Exploring Surround-View Fisheye Camera 3D Object Detection

Changcai Li[1,2], **Wenwei Lin**[1], **Zuoxun Hou**[3], **Gang Chen**[1*], **Wei Zhang**[2*], **Huihui Zhou**[2], **Weishi Zheng**[1]

[1]Sun Yat-sen University
[2]Pengcheng Laboratory
[3]Beijing Institute of Space Mechanics and Electricity
cheng83@mail.sysu.edu.cn, zhangwei1213052@126.com

Abstract

In this work, we explore the technical feasibility of implementing end-to-end 3D object detection (3DOD) with surround-view fisheye camera system. Specifically, we first investigate the performance drop incurred when transferring classic pinhole-based 3D object detectors to fisheye imagery. To mitigate this, we then develop two methods that incorporate the unique geometry of fisheye images into mainstream detection frameworks: one based on the bird's-eye-view (BEV) paradigm, named FisheyeBEVDet, and the other on the query-based paradigm, named FisheyePETR. Both methods adopt spherical spatial representations to effectively capture fisheye geometry. In light of the lack of dedicated evaluation benchmarks, we release Fisheye3DOD, a new open dataset synthesized using CARLA and featuring both standard pinhole and fisheye camera arrays. Experiments on Fisheye3DOD show that our fisheye-compatible modeling improves accuracy by up to 6.2% over baseline methods.

Code — https://github.com/weiyangdaren/Fisheye3DOD

Introduction

Reliable 360° perception is vital for autonomous systems like self-driving vehicles and various robots. Surround-view fisheye cameras enable this via a compact multi-camera setup, as shown in Figure 1 (**Right**), where four ultra-wide lenses (each exceeding 180° field of view (FoV)) capture the full surroundings seamlessly. Existing fisheye-based works focus on depth estimation (Won, Ryu, and Lim 2020; Xie, Wang, and Liu 2023) and segmentation (Deng et al. 2019; Playout et al. 2021), but 3DOD—a crucial task for dynamic obstacle avoidance—remains unexplored.

Compared to current vision-centric 3D perception systems that mainly use multi-pinhole-camera setups (Caesar et al. 2020; Sun et al. 2020), fisheye camera-based solutions offer several distinctive advantages. **First**, because of regulatory requirements, such as the 2018 U.S. mandate to prevent reversing accidents through the use of rear-view fisheye cameras (Sunstein 2019), modern mass-produced vehicles are widely equipped with such cameras (e.g., BMW (Hughes et al. 2009)), as illustrated in Figure 1 (**Left**). This allows direct use of the pre-installed sensors for perception, avoiding costly retrofits needed by pinhole systems. **Second**, compared to methods (Ge et al. 2023; Yan et al. 2023; Xie et al. 2025) that rely on algorithms to improve robustness against sensor failures, a surround-view fisheye system inherently provides physical redundancy via overlapping FoV. As shown in Figure 1 (**Right**), this overlap captures the same object from multiple viewpoints to enhance reliability. **Finally**, the ultra-wide FoV suits scenarios with constrained space or cost-sensitive deployment, such as indoor robotics or surveillance. Standard pinhole setups usually require more cameras for similar coverage (e.g., nuScenes (Caesar et al. 2020) uses six, Tesla autopilot (Tatarek, Kronenberger, and Handmann 2017) uses eight).

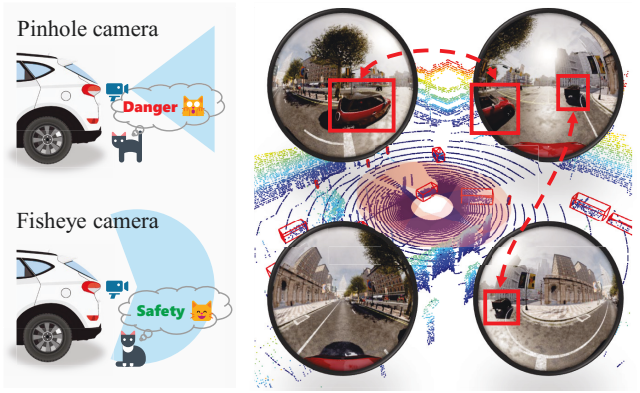


Figure 1: **Left:** The pinhole camera setup has blind spots in the near field, whereas the fisheye camera provides enhanced coverage for improved safety. **Right:** The same object is captured from multiple fisheye viewpoints.

Despite their advantages, fisheye cameras pose inherent challenges that can degrade detection performance. In particular, the non-linear projection compresses objects into very few pixels, making them harder to detect reliably. This issue is fundamental to fisheye imagery and motivates the following Research Questions (**RQs**): *1) how much accuracy is lost when transferring pinhole-based detectors to fisheye images*, and *2) how can the transfer be made more effective*? Due to the absence of a unified benchmark, prior

*Corresponding authors.

studies (Plaut, Ben Yaacov, and El Shlomo 2021; Yogamani et al. 2024a,b) mostly focus on evaluating methods within their respective imaging models—either on pinhole only or on fisheye only—without directly comparing the two. As a result, these **RQs** remain open. Answering these questions is vital both for advancing fisheye 3D perception and for guiding industrial design and sensor choices, where camera trade-offs affect cost, coverage, and reliability.

To address the **RQ1**, we create **Fisheye3DOD**, a synthetic dataset based on the CARLA simulator (Dosovitskiy et al. 2017). Fisheye3DOD provides synchronized multi-view data from both six surround pinhole cameras and four fisheye cameras under identical scenarios, enabling a direct and fair comparison between pinhole and fisheye imaging models for 3DOD task. Then, we leverage the Fisheye3DOD dataset to systematically evaluate representative pinhole-based 3D detectors on fisheye data after rectification, including both perspective and cylindrical projections.

To address the **RQ2**, we introduce spherical back-projection at the feature level to resolve the geometric incompatibilities between fisheye and pinhole models. Current pinhole-based 3D detectors, such as BEVDet (Huang et al. 2021) with explicit BEV construction or PETR (Liu et al. 2022) with implicit 3D query encoding, rely on perspective projection assumptions that are incompatible with fisheye optics' nonlinear distortions. To overcome this limitation, we propose two geometry-aligned frameworks that integrate spherical back-projection into mainstream detection pipelines. Specifically, we introduce **FisheyeBEVDet** and **FisheyePETR**, both of which project image features onto a spherical equirectangular representation. Then, FisheyeBEVDet adapts the BEV-based architecture by performing depth reasoning in spherical coordinates. FisheyePETR, on the other hand, employs spherical ray-based positional encodings to enhance projected features, which then interact with object queries. These two designs address the unique geometric distortions of fisheye cameras while maintaining compatibility with established 3D detection paradigms.

Experiments on Fisheye3DOD demonstrate that directly transferring pinhole-based detectors to rectified fisheye images leads to a significant accuracy drop. Through end-to-end optimization with spherical modeling at the feature level, our proposed FisheyeBEVDet and FisheyePETR effectively mitigate the performance gap, improving detection accuracy by 4.5 and 6.2 points, respectively.

In summary, our key contributions are as follows:

- To our knowledge, we present the first systematic and quantitative study comparing 3D perception performance between pinhole and fisheye imaging.
- We introduce Fisheye3DOD, a benchmark specifically designed to enable direct comparison between pinhole and fisheye cameras in the same driving environments.
- We propose two frameworks, FisheyeBEVDet and FisheyePETR, which leverage spherical modeling tailored for fisheye optics to improve performance.
- Extensive experiments on Fisheye3DOD validate our findings and demonstrate significant performance gains over direct transfer baselines.

Related Work

Multi-View 3D Object Detection. Most existing multi-view 3DOD methods rely on pinhole images as input, which can be broadly divided into two groups (Mao et al. 2023):

1) *BEV-based methods* (Huang et al. 2021; Huang and Huang 2022; Li et al. 2023b,a, 2024, 2023c) construct 3D detection spaces through Lift-Splat-Shoot (LSS) (Philion and Fidler 2020), which lifts 2D features into 3D via depth-context outer products and projects them onto BEV grids. To improve the accuracy of depth estimation, BEVDepth (Li et al. 2023b) and BEVStereo (Li et al. 2023a) introduce explicit supervision and temporal stereo cues, respectively. Additionally, some works focus on improving BEV features, such as bidirectional projection compensation in FB-BEV (Li et al. 2023c) and CRF-modulated depth estimation in BEVNeXt (Li et al. 2024; Liu, Salzmann, and He 2014).

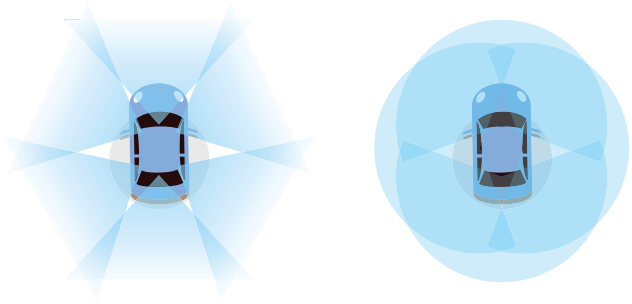
2) *Query-based methods* (Wang et al. 2022; Liu et al. 2022; Li et al. 2022b; Yang et al. 2023; Liu et al. 2023; Wang et al. 2023) utilize DETR's sparse 3D queries (Carion et al. 2020) projected onto multi-view images for feature sampling, with Transformer layers (Vaswani 2017) decoding bounding boxes. Building upon the query-based paradigm, some works (Liu et al. 2022, 2023; Wang et al. 2023) simplify projection via 3D positional encoding, while others (Li et al. 2022b; Yang et al. 2023) replace object queries with dense BEV-centric queries that aggregate features through spatio-temporal attention.

Fisheye Dataset. The majority of existing fisheye datasets suffer from limited availability and accessibility of annotations. WoodScape(Yogamani et al. 2019) and SynWoodScape(Sekkat et al. 2022) pioneered multi-task learning for fisheye imagery. However, WoodScape does not provide 3D annotations, and SynWoodScape, although it includes them, releases only about 500 samples, which is insufficient for training modern detectors. Real-world datasets such as FisheyeCityscapes(Ye et al. 2020) and Fisheye8K (Gochoo et al. 2023) are limited by task-specific constraints, with the former focusing on semantic segmentation and the latter designed for object detection. For synthetic alternatives, OmniScape (Sekkat et al. 2020) caters to stereo semantic segmentation, whereas Synthetic Urban/Deep360 (Won, Ryu, and Lim 2019; Li et al. 2022a) target omnidirectional depth estimation. Moreover, recent benchmarks (Samani et al. 2023; Yogamani et al. 2024a,b) are tailored for BEV semantic segmentation. Overall, these datasets either lack sufficient 3D annotations, are not publicly available due to commercial constraints, or do not simultaneously provide surround-view pinhole and fisheye data necessary for investigating our **RQ1**.

Monocular Fisheye 3D Object Detection. To our knowledge, the work by Plaut et al. (Plaut, Ben Yaacov, and El Shlomo 2021) is the only study that addresses 3D object detection using fisheye images. Their method warps fisheye inputs into cylindrical projections that resemble perspective views, allowing monocular detectors trained on pinhole data to be applied. However, it operates on single-view inputs without modeling multi-view feature interactions and does not report performance using pinhole images under the same environment. As a result, it does not address **RQ1**.

Dataset	Type	Scenes	Available	Image Properties			Annotation Details		
				Lens	Frames	Resolution	Class	Anno.	3D boxes
nuScenes	Real	1000	✓	**P**	1.4M	1600 × 900	23	40k	1.4M
Waymo Open	Real	1150	✓	**P**	1M	1920 × 1080	4	200k	12M
WoodScape	Real	N/A	no 3D	**F**	10k	1280 × 966	3	10k	N/A
SynWoodScape	Sim.	N/A	500 samples	**F**	10k	1280 × 966	3	10k	N/A
Fisheye8k	Real	22	no 3D	**F**	8k	1280 × 1280	5	8k	✗
Cognata	Sim.	5	✗	**F**	50k	1920 × 1208	5	12k	✗
Fisheye3DOD (Ours)	Sim.	8	✓	**P** **F**	432k 288k	1280 × 720 800 × 800	6	72k	607k

Table 1: A summary of the proposed Fisheye3DOD dataset and other published datasets, where Bold letters **P** and **F** denote pinhole and fisheye images, respectively.



Layout of six pinhole cameras Layout of four fisheye cameras

Figure 2: Camera layouts for surround-view perception.

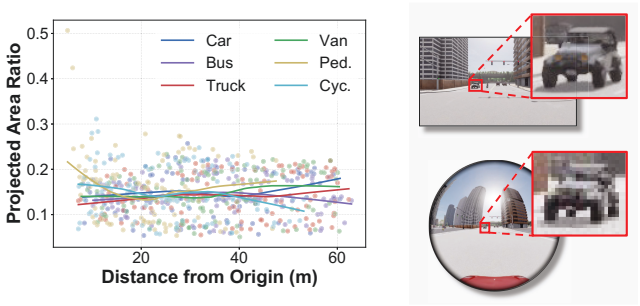


Figure 3: **Left:** The horizontal axis indicates the 3D distance from the ego vehicle. The vertical axis indicates the ratio between the largest projected 2D bounding box size of an object in any fisheye camera and that in any pinhole camera. The points in the figure correspond to 100 samples per category, with the curves fitted using LOWESS (Cleveland 1979). **Right:** Illustration of pixel compression in fisheye and pinhole images. The same object occupies approximately 70 × 80 pixels in the pinhole image, but only about 22 × 26 pixels in the fisheye image. The pixel area in the fisheye image is roughly 0.1 times that of the pinhole image.

Fisheye3DOD Dataset

Data Collection

To address the existing gap in fisheye 3D object detection datasets, we developed Fisheye3DOD, a synthetic benchmark created through the CARLA simulator (Dosovitskiy et al. 2017). This dataset comprises 144 driving sequences covering urban and suburban environments, spanning diverse illumination conditions (daytime noon, sunset, night) and weather patterns (clear, cloudy, rainy). Each scenario contains temporally aligned sensor data captured at 10Hz over 50-second episodes, yielding a total of 500 frames per sequence. A detailed comparison between Fisheye3DOD and the existing dataset can be found in Table 1.

Our sensor configuration includes six surround-view pinhole cameras and four wide-angle fisheye cameras (FoV = 220°) (Won, Ryu, and Lim 2019) with corresponding 3D bounding box annotations. Moreover, we capture LiDAR point clouds, semantic LiDAR data, and high-precision ego-vehicle trajectories to support potential future work (Huang et al. 2023; Hu et al. 2023). The sensor pose of Fisheye3DOD adheres to industry-standard practices (Caesar et al. 2020; Won, Ryu, and Lim 2019), ensuring benchmark compatibility, as shown in Figure 2.

Noting that CARLA lacks native fisheye sensor support (Dosovitskiy et al. 2017), we mathematically model fisheye distortion via the Kannala-Brandt projection (Kannala and Brandt 2006). This projection model captures the non-linear relationship between incident angle θ and radial displacement r through a ninth-order polynomial:

$$r(\theta) = k_0\theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9 \quad (1)$$

where $k = \{k_i\}_{i=0}^{4}$ are commonly used as fisheye-specific distortion coefficients.

Challenges of Fisheye Images

The Figure 3 highlights a key challenge in fisheye imagery: pixel compression. Fisheye projection nonlinearly compresses wide-angle scenes into limited image regions, resulting in significantly fewer pixels per object compared to pinhole projection. In our dataset, objects in fisheye views occupy only about 15% of the pixel area of their counterparts in pinhole images, as shown in Figure 3 (**Left**). This leads to a substantial loss of spatial resolution and visual detail, making reliable detection more challenging. Note that this loss of information entropy during imaging is irreversible and cannot be recovered through fisheye rectification. As shown in Figure 3 (**Right**), even when magnified to the same scale, the

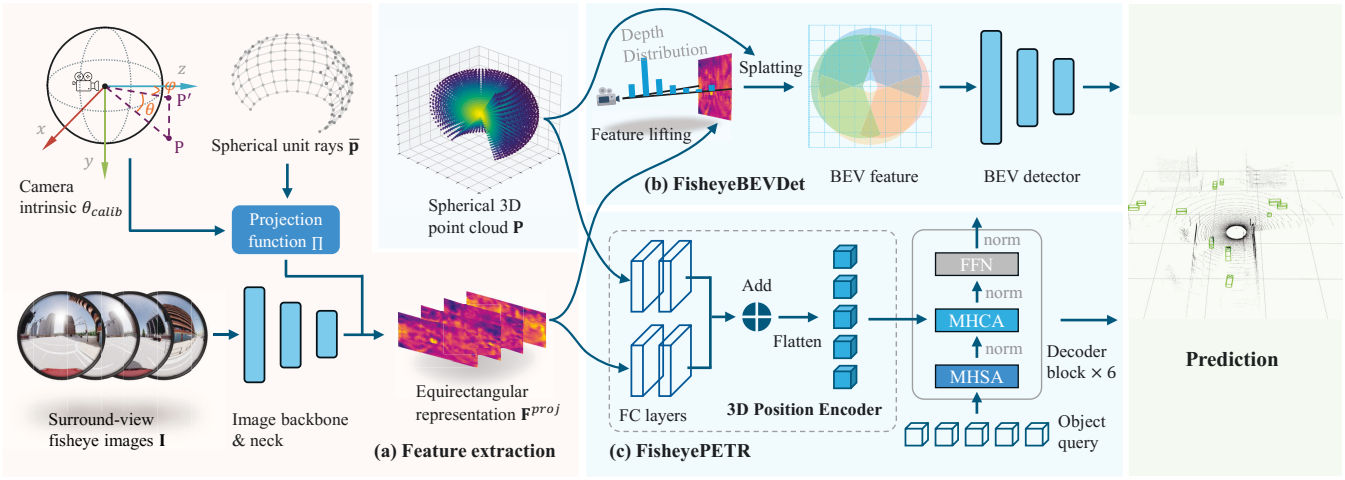


Figure 4: The architecture of the proposed methods. **(a):** Multi-view fisheye images are processed by a shared backbone, and their features are projected into an equirectangular representation via the projection function Π. **(b):** In FisheyeBEVDet, the projected 2D features are lifted onto a 3D spherical grid to construct a BEV representation. **(c):** In FisheyePETR, the 2D features are encoded with spherical coordinates and interact with object queries through multi-head cross-attention (MHCA).

object region in the fisheye image appears markedly blurrier than that in the pinhole, since no new information is introduced and the original pixel density is limited.

Evaluation Metrics

Our evaluation follows the nuScenes benchmark (Caesar et al. 2020) protocol, employing center-distance-based AP with decomposed True Positive ($\mathbb{TP}$) metrics: mATE (mean Average Translation Error), mASE (mean Average Scale Error), and mAOE (mean Average Orientation Error), to ensure standardized assessment. This aims to address the shortcomings of IoU-based metrics, where small-footprint objects receive a score of zero even under minor localization errors. The composite performance metric, named Fisheye Detection Score (FDS), is formulated as:

$$\text{FDS} = \frac{1}{6}\left[3\text{mAP} + \sum_{\text{mTP}\in\mathbb{TP}} (1 - \min(1, \text{mTP}))\right] \quad (2)$$

Given matching thresholds $\mathbb{D} = \{0.5, 1, 2, 4\}$ meters, the mean Average Precision (mAP) across all classes $\mathbb{C}$ is calculated as $\text{mAP} = \frac{1}{|\mathbb{C}||\mathbb{D}|}\sum_{c\in\mathbb{C}}\sum_{d\in\mathbb{D}} \text{AP}_{\text{c,d}}$.

Our Detector

Spherical Feature Representation

To fairly answer **RQ2**, we forgo the "bells and whistles" but instead pursue an end-to-end "tabula rasa" approach to investigate two distinct 3D detection paradigms: BEV-based and query-based. Both methods model 3D space in spherical coordinates and perform back-projection to establish image-to-space correspondence.

As illustrated in Figure 4, given a set of surround-view fisheye images $\mathbf{I} = \{\mathbf{I}_i \in \mathbb{R}^{H_\mathbf{I}\times W_\mathbf{I}\times 3}\}_{i=1}^{N}$, each image is directly fed into the backbone network (e.g., ResNet (He et al. 2016)) to extract 2D features $\mathbf{F}^{2d}$. The extracted features are then warped into an equirectangular representation via the calibrated fisheye projection function Π to align with the spherical 3D space. Formally, the projected feature map $\mathbf{F}^{proj} \in \mathbb{R}^{H\times W\times C}$ is computed as:

$$\mathbf{F}^{proj} = \mathbf{F}^{2d} \circ \mathbf{G}_{sph} \quad (3)$$

where $\circ$ denotes the differentiable warping operation, and $\mathbf{G}_{sph}$ is a precomputed sampling grid mapping 3D spherical direction vectors to image-plane coordinates via the projection function Π. Specifically, the grid is derived from the calibrated camera projection model (Scaramuzza, Martinelli, and Siegwart 2006) as:

$$\mathbf{G}_{sph} = \sigma\big(\Pi\left(\bar{\mathbf{p}}; \theta_{calib}\right)\big) \quad (4)$$

where $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$ denotes the projection function from 3D rays to image coordinates, parameterized by the camera calibration parameters θ_{calib}. The function σ normalizes the coordinates to the range $[-1, 1]$ for grid sampling. Each direction vector $\bar{\mathbf{p}} \in \mathbb{R}^3$ is parameterized by the azimuth and elevation angles (ϕ, θ) in spherical coordinates as:

$$\bar{\mathbf{p}} = [\cos\theta\cos\phi,\ \sin\theta,\ \cos\theta\sin\phi]^\top \quad (5)$$

Theoretically, $\mathbf{F}^{proj}$ can also be represented in cylindrical coordinates with $\bar{\mathbf{p}} = [\sin\phi,\ y,\ \cos\phi]^\top$. However, our experiments show that the spherical (equirectangular) representation is superior; please refer to Table 2.

BEV-based Detection

The projected features $\mathbf{F}^{proj}$ have now been aligned with unit directional vectors on the spherical surface. We next sample along each $\bar{\mathbf{p}}$ across discrete depth levels to realize the back-projection from image to 3D space. Specifically, for FisheyeBEVDet, we represent the BEV space using hierarchical spherical shells, as opposed to the parallel planar

stratification in LSS (Philion and Fidler 2020). Each shell discretizes the depth space aligned with the camera's viewing direction from the ego coordinate origin.

To define these concentric spherical shells, we first discretize the radial depth space into D uniformly spaced bins ranging from r_{min} to r_{max}. Let r_d denote the radial distance at the d-th depth level:

$$r_d = r_{min} + d \times \delta, \quad d \in [0, D-1]_{\mathbb{Z}} \tag{6}$$

where $\delta = \frac{r_{max} - r_{min}}{D}$ is the fixed interval between consecutive depth levels. Building upon the defined radial samples, a 3D point at the d-th shell along the unit direction vector $\bar{\mathbf{p}}$ corresponding to pixel location (h, w) is computed as:

$$\mathbf{p}^{cam}_{d,h,w} = r_d \times \bar{\mathbf{p}}_{h,w} \tag{7}$$

Each point in the camera coordinate system is then transformed into the unified LiDAR coordinate system using the camera-to-LiDAR transformation matrix $\mathbf{M} \in \mathbb{R}^{4\times4}$:

$$\mathbf{p}_{d,h,w} = \mathbf{M} \cdot \left[\left(\mathbf{p}^{cam}_{d,h,w}\right)^\top,\ 1 \right]^\top_{:3} \tag{8}$$

By applying this transformation to all points, we obtain the full point cloud $\mathbf{P} \in \mathbb{R}^{D\times H\times W\times 3}$. These points constitute multiple spherical shells that act as spatial anchors for the subsequent BEV feature projection.

Note that for each projected feature $\mathbf{f} \in \mathbf{F}^{proj}$, its association with a unit ray direction $\bar{\mathbf{p}}$ on the spherical surface has already been established through the warping process. Based on this, we can estimate a depth probability distribution along the corresponding ray $\bar{\mathbf{p}}$, enabling compatibility with the LSS paradigm. Specifically, given the feature $\mathbf{f}$, a fully-connected (FC) layer predicts a context vector $\mathbf{c} \in \mathbb{R}^C$ and a depth probability distribution $\alpha \in \Delta^{D-1}$, where $\Delta^{D-1} := \{\alpha \in \mathbb{R}^D \mid \alpha_d \geq 0, \sum_{d=0}^{D-1} \alpha_d = 1\}$. The depth-specific feature $\mathbf{c}_d \in \mathbb{R}^C$ at d-th shell is computed as:

$$\mathbf{c}_d = \alpha_d \cdot \mathbf{c} \tag{9}$$

where α_d denotes the probability of the feature being present at d-th shell.

Stacking $\mathbf{c}_d$ across all locations and depths yields a lifted feature volume $\mathbf{F}^{lift} \in \mathbb{R}^{D\times H\times W\times C}$. This volume is then projected into the BEV space using the corresponding 3D points $\mathbf{P}$, making it compatible with the downstream components of the original BEVDet framework.

Query-based Detection

As illustrated in Figure 4(c), FisheyePETR directly encodes the projected features $\mathbf{F}^{proj}$ with spherical coordinates and leverages object queries to interact with these features via MHCA, enabling end-to-end 3D object detection without explicit BEV representation.

Similar to FisheyeBEVDet, FisheyePETR is also required to construct spherical frustum points as spatial anchors to associate the projected features $\mathbf{F}^{proj}$ with their corresponding 3D locations. To align with PETR (Liu et al. 2022), FisheyePETR adopts quadratically increasing depth spacing instead of uniform intervals. The depth value r_d at the d-th level is computed as:

$$r_d = r_{min} + \frac{r_{max} - r_{min}}{D(D+1)} \times d(d+1) \tag{10}$$

Based on these depths, the 3D spherical frustum points are computed by combining each r_d with the corresponding unit directional vectors using the geometric mappings defined in Eq.7 and Eq.8. These points are then used as positional encodings, which are fused with the projected features $\mathbf{F}^{proj}$ to enhance 3D spatial awareness in subsequent processing.

Finally, FisheyePETR employs a detection transformer decoder, where object queries interact through self-attention and attend to the projected features via cross-attention, enabling end-to-end 3D detection.

Experiments

Implementation Details

The experiments are implemented on a single NVIDIA A6000 GPU platform. The dataset is split into training and testing sets based on scene sequences, using the first 70% of frames from each scene for training and the remaining 30% for testing. Similar to nuScenes (Caesar et al. 2020), the Fisheye3DOD dataset is sampled at 2Hz intervals throughout the experiments. The model is trained for 20 epochs with a batch size of 4. The AdamW optimizer (Loshchilov and Hutter 2017) is employed for parameter updates, configured with an initial learning rate of 0.0002 and weight decay of 0.01. The learning rate first undergoes a linear warm-up for the first 500 iterations, followed by a cosine annealing schedule (Loshchilov and Hutter 2016). The detection range spans a cuboid volume, defined by the bounds $\{(X, Y, Z) \mid X \in [-48, 48]\,\text{m}, Y \in [-48, 48]\,\text{m}, Z \in [-5, 5]\,\text{m}\}$. During the training phase, the dataset is loaded through a class-balanced sampler (CBGS (Zhu et al. 2019)) for effective mitigation of potential data imbalance.

Answering the Research Questions

Here, we evaluate the research questions posed earlier. The experimental results are presented in Table 2.

RQ1: How much accuracy is lost when transferring pinhole-based detectors to fisheye images? Table 2 presents the performance of representative pinhole-based 3D object detectors applied to fisheye data after standard rectification using perspective or cylindrical projection. Despite this preprocessing, both BEVDet and PETR suffer substantial accuracy drops compared to their original configurations on 6-camera pinhole images. Specifically, FDS decreases by over 12 points for both models, and other metrics also show clear degradation. This degradation stems from the intrinsic limitations of fisheye imaging. As previously discussed in the fisheye challenges, due to nonlinear projection compression, objects in fisheye images occupy only about 15% of the pixel area they would in pinhole images. This severe reduction in effective pixel density leads to irreversible information loss, which cannot be recovered through rectification and directly impacts detection performance.

RQ2: How can the transfer be made more effective? Table 2 compares our proposed methods, FisheyeBEVDet and FisheyePETR, against fisheye-input baselines which rely on image-level rectification, including both perspective and cylindrical projections. Both methods consistently outperform their rectified baselines by a significant margin

Methods	Camera	Rectification	FDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓
BEVDet	6 × **P**	–	0.563	0.506	0.458	0.161	0.520
BEVDet	4 × **F**	Perspective	0.440	0.304	0.588	0.177	0.505
BEVDet	4 × **F**	Cylindrical	0.453	0.322	0.591	0.178	0.478
FisheyeBEVDet	4 × **F**	Cylindrical	0.476	0.361	0.581	0.162	0.482
FisheyeBEVDet	4 × **F**	Equirectangular	0.485	0.382	0.591	0.164	0.480
PETR	6 × **P**	–	0.553	0.482	0.580	0.120	0.430
PETR	4 × **F**	Perspective	0.408	0.274	0.783	0.161	0.433
PETR	4 × **F**	Cylindrical	0.411	0.285	0.773	0.169	0.447
FisheyePETR	4 × **F**	Cylindrical	0.441	0.330	0.758	0.159	0.425
FisheyePETR	4 × **F**	Equirectangular	0.470	0.374	0.727	0.142	0.434

Table 2: Comparison of pinhole-based and fisheye-based methods. Bold letters **P** and **F** denote pinhole and fisheye inputs, respectively. Numeric prefixes (e.g., 6 ×) indicate camera count. Metric definitions are in the Fisheye3DOD Dataset section.

Methods	Camera	FDS ↑	mAP ↑
BEVDet	4 × **P** (w/o ↕)	0.370	0.206
FisheyeBEVDet	2 × **F** (↕)	0.454	0.324
FisheyeBEVDet	2 × **F** (↔)	0.431	0.315
FisheyeBEVDet	4 × **F**	0.485	0.382
PETR	4 × **P** (w/o ↕)	0.321	0.142
FisheyePETR	2 × **F** (↕)	0.421	0.289
FisheyePETR	2 × **F** (↔)	0.382	0.244
FisheyePETR	4 × **F**	0.470	0.374

Table 3: Evaluation of robustness under sensor failure and comparison across different fisheye camera layouts. Arrow directions (↕, ↔) indicate front-rear and left-right sensor arrangements, respectively.

Method	0-30 m		0-48 m	
	FDS ↑	mAP ↑	FDS ↑	mAP ↑
BEVDet	0.673	0.684	0.563	0.506
FisheyeBEVDet	0.586	0.555	0.485	0.382
PETR	0.652	0.634	0.553	0.482
FisheyePETR	0.564	0.516	0.470	0.374

Table 4: Detection performance (FDS and mAP) across cumulative distance ranges (0-30 m to 0-48 m).

across all key metrics. In particular, FisheyeBEVDet and FisheyePETR with equirectangular representation improve FDS by 4.5 and 6.2 points over the perspective-rectified baseline. This improvement stems from end-to-end modeling of fisheye geometry at the feature level, which preserves richer spatial and semantic information than image-level rectification. Moreover, they outperform their cylindrical counterparts by 0.9 and 2.9 points, respectively. This may be due to their more uniform angular sampling, particularly along the vertical direction.

It should be acknowledged that, despite these improvements, fisheye-based methods still lag behind pinhole detectors due to intrinsic imaging challenges. In our dataset, pinhole images provide nearly ten times the effective pixel area of fisheye images for objects. It is therefore unrealistic to expect fisheye-based detectors to achieve comparable accuracy while operating with significantly less spatial evidence.

Additional Analysis

To gain deeper insights, we conduct additional analysis and identify the following key Research Findings (**RFs**):

RF1 (System Robustness): Multi-fisheye systems inherently mitigate sensor failures via extensive FoV overlap. Recent works (Ge et al. 2023; Yan et al. 2023; Xie et al. 2025) have explored 3D detection under partial sensor failures. We argue that the extensive FoV overlap inherent in multi-fisheye setups naturally provides strong robustness against such failures. To validate this, Table 3 compares pinhole and fisheye configurations both missing front and rear cameras: specifically, BEVDet with four pinhole cameras without front-rear sensors (4 × **P** (w/o ↕)) versus FisheyeBEVDet with two fisheye cameras arranged left-right (2 × **F** (↔)). Similar comparisons are made for PETR variants. Results show that fisheye methods experience much smaller drops in FDS than pinhole counterparts when front and rear cameras are removed. This is because pinhole setups develop blind spots under these extreme conditions, while multi-fisheye setups maintain full coverage.

RF2 (Sensor Layout Impact): Front-rear sensor layouts outperform lateral ones, with full surround achieving the best results. To assess the effect of sensor placement in multi-fisheye systems, we compare three multi-fisheye layouts: front-rear, left-right, and full surround. As shown in Table 3, front-rear yields 2-4% higher FDS than left-right, since most traffic participants (e.g., cars, vans) cluster along the vehicle's longitudinal axis, allowing better shape preservation by reducing radial distortion. In contrast, the left-right layout pushes objects toward image edges, increasing pixel compression and detection difficulty. Moreover, the full surround layout further improves FDS by 3-5% over front-rear, achieving the highest accuracy. This improvement arises from multi-camera synergy: front-rear sensors optimize longitudinal coverage, while full surround mitigates lateral distortion by preserving complementary edge details.

RF3 (Distance Degradation): Fisheye camera capabilities align well with near-field sensing. Recent studies advocate for the use of fisheye cameras in near-field percep-

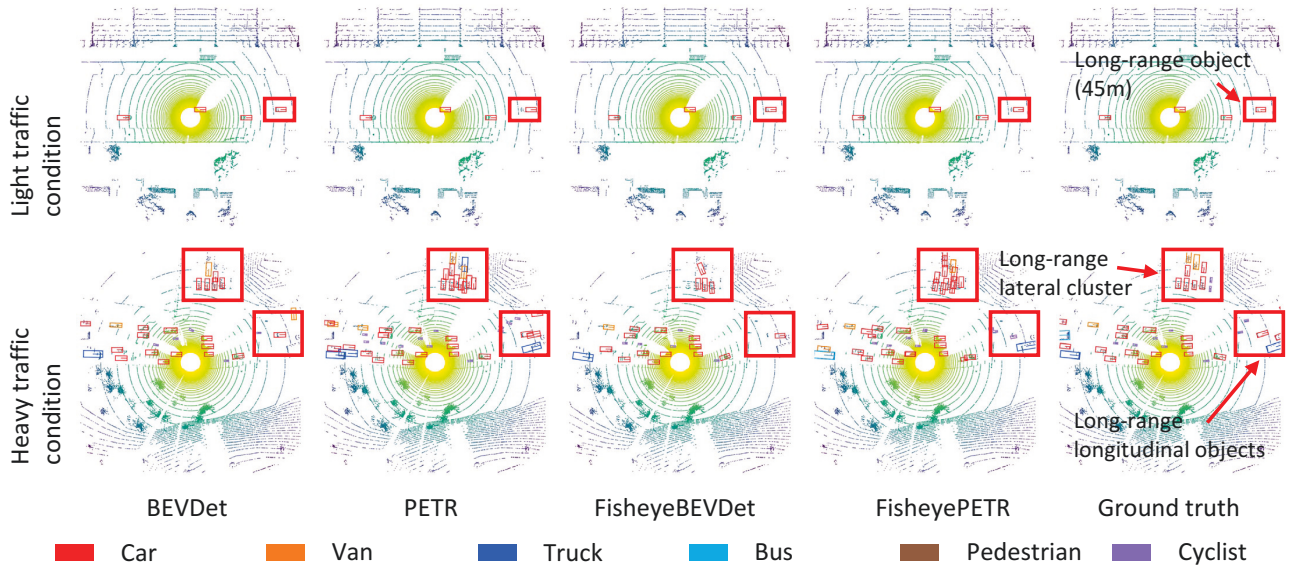


Figure 5: We visualize the predictions in LiDAR point clouds for a clearer comparison. The first row shows a sparse traffic scenario with isolated vehicles on open roads, whereas the second row depicts a dense urban junction with multi-object occlusion.

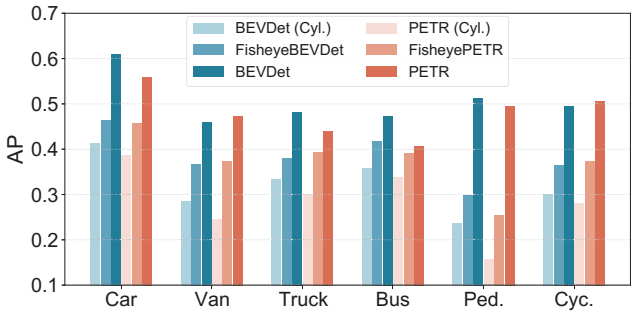


Figure 6: Per-class detection performance. Models with (Cyl.) use cylindrical rectified fisheye images as input.

tion. For example, F2BEV (Samani et al. 2023) and FisheyeBEVSeg (Yogamani et al. 2024b,a) adopt perception ranges of only 16 and 25 meters, respectively. Our experiments in Table 4 validate this, showing fisheye variants achieve 0.586 FDS at 0-30m, comparable to pinhole systems' 0.563 FDS at 0-48m — a critical range covering the under-30m braking distance at 60 km/h (Hosseinlou, Ahadi, and Hematian 2012). This capability makes fisheye cameras especially suitable for low-speed scenarios such as automated parking systems, warehouse robots, and sidewalk delivery robots.

RF4 (Failure Modes and Limitations): Small-footprint objects exacerbate challenges under fisheye distortion. Figure 6 compares per-class AP between our fisheye models, cylindrical rectified baselines, and their pinhole counterparts. We observe that small-footprint classes, such as *Pedestrian* and *Cyclist*, suffer the most significant performance drop when shifting input from pinhole to fisheye. This may be due to their inherently small size, which results in fewer visual cues being preserved under fisheye-induced pixel compression. In simulation environments, these issues are further compounded due to the limited texture richness. Mitigating this issue may benefit from insights in small object detection. Notably, our fisheye models significantly outperform their cylindrical baselines across all categories, suggesting the effectiveness of our approach.

Qualitative Analysis

Figure 5 presents the prediction visualizations of multiple detectors under varying traffic densities. In the light traffic scenario, fisheye variants achieve distance-equivariant detection performance to their standard pinhole counterparts for frontally distant objects ($\approx$ 45m range), even with their inherent radial distortion and pixel compression. Under heavy traffic with multi-vehicle occlusion, all detectors exhibit performance degradation on distant objects, with fisheye variants showing a slightly greater decline. This observed discrepancy may be due to the amplified impact of pixel compression under dense occlusion. Notably, fisheye variants maintain near-field detection accuracy equivalent to pinhole models even under these challenging conditions, suggesting their potential as complementary sensors for close-range perception tasks.

Conclusion

We present Fisheye3DOD, a benchmark dataset featuring synchronized multi-fisheye images and 3D annotations, to enable systematic study of fisheye-based 3D detection. Based on this dataset, we develop FisheyeBEVDet and FisheyePETR, two end-to-end multi-view detectors tailored for fisheye imagery. Our best model outperforms rectification baselines by up to 6.2 FDS. While a performance gap remains compared to pinhole systems, fisheye-based detection proves highly suitable for compact, low-speed robotic platforms with strict space constraints.

Acknowledgments

This research was supported by the Guangzhou Basic and Applied Basic Research Foundation under Grant SL2024A04J0183, the Guangxi Key Research and Development Project under Grant 2024AB08049, the National Natural Science Foundation of China under Grant 92470202, the Fund of National Key Laboratory of Multispectral Information Intelligent Processing Technology (No. 202410487201), and the Major Key Project of Pengcheng Laboratory (PCL2025A02).

References

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Cleveland, W. S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368): 829–836.

Deng, L.; Yang, M.; Li, H.; Li, T.; Hu, B.; and Wang, C. 2019. Restricted deformable convolution-based road scene semantic segmentation using surround view cameras. *IEEE Transactions on Intelligent Transportation Systems*, 21(10): 4350–4362.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.

Ge, C.; Chen, J.; Xie, E.; Wang, Z.; Hong, L.; Lu, H.; Li, Z.; and Luo, P. 2023. Metabev: Solving sensor failures for 3d detection and map segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8721–8731.

Gochoo, M.; Otgonbold, M.-E.; Ganbold, E.; Hsieh, J.-W.; Chang, M.-C.; Chen, P.-Y.; Dorj, B.; Al Jassmi, H.; Batnasan, G.; Alnajjar, F.; et al. 2023. Fisheye8k: A benchmark and dataset for fisheye camera object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5305–5313.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hosseinlou, M. H.; Ahadi, H.; and Hematian, V. 2012. A study of the minimum safe stopping distance between vehicles in terms of braking systems, weather and pavement conditions. *Indian Journal of Science and Technology*, 5(10): 3421–3427.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17853–17862.

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

Huang, J.; Huang, G.; Zhu, Z.; Ye, Y.; and Du, D. 2021. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

Huang, Y.; Zheng, W.; Zhang, Y.; Zhou, J.; and Lu, J. 2023. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9223–9232.

Hughes, C.; Glavin, M.; Jones, E.; and Denny, P. 2009. Wide-angle camera technology for automotive applications: a review. *IET Intelligent Transport Systems*, 3(1): 19–31.

Kannala, J.; and Brandt, S. S. 2006. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8): 1335–1340.

Li, M.; Jin, X.; Hu, X.; Dai, J.; Du, S.; and Li, Y. 2022a. MODE: Multi-view omnidirectional depth estimation with 360° cameras. In *European Conference on Computer Vision*, 197–213. Springer.

Li, Y.; Bao, H.; Ge, Z.; Yang, J.; Sun, J.; and Li, Z. 2023a. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1486–1494.

Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; and Li, Z. 2023b. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1477–1485.

Li, Z.; Lan, S.; Alvarez, J. M.; and Wu, Z. 2024. BEVNeXt: Reviving Dense BEV Frameworks for 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20113–20123.

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 *European conference on computer vision*, 1–18. Springer.

Li, Z.; Yu, Z.; Wang, W.; Anandkumar, A.; Lu, T.; and Alvarez, J. M. 2023c. Fb-bev: Bev representation from forward-backward view transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6919–6928.

Liu, M.; Salzmann, M.; and He, X. 2014. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 716–723.

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, 531–548. Springer.

Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, A.; Wang, T.; and Zhang, X. 2023. Petrv2: A unified framework for 3d perception

from multi-camera images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3262–3272.

Loshchilov, I.; and Hutter, F. 2016. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Mao, J.; Shi, S.; Wang, X.; and Li, H. 2023. 3D object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131(8): 1909–1963.

Philion, J.; and Fidler, S. 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, 194–210. Springer.

Plaut, E.; Ben Yaacov, E.; and El Shlomo, B. 2021. 3D object detection from a single fisheye image without a single fisheye training image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3659–3667.

Playout, C.; Ahmad, O.; Lecue, F.; and Cheriet, F. 2021. Adaptable deformable convolutions for semantic segmentation of fisheye images in autonomous driving systems. *arXiv preprint arXiv:2102.10191*.

Samani, E. U.; Tao, F.; Dasari, H. R.; Ding, S.; and Banerjee, A. G. 2023. F2BEV: Bird's Eye View Generation from Surround-View Fisheye Camera Images for Automated Driving. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9367–9374. IEEE.

Scaramuzza, D.; Martinelli, A.; and Siegwart, R. 2006. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, 45–45. IEEE.

Sekkat, A. R.; Dupuis, Y.; Kumar, V. R.; Rashed, H.; Yogamani, S.; Vasseur, P.; and Honeine, P. 2022. SynWoodScape: Synthetic surround-view fisheye camera dataset for autonomous driving. *IEEE Robotics and Automation Letters*, 7(3): 8502–8509.

Sekkat, A. R.; Dupuis, Y.; Vasseur, P.; and Honeine, P. 2020. The omniscape dataset. In *2020 IEEE International conference on robotics and automation (ICRA)*, 1603–1608. IEEE.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; 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*, 2446–2454.

Sunstein, C. R. 2019. Rear visibility and some unresolved problems for economic analysis (with notes on experience goods). *Journal of Benefit-Cost Analysis*, 10(3): 317–350.

Tatarek, T.; Kronenberger, J.; and Handmann, U. 2017. Functionality, advantages and limits of the tesla autopilot.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, S.; Liu, Y.; Wang, T.; Li, Y.; and Zhang, X. 2023. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3621–3631.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 180–191. PMLR.

Won, C.; Ryu, J.; and Lim, J. 2019. Sweepnet: Wide-baseline omnidirectional depth estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, 6073–6079. IEEE.

Won, C.; Ryu, J.; and Lim, J. 2020. End-to-end learning for omnidirectional stereo matching with uncertainty prior. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 3850–3862.

Xie, S.; Kong, L.; Zhang, W.; Ren, J.; Pan, L.; Chen, K.; and Liu, Z. 2025. Benchmarking and improving bird's eye view perception robustness in autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xie, S.; Wang, D.; and Liu, Y.-H. 2023. OmniVidar: omnidirectional depth estimation from multi-fisheye images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21529–21538.

Yan, J.; Liu, Y.; Sun, J.; Jia, F.; Li, S.; Wang, T.; and Zhang, X. 2023. Cross modal transformer: Towards fast and robust 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 18268–18278.

Yang, C.; Chen, Y.; Tian, H.; Tao, C.; Zhu, X.; Zhang, Z.; Huang, G.; Li, H.; Qiao, Y.; Lu, L.; et al. 2023. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17830–17839.

Ye, Y.; Yang, K.; Xiang, K.; Wang, J.; and Wang, K. 2020. Universal semantic segmentation for fisheye urban driving images. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 648–655. IEEE.

Yogamani, S.; Hughes, C.; Horgan, J.; Sistu, G.; Varley, P.; O'Dea, D.; Uricár, M.; Milz, S.; Simon, M.; Amende, K.; et al. 2019. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9308–9318.

Yogamani, S.; Unger, D.; Narayanan, V.; and Kumar, V. R. 2024a. DaF-BEVSeg: Distortion-aware Fisheye Camera based Bird's Eye View Segmentation with Occlusion Reasoning. *arXiv preprint arXiv:2404.06352*.

Yogamani, S.; Unger, D.; Narayanan, V.; and Kumar, V. R. 2024b. FisheyeBEVSeg: Surround View Fisheye Cameras based Bird's-Eye View Segmentation for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1331–1334.

Zhu, B.; Jiang, Z.; Zhou, X.; Li, Z.; and Yu, G. 2019. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*.