camera inputs to the BEV coordinate. A similar attempt as ours is conducted in (Saha et al. 2022) but limited in a couple of aspects: (i) It is restricted to monocular input for a straightforward 2D segmentation task while we consider multiple cameras collectively for more challenging 3D object detection; (ii) We uniquely provide a solid multi-scale Polar BEV transformation to tackle the unconstrained object scale variations and followed by a jointly optimized cross-attention based Polar head.

3D object detection in Polar coordinate 3D object detection in the Polar or Polar-like coordinate system has been attempted in LiDAR-based perception methods. For example, CyliNet (Zhu et al. 2021) introduces range-based guidance for extracting Polar-consistent features. In particular, it adapts a Cartesian heatmap to a Polar version for object classification, whilst learning relative heading angles and velocities. However, CyliNet still lags clearly behind the Cartesian counterpart. Recently, PolarSteam (Chen, Vora, and Bei-
Eq. (2) suggests that the azimuth $\phi^{(P)}$ is irrelevant to the vertical value of image coordinate. It is hence natural to build a one-to-one relationship between Polar rays and image columns (Saha et al. 2022). However, we need object depth $z^{(C)}$ to compute radius $\rho^{(P)}$, the estimation of which is ill-posed. Instead of explicit depth estimation, we leverage the attention mechanism (Vaswani et al. 2017) to model the relationship between pixels along the image column and positions along the Polar ray.

Let $f_{n,u,w} \in \mathbb{R}^{H_i \times C}$ represent the image column from $n$th camera, $u$th scale and $w$th column, and $\hat{p}_{n,u,w} \in \mathbb{R}^{R_u \times C}$ denote the corresponding Polar ray query we introduce, where $H$ and $R$ are the image feature map’s height and Polar map’s range. We formulate cross-plane attention as:

$$
p_{n,u,w} = \text{MultiHead}(\hat{p}_{n,u,w}, f_{n,u,w}),
$$

where

$$
\text{head}_i = \text{Attention}(\hat{p}_{n,u,w} W^{Q,i}, f_{n,u,w} W^{K,i} \odot f_{n,u,w} W^{V,i}),
$$

where $W^{Q,i} \in \mathbb{R}^{d_{model} \times d}, W^K \in \mathbb{R}^{d_{model} \times d_k}, W^V \in \mathbb{R}^{d_{model} \times d_u}$. $W^{Q,i}$, $W^K$ and $W^V$ are the projection parameters, $d_q = d_k = d_u = d_{model} / h$, $d$ is the feature dimension and $h$ is the number of heads.

Stacking the Polar ray features $p_{n,u,w} \in \mathbb{R}^{R_u \times C}$ along azimuth axis, we obtain the Polar feature map (i.e., BEV Polar representation) $P_{n,u}$ for $n$th camera and $u$th scale as:

$$
P_{n,u} = \text{Stack}(P_{n,u,1}, \ldots, P_{n,u,W_{u}}),
$$

where $W_{u}$ denotes the azimuth dimension. This sequence-to-sequence cross-attention-based encoder can encode geometric imaging prior and implicitly learn a proxy for depth efficiently. Next, we show how to integrate independent Polar rays from multiple cameras into a coherent and structured BEV map.

**Polar Alignment across Multiple Cameras**

Our Polar alignment module transforms Polar rays from different camera coordinates to a shared world coordinate. Taking multi-view Polar feature maps $\{P_{n,u}\}_{n=1}^{N}$ and camera matrix $\{\Pi_n, E_n\}_{n=1}^{N}$ as inputs, it produces a coherent BEV Polar map $G_u \in \mathbb{R}^{R_u \times N_u \times C}$, covering all camera views, where $R_u$, $N_u$ and $C$ are the dimensions of radius, azimuth and feature. Concretely, it first generates a set of 3D points in the cylindrical coordinate uniformly, denoted by $G^{(P)} = \{(p_i^{(P)}, \phi_i^{(P)}, z_i^{(P)}) | i = 1, \ldots, R_u; j = 1, \ldots, N_u; k = 1, \ldots, Z_u\}$, where $Z_u$ is the number of points along $z$ axis. Since cylindrical coordinate and Polar coordinate share radius and azimuth axis, their superscripts are both denoted with $P$. The points are then projected to the image plane of
Figure 2: Schematic illustration of our proposed **PolarFormer** for multi-camera 3D object detection. For each image captured by any camera view, our model first extracts the feature maps at multiple spatial scales. Given such a feature map, the cross-plane encoder (a) then transforms all the feature columns to a set of Polar rays in a sequence-to-sequence manner via polar queries based cross-attention. The polar rays from all the cameras are subsequently processed by a Polar alignment module (b) to generate a structured multi-scale Polar BEV map, followed by further enhancement via interactions among different scales using a Polar BEV encoder (c). At last, a specially designed Polar Head decodes multi-scale Polar BEV features for making final predictions in the Polar coordinate.

The \( n \)th camera to retrieve the index of Polar ray, estimated by:

\[
\begin{bmatrix}
sx_{i,j,k,n}^{(I)} \\
sy_{i,j,k,n}^{(I)} \\
1
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 1 \\
\Pi_n & 0 & 1 \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
\rho_i^{(P)} \\
\phi_j^{(P)} \\
z_k^{(P)}
\end{bmatrix},
\]

(7)

where \( s \) is the scale factor. Coherent BEV Polar map of \( u \)th scale can be then generated by:

\[
G_u(\rho_i^{(P)}, \phi_j^{(P)}) = \frac{1}{\sum_{n=1}^{N} \sum_{k=1}^{Z} \lambda_n(\rho_i^{(P)}, \phi_j^{(P)}, z_k^{(P)})} 	imes \sum_{n=1}^{N} \sum_{k=1}^{Z} B(P_{n,u}, (\Pi(I))_{i,j,k,n})
\]

(8)

where \( \lambda_n(\rho_i^{(P)}, \phi_j^{(P)}, z_k^{(P)}) \) is binary weighted factor indicating visibility in \( n \)th camera, \( B(P, (x, y)) \) denotes the bi-linear sampling \( P \) at location \( (x, y) \), \( \Pi(I)_{i,j,k,n} \) and \( \tau_{i,j,k,n} \) denote the normalized Polar ray index and radius index. Note the radius \( r \) is the distance between the point and the camera origin in BEV. Our Polar alignment module incorporates the features at different heights by generating the points along z axis. As validated in Table 2a, learning Polar representation is superior over Cartesian coordinate due to minimal information loss and higher consistency with raw visual data.

**Polar BEV Encoder at Multiple Scales**

We leverage multi-scale feature maps for handling object scale variance in the Polar coordinate. To that end, the BEV Polar encoder performs information exchange among neighbouring pixels and across multi-scale feature maps. Formally, let \( \{G_u\}_{u=1}^{U} \) be the input multi-scale Polar feature maps and \( \hat{x}_q \in [0, 1]^2 \) be the normalized coordinates of the reference points for each query element \( q \), we introduce a multi-scale deformable attention module (Zhu et al. 2020) as:

\[
\text{MSDeformAttn}(z_q, x_q, \{G_u\}_{u=1}^{U}) = \sum_{m=1}^{M} \sum_{u=1}^{U} \sum_{k=1}^{K} A_{muqk} W'_m G_u(s_u(\hat{x}_q) + \Delta x_{muqk})
\]

(9)

where \( m \) and \( k \) are the index of the attention head and the sampling point. \( z_q \) is the query feature. \( \Delta x_{muqk} \) and \( A_{muqk} \) denote the sampling offset and the attention weight.
of the $k$th sampling point in $u$th feature level and $m$th attention head. The attention weight $A_{muk}$ is normalized by $\sum_{u=1}^{U} \sum_{k=1}^{K} A_{muk} = 1$. Sampling offsets $\Delta x_{muk}$ are generated by applying MLP layers on query $q$. Function $\zeta_{u}$ generates the sampling offsets and rescales the normalized coordinate $\tilde{x}_u$ to the $u$th feature scale. $W_m$ and $W'_m$ are learnable parameters. Serving as query, each pixel in the multi-scale feature maps exploits the information from both neighbouring pixels and pixels across scales. This enables learning richer semantics across all feature scales.

**Polar BEV Decoder at Multiple Scales**

The Polar decoder decodes the above multi-scale Polar features to make predictions in the Polar coordinate. We construct the Polar BEV decoder with deformable attention (Zhu et al. 2020). Specifically, we query $q$ in Eq. (9) as learnable parameters.

Unlike 2D reference points in the encoder, here the reference points are in 3D cylindrical coordinate, equal to Polar coordinate when projected to BEV. The classification branch in each decoder layer outputs the confidence score vector $c \in \mathbb{R}^C$, where $C$ is the number of categories. The key learning targets of regression branch are in polar coordinate instead of Cartesian coordinate, as illustrated in Figure 3.

For simplicity, superscript $(P)$ is omitted. Reference points $(\rho, \phi, z)$ are iteratively refined in the decoder. With reference points, the regression branch regresses the offsets $d_\rho, d_\phi$ and $d_z$. The learning targets for orientation $\theta$ and velocity $v$ are relative to azimuths of objects and separated to orthogonal components $\theta_\phi, \theta_\rho, v_\phi$ and $v_\rho$, defined by:

$$\tilde{\theta}_\phi = \theta_\phi - \phi, \quad \theta_\phi = \sin \tilde{\theta}_\phi, \quad \theta_\rho = \cos \tilde{\theta}_\phi, \quad (10)$$

$$\theta_v = \theta_v - \phi, \quad v_\phi = v_{abs} \sin \tilde{\theta}_\phi, \quad v_\rho = v_{abs} \cos \tilde{\theta}_\phi. \quad (11)$$

Here, $\theta_\phi$ is the yaw angle of the bounding box. $v_{abs}$ and $\theta_v$ are the absolute value and angle of velocity. We regress the object size $l, w$ and $h$ as $\log l, \log w$ and $\log h$. We adopt Focal loss (Lin et al. 2017) and L1 loss for classification and regression respectively.

**Experiments**

**Dataset** We evaluate the PolarFormer on the nuScenes dataset (Caesar et al. 2020). It provides images with a resolution of $1600 \times 900$ from 6 surrounding cameras (Figure 1). The total of 1000 scenes, where each sequence is roughly 20 seconds long and annotated every 0.5 second, is split...
officially into train/val/test set with 700/150/150 scenes.

**Implementation details** We implement our approach based on the codebase mmdetection3d (Contributors 2020). Following DETR3D (Wang et al. 2022b) and FCOS3D (Wang et al. 2021), a ResNet-101 (He et al. 2016), with 3rd and 4th stages equipped with deformable convolutions is adopted as the backbone architecture. The number of cross-plane encoder layer is set to 3 for each feature scale. The resolution of radius and azimuth for our multi-scale. The resolution of radius and azimuth for our multi-scale Polar BEV maps are (64, 256), (32, 128), (16, 64) respectively. We use 6 Polar BEV encoder and 6 decoder layers. Following DETR3D (Wang et al. 2022b), our backbone is initialized from a checkpoint of FCOS3D (Wang et al. 2021) trained on nuScenes 3D detection task, while the rest is initialized randomly. We use above setting for prototype verification. To fully leverage the sequence data, we further conduct temporal fusion between the current frame and one history sweep in the BEV space. Following BEVDet4D (Huang and Huang 2022), we simply concatenate two temporally adjacent multi-scale Polar BEV maps along the feature dimension and feed to the BEV Polar encoder. We randomly sample a history sweep from [3T; 27T] during training, and sample the frame at 15T for inference. T (≈ 0.083s) refers to the time interval between two sweep frames. We term our temporal version as PolarFormer-T.

**Training** Following DETR3D (Wang et al. 2022b) we train our models for 24 epochs with the AdamW optimizer and cosine annealing learning rate scheduler on 8 NVIDIA V100 GPUs. The initial learning rate is $2 \times 10^{-4}$, and the weight decay is set to 0.075. Total batch size is set to 48 across six cameras. Synchronized batch normalization is adopted. All experiments use the original input resolution. Note our image variant uses the same amount of training data as the monocular pipelines (Wang et al. 2021) and the multi-camera counterparts (Wang et al. 2022b; Li et al. 2022b). Multi-camera and monocular paradigms share the same evaluation metrics.

**Inference** We evaluate our model on nuScenes validation set and test server. We do not adopt model-agnostic trick such as model ensemble and test time augmentation.

**Comparison with the State of the Art**

We compare our method with the state of the art on both test and val sets of nuScenes. In addition to the (i) prototype setting mentioned in implementation details, we also evaluate our model in the (ii) improved setting, with VoVNet (V2-99) (Lee et al. 2019) as backbone architecture with a pretrained checkpoint from DD3D (Park et al. 2021) (fine-tuned on extra DDAD15M (Guizilini et al. 2020) dataset) to boost performance.

Table 1 compares the results on nuScenes test set. We observe that our PolarFormer achieves the best performance under both the (i) prototype and (ii) improved setting in terms of mAP and NDS metrics, indicating the superiority of learning representation in the Polar coordinate. With temporal information PolarFormer-T can further boost performance substantially. Additional experiments results on val set and qualitative results are shown in supplementary materials.

**Ablation Studies**

We conduct a series of ablation studies on nuScenes val set to validate the design of PolarFormer. Each proposed component and important hyperparameters are examined thoroughly.

**Polar v.s. Cartesian** Table 2a ablates the coordinate system. We make several observations: (I) Learning the representation and making the prediction both on Cartesian, Centerpoint (Yin, Zhou, and Krähenbühl 2021) gives a strong baseline with 0.378 mAP and 0.454 NDS; After applying circle NMS, Centerpoint* can further improve; (II)}
Figure 5: 3D object detection in (a) Cartesian BEV vs. (b) Polar BEV, and (c) Performance comparison (mAP/NDS) at three distances (Near/Medium/Far). Red and green boxes show the same objects in different coordinates.

Our PolarFormer-CC (with Cartesian feature and prediction) outperforms the Centerpoint equipped with specially designed CBGS head (Yin, Zhou, and Krähenbühl 2021); (III) When Polar BEV map is used to feed into a Cartesian decoder head, on-par performance is achieved with the post-processing counterpart; (IV) PolarFormer, our full model that predicts all 10 categories with one head and without any post-processing procedure, exceeds highly optimized Centerpoint by 1.1% in mAP and 0.2% in NDS. This suggests the significance of Polar in both representation learning and exploitation (i.e., decoding).

Visualizations With the quantitative evaluation in Figure 5, (I) it is evident that our model in Polar coordinate yields better results than Cartesian consistently. Specifically, Polar outperforms Cartesian by mAP 3.1% and NDS 0.3% in nearby area, mAP 1.3% and NDS 0.7% in medium area. (II) As shown in Figure 5a and Figure 5b, compared to the Polar map, Cartesian usually downsamples the nearby area (red) with information loss, while upsamples the distant area (green) without actual information added. This would explain the inferiority of Cartesian. (III) Figure 6 shows the attention map of the decoder query in multi-scale Polar BEV features. For better viewing, we resize the multi-scale features into the same resolution. The bottom/top corresponds to the largest/smallest scale features. The y axis represents the radius of the Polar map. It is shown that larger objects represented in the small map (top) are close to the ego car (small radius) whilst small objects in the large map (bottom) distribute through the distant area. This is highly consistent with the geometry structure of raw images (Figure 1), which has shown to be a more effective coordinate for 3D object detection as above.

Architecture (I) We first evaluate three designs of positional embedding (PE): 2D learnable PE, fixed Sine PE, and 3D PE (generated based on a set of 3D points for each Polar ray position). (II) As our cross-plane encoder transforms different levels of feature from FPN into Polar rays independently, we can fuse the multi-level features into a single BEV or naturally shape multiple BEVs with different or same resolutions; Table 2c clearly shows that a model with multi-scale Polar BEVs outperforms the single-scale counterpart under either coordinate. In contrast, little performance gain is achieved from multi-scale features in Cartesian. This suggests that object scale variation is a unique challenge with Polar, but absent with Cartesian. Our design consideration is thus verified. (III) We study the resolution of polar map by adjusting the azimuth $N_1$ and radius $R_1$ (the number of Polar query in the cross-plane encoder); Table 2d shows that the angle with 256 and radius with 64 gives the best performance.

Conclusions We have proposed the Polar Transformer (PolarFormer) for 3D object detection in multi-camera 2D images from the ego car’s perspective. With a rasterized BEV Polar representation geometrically aligned to visual observation, PolarFormer overcomes irregular Polar grids by a cross-attention based decoder. Further, a multi-scale representation learn-
ing strategy is designed for tackling the intrinsic object scale variation challenge. Extensive experiments on the nuScenes dataset validate the superiority of our PolarFormer over previous alternatives on 3D object detection.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant No. 2018AAA0102803, 2018AAA0102802, 2018AAA0102800), the Natural Science Foundation of China (Grant No. 621002439, U22B2056, 61972394, 62036011, 62192782, 61721004, 62102417), Lingang Laboratory (Grant No. LG-QS-202202-07), Natural Science Foundation of Shanghai (Grant No. 22ZR1407500) Beijing Natural Science Foundation (Grant No. L223003, JQ22014), the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (Grant No. 2017KZDXM081, 2018KZDXM066), Guangdong Provincial University Innovation Team Project (Grant No. 2020KICXTD045). Jin Gao was also supported in part by the Youth Innovation Promotion Association, CAS.

References

Bewley, A.; Sun, P.; Mensink, T.; Anguelov, D.; and Sminchisescu, C. 2020. Range conditioned dilated convolutions for scale invariant 3d object detection. arXiv preprint arXiv:2005.09927.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A multimodal dataset for autonomous driving. In CVPR, 11621–11631.

Can, Y. B.; Liniger, A.; Paudel, D. P.; and Van Gool, L. 2021. Structured Bird’s-Eye-View Traffic Scene Understanding from Onboard Images. In ICCV, 15661–15670.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In ECCV, 213–229.

Chen, Q.; Vora, S.; and Beijbom, O. 2021. PolarStream: Streaming Object Detection and Segmentation with Polar Pillars. In NeurIPS, 26871–26883.

Chitta, K.; Prakash, A.; and Geiger, A. 2021. NEAT: Neural Attention Fields for End-to-End Autonomous Driving. In ICCV, 15793–15803.

Contributors, M. 2020. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d. Accessed: 2023-03-03.

Ding, M.; Huo, Y.; Yi, H.; Wang, Z.; Shi, J.; Lu, Z.; and Luo, P. 2020. Learning depth-guided convolutions for monocular 3d object detection. In CVPR, 1000–1001.

Guizilini, V.; Ambrus, R.; Pillai, S.; Raventos, A.; and Gaidon, A. 2020. 3D Packing for Self-Supervised Monocular Depth Estimation. In CVPR, 2485–2494.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In CVPR, 770–778.

Hu, A.; Murez, Z.; Mohan, N.; Dudás, S.; Hawke, J.; Badrinarayanan, V.; Cipolla, R.; and Kendall, A. 2021. FIERY: Future Instance Prediction in Bird’s-Eye View From Surround Monocular Cameras. In ICCV, 15273–15282.

Huang, J., and Huang, G. 2022. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054.

Lee, Y.; Hwang, J.-w.; Lee, S.; Bae, Y.; and Park, J. 2019. An energy and GPU-computation efficient backbone network for real-time object detection. In CVPR workshops.

Li, Q.; Wang, Y.; Wang, Y.; and Zhao, H. 2022a. Hdmapnet: An online hd map construction and evaluation framework. In ICRA, 4628–4634.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022b. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In ECCV, 1–18.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In ICCV, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In ECCV, 740–755.

Lu, J.; Zhou, Z.; Zhu, X.; Xu, H.; and Zhang, L. 2022. Learning Ego 3D Representation as Ray Tracing. In ECCV, 129–144.

Ma, X.; Wang, Z.; Li, H.; Zhang, P.; Ouyang, W.; and Fan, X. 2019. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In ICCV, 6851–6860.

Park, D.; Ambrus, R.; Guizilini, V.; Li, J.; and Gaidon, A. 2021. Is Pseudo-Lidar needed for Monocular 3D Object detection? In ICCV, 3142–3152.

Philion, J.; and Fidler, S. 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In ECCV, 194–210.

Rapoport-Lavie, M.; and Raviv, D. 2021. It’s All Around You: Range-Guided Cylindrical Network for 3D Object Detection. In ICCV, 2992–3001.

Reading, C.; Harakeh, A.; Chae, J.; and Waslander, S. L. 2021. Categorical depth distribution network for monocular 3d object detection. In CVPR, 8555–8564.

Roddick, T.; and Cipolla, R. 2020. Predicting semantic map representations from images using pyramid occupancy networks. In CVPR, 11138–11147.

Roddick, T.; Kendall, A.; and Cipolla, R. 2019. Orthographic feature transform for monocular 3d object detection. In BMVC.

Saha, A.; Maldonado, O. M.; Russell, C.; and Bowden, R. 2022. Translating Images into Maps. In ICCVR, 9200–9206.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In ICCV, 9627–9636.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In NeurIPS, volume 30.
Wang, T.; Xinge, Z.; Pang, J.; and Lin, D. 2022a. Probabilistic and geometric depth: Detecting objects in perspective. In CoRL, 1475–1485.

Wang, T.; Zhu, X.; Pang, J.; and Lin, D. 2021. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In ICCV, 913–922.

Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In CVPR, 8445–8453.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022b. Det3d: 3d object detection from multi-view images via 3d-to-2d queries. In CoRL, 180–191.

Xie, E.; Yu, Z.; Zhou, D.; Philion, J.; Anandkumar, A.; Fidler, S.; Luo, P.; and Alvarez, J. M. 2022. M* 2BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation. arXiv preprint arXiv:2204.05088.

Xu, B.; and Chen, Z. 2018. Multi-level fusion based 3d object detection from monocular images. In CVPR, 2345–2353.

Yang, W.; Li, Q.; Liu, W.; Yu, Y.; Ma, Y.; He, S.; and Pan, J. 2021. Projecting Your View Attentively: Monocular Road Scene Layout Estimation via Cross-view Transformation. In CVPR, 15536–15545.

Yin, T.; Zhou, X.; and Krähenbühl, P. 2021. Center-based 3D Object Detection and Tracking. In CVPR, 11784–11793.

You, Y.; Wang, Y.; Chao, W.-L.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. arXiv preprint arXiv:1906.06310.

Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; and Foroosh, H. 2020. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In CVPR, 9601–9610.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as points. arXiv preprint arXiv:1904.07850.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection.

Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Ma, Y.; Li, W.; Li, H.; and Lin, D. 2021. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In CVPR, 9939–9948.