BEVMap: Map-Aware BEV Modeling for 3D Perception

Mincheol Chang[1], Seokha Moon[1], Reza Mahjourian[2], and Jinkyu Kim[1]*
[1]Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea
[2]Waymo Research, Mountain View, CA 94043, USA
*Correspondence: jinkyukim@korea.ac.kr

Abstract

In autonomous driving applications, there is a strong preference for modeling the world in Bird's-Eye View (BEV), as it leads to improved accuracy and performance. BEV features are widely used in perception tasks since they allow fusing information from multiple views in an efficient manner. However, BEV features generated from camera images are prone to be imprecise due to the difficulty of estimating depth in the perspective view. Improper placement of BEV features limits the accuracy of downstream tasks. We introduce a method for incorporating map information to improve perspective depth estimation from 2D camera images and thereby producing geometrically- and semantically-robust BEV features. We show that augmenting the camera images with the BEV map and map-to-camera projections can compensate for the depth uncertainty and enrich camera-only BEV features with road contexts. Experiments on the nuScenes dataset demonstrate that our method outperforms previous approaches using only camera images in segmentation and detection tasks.

1. Introduction

Self-driving systems rely on various sensors to perceive their surroundings and make informed decisions. Cameras are commonly used to capture rich semantic information of a particular view, while LiDAR provides spatial information like depth, orientation, and coordinates. To enhance perception, state-of-the-art methods often integrate different modalities through sensor fusion. Although LiDAR is usually favored due to its accuracy and reliability, adverse weather conditions like heavy rain or fog can negatively affect its performance. Moreover, LiDARs are expensive, which restricts their use in some applications. As a result, recent research has explored creating low-cost LiDAR-free autonomous driving systems using only cameras.

In low-cost camera-based autonomous driving systems, it is a long-standing challenge to generate 3D features from just 2D RGB images. In absence of LiDAR, multi-view (or surround-view) camera setups are a great alternative for obtaining depth cues and semantic information from the environment. While earlier multi-view methods relied on extracting features individually from each camera in a monocular framework [22, 27, 33], recently, BEV-based approaches, which represent surrounding scenes in BEV space have gained popularity [9, 24, 29]. BEV modeling has the advantage that it can aggregate semantic information from multiple views and create a comprehensive representation of the surrounding scene at any specific timestep. State-of-the-art methods in 3D object detection [11, 18, 21] aggregate features from multiple images and create BEV features to perform detection. Although BEV-based methods have greatly improved the performance of camera-only 3D detection, their performance is still limited by inaccurate depth estimation of 2D images as an intermediate step in BEV feature generation.

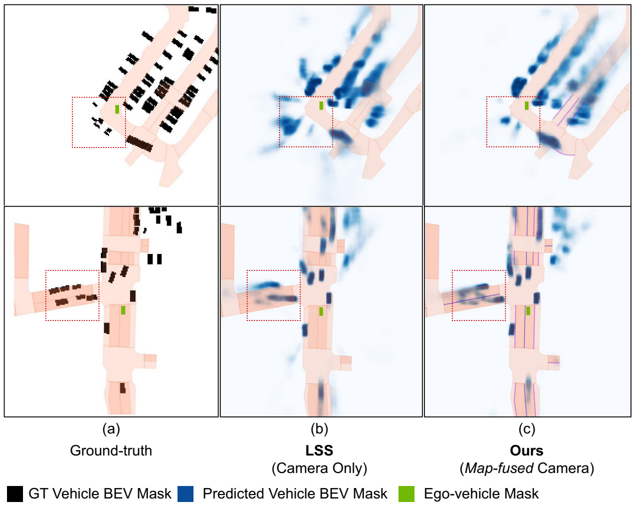
(a)
Ground-truth
(b)
LSS
(Camera Only)
(c)
Ours
(*Map-fused* Camera)

Figure 1. Our model leverages the Bird's-Eye View (BEV) map to improve perspective depth estimation from 2D multi-view camera images, producing geometrically- and semantically-robust BEV features (challenging areas highlighted by dotted boxes). **(a)** Ground-truth **(b)** BEV masks predicted by LSS [24] **(c)** Our predictions.

Generally, BEV-based frameworks using camera images estimate pixel-wise depth and context on images from multiple views [24, 25], and employ a module which splats the image features onto corresponding BEV coordinates based on predicted depth. However, generated BEV features are dependent on semantic features unaware of 3D geometry, leading to incorrect feature projection onto BEV space. Furthermore, as BEVDepth [16] suggests, estimated depth distribution from camera sensors is likely to be far away from the correct depth of 2D points provided by LiDAR. Hence, it is essential to employ auxiliary tools to guide BEV generation models to learn the correct depth distribution of 2D images and enhance produced camera BEV features during training. Previous methods [1, 21] attempted to fuse BEV features extracted from LiDAR and camera to preserve both geometric and semantic information from multiple sensors. Other works [16, 19] proposed to use radar and LiDAR to provide more accurate depth for camera BEV features.

In this work, we introduce a novel method for incorporating map features into BEV modeling for 3D perception tasks. Our method achieves superior performance through two means: 1) we encode and utilize camera-projected map information to improve multi-view depth estimation and 2) we utilize BEV map based spatial normalization to refine the BEV features computed from cameras prior to feeding them to downstream detection and BEV segmentation tasks. To our best knowledge, this is the first map-aware method proposed for camera-based 3D object detection and BEV segmentation. Our method addresses the challenges arising from inaccurate depth estimation and feature smearing in prior methods. We demonstrate its effectiveness through experiments on the nuScenes dataset [3].

2. Related Work

2.1. Multiview Camera-Only 3D Object Detection.

Recent advances in object detection [6, 20, 26, 30] have led to the development of CNN-based 3D object detection models [2, 32, 33, 36] that can generate 3D bounding boxes from monocular images. These models have shown promising results in the field of autonomous driving and robotics. One such approach is the use of the DETR [4] architecture as a base for 3D detection models using surround-view images, which has led to the development of several camera 3D detection models utilizing Transformers, namely DETR3D [34], ORA3D [28]. These models initialize 3D object queries and refine queries by extracting related camera features from multiple views by utilizing spatial cross-attention. BEVFormer [18] further improves detection performance by adopting BEV spatial queries and temporal attention with previous frames.

Another branch of multiview camera detection predicts depth and context in perspective views followed by view-transform operation to BEV features. OFT [27] was the first to adopt a transformation from 2D image features to BEV features. In OFT, image-based features are projected onto the voxel grid and orthographic ground plane to create BEV features. LSS [24] builds upon OFT, utilizing six surround images to extract depth distribution and context features from the image features. These frustum-shaped features are then pooled into BEV features using a splat module. More recent methods like BEVDet [11], BEVDepth [16], BEVStereo [15], and BEVDet4D [10] follow network design of OFT [27], LSS [24] and conduct multiview 3D detection tasks. These methods fuse features from surround-view cameras into the BEV view and improve detection performance by utilizing both spatial and temporal information. Overall, the field of camera 3D object detection is rapidly evolving, and new approaches are being developed to overcome the limitations of existing methods. Along this line of work, we improve depth distribution using map information, which further improves the accuracy and robustness of 3D object detection models.

2.2. Multimodal Fusion.

Recently, many works in 3D object detection have focused on multimodal fusion techniques that combine data from different sensors to improve perception accuracy. (i) Camera-LiDAR: MV3D [5] uses multiple cameras and LiDAR sensors to incorporate 2D and 3D information. AVOD [13] first generates a set of frustums from the LiDAR point cloud, and then extracts features from the RGB image within each frustum. The features are aggregated across all frustums to obtain 3D object detections. TransFusion [1] presents a robust LiDAR-camera fusion approach using transformer-based [31] networks to extract features from both modalities and fuse them for improved object detection. DeepFusion [17] introduces two techniques for effectively aligning the transformed features from the two modalities using the attention-based method. (ii) Camera-Radar: RadarDepth [19] investigates combining monocular images and sparse radar data for depth estimation in 3D object detection.

There also exist approaches to leverage High-Definition (HD) maps with LiDAR sensors. HDNet [35] is a single-stage detector for 3D object detection that operates in BEV and fuses LiDAR information with rasterized maps. The goal of HDNet is to exploit HD maps, which provide strong priors, to boost the performance and robustness of modern object detectors. MapFusion [7] is a framework proposed to integrate HD maps with point cloud data into 3D object detection pipelines to improve performance. Unlike these, we want to explore a fusion between multiview cameras and HD maps. To our best knowledge, ours is the first attempt to use a map to compensate for depth ambiguities from monocular cameras, further improving perception accuracy.

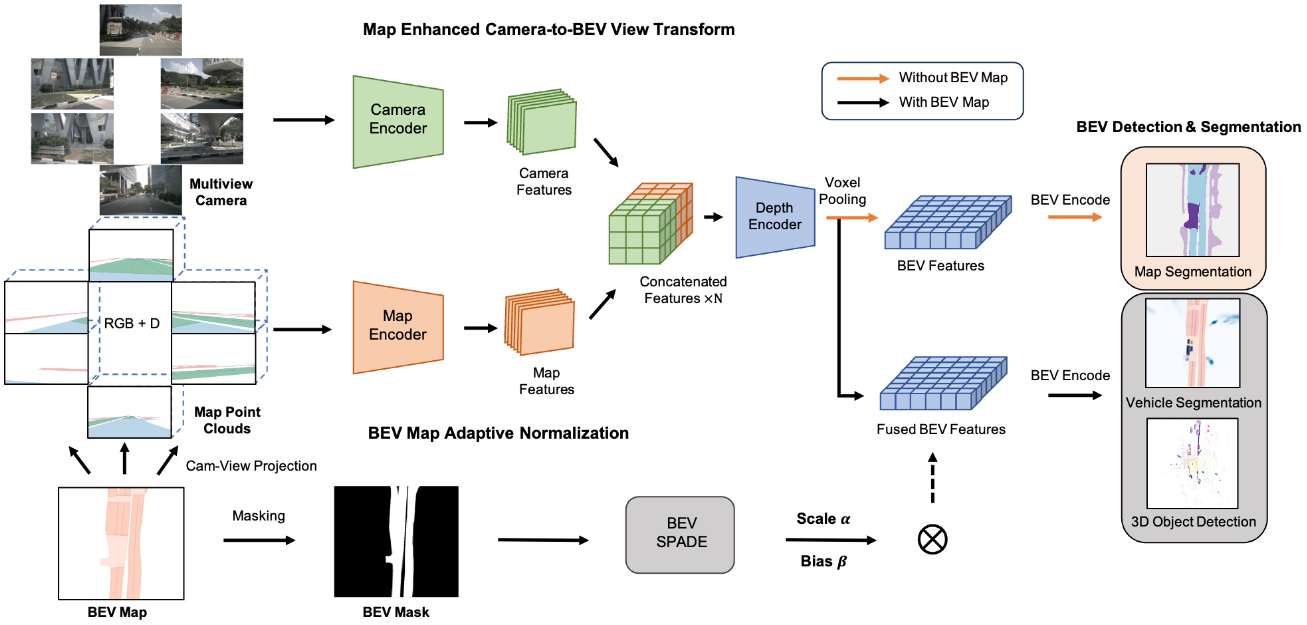

Figure 2. Overview of our BEVMap framework. The BEV map is projected onto multi-view camera images and separately encoded. These encodings are then merged with RGB features obtained from multi-view images and used to predict perspective depth for each view. The predicted depth is used to place perspective features inside a BEV feature map. Separately, BEV map mask is utilized to spatially modulate scale and bias of BEV features obtained from multi-view cameras to generate BEV map-aware features. (Orange arrow denotes task performed without BEV Map mask input and black denotes tasks with BEV Map mask input.)

3. Method

Following existing approaches [11, 16], we start from LSS [24], which introduces the camera-to-BEV transformation approach. In the first two steps of LSS, they encode multi-view features into 3D frustums with estimated depth (i.e. "lift") and then transform them onto a unified plane in BEV (i.e. "splat"). This BEV representation enables encoding multiple camera inputs into a unified space, which can eventually be used for various downstream tasks, such as 3D perception, motion planning/prediction, and BEV segmentation.

Building upon LSS [24], we leverage road information as additional depth cues to improve the quality of estimated depth from multi-view cameras. Similar to existing approaches, our model takes as an input a set of K images I_c of different views from arbitrary cameras, and produces a BEV representation of the scene around the ego vehicle. We assume that camera intrinsics $\mathbb{I}_c \in \mathbb{R}^{3\times3}$ and extrinsics $\mathbb{E}_c \in \mathbb{R}^{3\times4}$ are known for each camera c. Furthermore, we assume availability of a map $\mathbb{M}$ in the top-down coordinate system, but not any other depth sensor, such as LiDAR.

3.1. Map Features in Perspective View

As shown in Fig. 2 (bottom left), we first generate an ego-centered BEV map in the top-down coordinate system. This map contains various map elements (e.g. traffic lights, car parking areas, stop lines, walkways, etc.) represented as polygons. A top-down coordinate system is built such that the pose of ego vehicle at any time t is always at a fixed location in the image. The polygons for the map elements are then rasterized into this coordinate system. Each map element is mapped to a specific RGB color according to its label, yielding a 3-channel image of size H×W containing rasterized polygons corresponding to different map elements.

In addition to rendering the map elements, we also generate a grayscale image with the same size H×W rendering the distance from the ego vehicle to each point on the BEV map. This distance map is then concatenated with the rendered map elements and projected into the perspective view for each camera and shown in Fig. 2 (middle left) and in more detail in Fig. 3. More specifically, a grid of equally-spaced points (x_i, y_i) are sampled from the rendered BEV map and transformed to each camera c using the camera extrinsic and intrinsic matrices $\mathbb{E}_c, \mathbb{I}_c$ to obtain image coordinates (u_i, v_i) for that camera:

$$\begin{aligned}(u_i, v_i) &= \mathbb{I}_c(\mathbb{E}_c(x_i, y_i, z_i)) \\ &\text{where } z_i = -1\end{aligned} \tag{1}$$

We set all z_i to -1 to compensate for the height of the camera sensors. This process yields a 4-channel per-camera

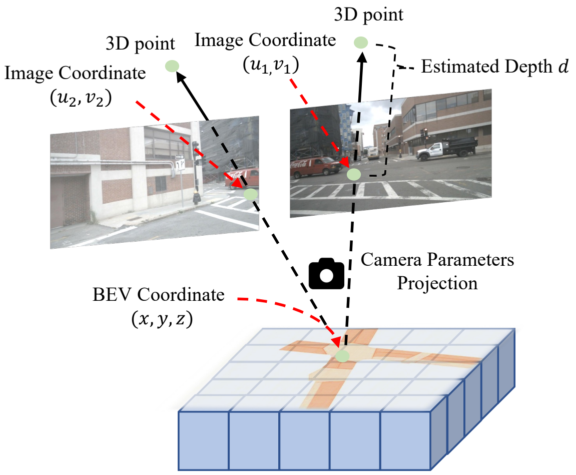

Figure 3. Given an ego-centered BEV map, we use map-to-camera projections based on each views' camera parameters to assign map features to corresponding pixels in the perspective view.

map $M_c \in \mathbb{R}^{H \times W \times 4}$ for each camera c. We dub this RGB+D map as Map PointClouds.

3.2. Perspective Encoding and Fusion

Multi-view camera images I_c and corresponding projected map images M_c are then fed into image encoders f_I and map encoder f_M, respectively. We use the same backbone from LSS [24], which consists of four 3×3 convolution layers with BatchNorm, ReLU activation, and MaxPool layers. Both encoders produce 512-dimensional features of size 22×8, where the decoder will consume these features for downstream tasks such as BEV segmentation and 3D perception tasks.

We then concatenate the features from encoders f_I and f_M to produce the encoding $f_I(I_c) \oplus f_M(M_c)$ joining semantic information from the camera image and and geometric information from the scene map.

3.3. Map-Aware Perspective Depth Prediction

This concatenated feature is then fed into the depth encoder f_d to produce two intermediate features: depth distribution and context features.

$$[F_{\text{depth}}; F_{\text{context}}] = f_D(f_I(I_c) \oplus f_M(M_c)) \tag{2}$$

More specifically, the depth encoder maps each pixel (u, v) of an image $I_c \in \mathbb{R}^{H \times W \times 3}$ to a set of D discrete depth values $F_{\text{depth}} \in \mathbb{R}^{|D|} = \{d_1, d_2, ..., d_D\}$. In addition, the network produces a feature vector capturing the semantic context $F_{\text{context}} \in \mathbb{R}^C$ for each pixel.

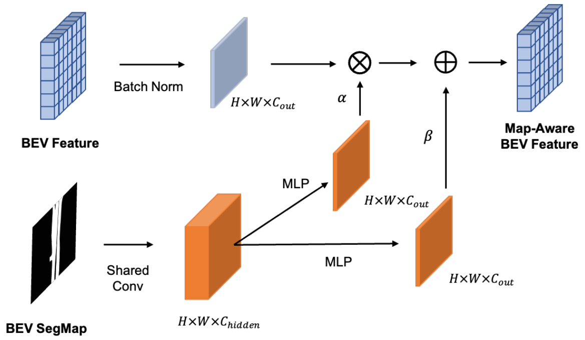
BEV Feature
Batch Norm
$H \times W \times C_{out}$
α
β
Map-Aware
BEV Feature
MLP
$H \times W \times C_{out}$
BEV SegMap
Shared
Conv
MLP
$H \times W \times C_{out}$
$H \times W \times C_{hidden}$

Figure 4. BEV Map-based spatial adaptive normalization. We use spade module [23] to allow network to adaptively modulate BEV feature maps obtained from perspective view to learn statistics of BEV map's road elements.

3.4. Perspective to BEV Projection

For mapping the perspective feature maps to BEV, we follow the lift-splat process in LSS [24], which transforms all pixels in the 2D image plane of all cameras to a unified 3D BEV coordinate system. More specifically, we use the predicted depth distribution F_{depth} for each pixel to create a point cloud by scattering points across all possible depths for that pixel. Each point in this point cloud carries the feature vector F_{context} for that pixel.

Using a voxelization and pooling process, the model projects the point cloud features F_{context} into BEV as

$$BEV_{\text{cam}} = \text{Voxel-Pool}(F_{context}). \tag{3}$$

We filter out points which do not fall within map polygons, or within the model's BEV field of view.

3.5. BEV Map Spatial Normalization

As shown in Fig. 2 (bottom), we also employ a BEV map-based spatially adaptive normalization, named as BEV-SPADE. By using BEV binary map mask transformed from BEV RGB map as an indicator, we aim to manipulate the statistics of objects on roads and objects off roads within the bird's-eye view (BEV) representation for reducing false positive predictions. Following the works of SPADE [23], we adopt a spatially adaptive normalization layer on view-transformed BEV features from camera and perspective view maps as illustrated in Fig. 4. Our BEV-SPADE module is formulated as:

$$BEV_{\text{fused}} = \alpha_{c,y,x}(BEV_{\text{m}}) \frac{BEV_{\text{cam}} - \mu}{\sigma} + \beta_{c,y,x}(BEV_{\text{m}}), \tag{4}$$

where BEV_{m}, BEV_{cam} denote corresponding input BEV segmentation mask of 3D scene and BEV features from perspective view. μ and σ are mini-batch mean and variance. α and β are the channel-wise modulation parameters for BEV_{cam} features. As shown in Fig. 4, BEV features from perspective view go through BatchNorm layer and output normalized activations. We extract features from given BEV segmap with C_{hidden} channels. Depending on the input position (y, x) of the road segmap and order of channel c, MLPs $\alpha_{c,y,x}$ and $\beta_{c,y,x}$ will transform BEV_{m} features to modulated scale and bias for each pixel of BEV feature maps from perspective view.

4. Experiments

4.1. Implementation and Evaluation Details

Implementation Details. BEVMap framework is evaluated in both BEV segmentation and 3D object detection tasks and baseline methods are LSS [24], BEVDet [11], BEVDepth [16]. To demonstrate our framework's effectiveness, the same experiment settings are used in both baseline and our BEVMap methods. For BEV segmentation task, both [24] and BEVMap are trained for 300k steps, around 43 epochs with batch size of 4. The input resolution of both camera images and projected map is 128×352, which is same as camera resolution of LSS network. Adam [12] optimizer with learning rate 1e-3, weight decay 1e-7 is used to optimize binary cross entropy. Generated BEV grid resolution is set as 200×200, and we sample 100×100 map point clouds on BEV. For 3D detection task, the input resolution of models is set as 256×704, and BEV grid resolution is 128×128. We sample 128×128 map point clouds on BEV grid, which are projected to each perspective view as model inputs with projected maps. We train all detection models with 24 epochs, with learning rate of 2e-4 and a total batch size of 56 on 7 NVIDIA GeForce 3090 GPUs. To fully utilize depth featured from map, we exclude lidar-supervised depth loss for BEVDepth baseline and ours while training.

Evaluation Metrics. To evaluate the effectiveness of our proposed model, we perform object and map segmentation tasks in terms of a widely-used Intersection-over-Union (IoU) score, which measures Binary Cross Entropy between segmentation prediction and ground-truth binary mask. Note that a range of 100m × 100m is set for this segmentation task where ego-vehicle is centered (effective forward sensing range of ego-vehicle is 50m).

Further, we also follow the evaluation protocol of nuScenes [3] for the 3D object detection task. We use the following 5 TP metrics: Average Translation Error(ATE), Average Scale Error(ASE), Average Orientation Error(AOE), Average Velocity Error(AVE), and Average Attribute Error(AAE). Note that all TP metrics are also calculated using a 2m center distance threshold during matching. We also measure mean average precision (mAP) by taking the mean value of AP (average precision) of different object classes based on 2D center distance on ground plane. Lastly, we use the nuScenes Detection Score (NDS) to measure a consolidated scalar metric defined as follows: $\text{NDS} = \frac{1}{10}[5\,\text{mAP} + \sum_{\text{mTP}\in\mathbb{TP}}(1 - \min(1, \text{mTP}))]$ where $\mathbb{TP}$ is five TP metrics.

Dataset. We use the nuScenes [3] dataset for our evaluation. nuScenes is a large-scale autonomous driving benchmark dataset that provides a full 360-degree field of view captured by six different cameras on a fixed camera rig. This comprises 20-second-long 1,000 video sequences, which are fully annotated with 3D bounding boxes for ten object classes. The dataset covers 28k annotated samples for training, and validation and testing contain 6k scenes each. Also, nuScenes provides the BEV map, annotating commonly-observed map features such as road segments, lanes, crosswalks, walkways, stop lines, and parking lots. For all scenes, we generate an ego-vehicle-centered map in a top-down coordinate system, and over 34k maps are generated. We will make our generated map publicly available.

Data Augmentation. We also use standard data augmentation techniques over perspective-view map inputs consisting of a set of polygons and bird-eye-view map. Following widely-used data augmentation techniques, we apply random scaling, rotation and flipping of input camera image augmentation to PV maps. We apply the same camera augmentation schemes to projected map depth values on the image plane. Plus, we apply random flipping and rotation to BEV map input along with 3D gt augmentation in BEV space. We observe that applying these augmentation techniques generally improves the representation power of BEV features.

4.2. 3D Object Detection Performance

Quantitative Analysis. To evaluate the effectiveness of our proposed method, we add our map-based approach to existing state-of-the-art methods, including BEVDet [11] and BEVDepth [16], which are multi-view 3D object detection models based on view transformation into BEV representations. As shown in Table 1, models with our map-based approach generally outperform baselines (compare with and without our proposed BEVMap) in terms of a consolidated metric NDS and other error metrics on nuScenes validation data, respectively. Notably, BEVMap model implemented on BEVDepth, which is trained without lidar supervision (depth loss), generally outperforms BEVDepth without lidar supervision and achieves comparable performance with BEVDepth baseline model with lidar supervison. This confirms that using a map modality helps improve overall 3D perception performance. In our analysis, this gain is obtained mainly by improvement in depth estimation and utilization of more road contexts in generating bev features, which we will explain later.

Qualitative Analysis. We observe in Fig. 5 that leveraging a map as an input improves the overall 3D perception performance. Fig. 5 shows the visualized results of 3D bounding boxes predicted by BEVDet [11] (bottom) and BEVDet [11] with our proposed method (top). The pre-

Table 1. 3D object detection performance comparison with the state-of-the-art approaches BEVDet [11] and BEVDepth [16]. nuScenes [3] validation set is used. All models trained with CBGS † : without lidar supervision [39], ‡ : with lidar supervision.

Methods	Modality	Resolution	NDS ↑	mATE↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
BEVDet [11]	Camera	256x704	0.377	0.734	0.274	0.573	0.907	0.239
BEVDet [11] + BEVMap (Ours)	*Map-Fused* Camera	256x704	0.395	0.700	0.275	0.559	0.788	0.227
BEVDepth† [16]	Camera	256x704	0.396	0.666	0.277	0.577	0.909	0.250
BEVDepth† [16] + BEVMap (Ours)	*Map-Fused* Camera	256x704	0.408	0.667	0.271	0.559	0.817	0.212
BEVDepth‡ [16]	Camera, LiDAR	256x704	0.406	0.663	0.271	0.558	0.876	0.237

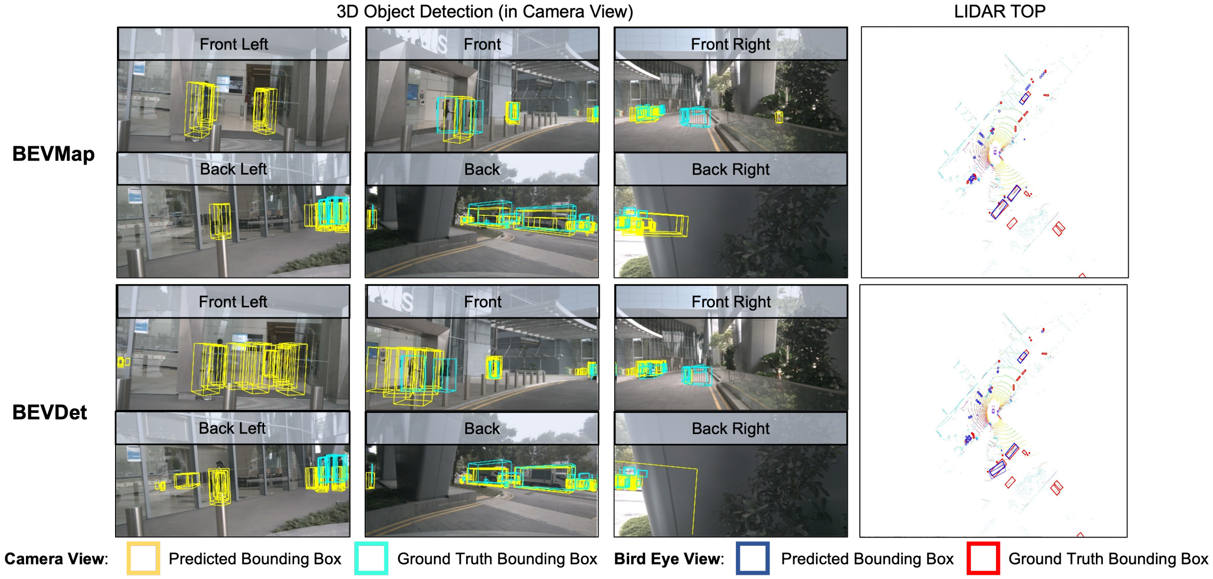

Figure 5. Examples of 3D object bounding box prediction results from ours (top) and our baseline (BEVDet [11]).

dicted bounding boxes are colored yellow with the ground-truth bounding boxes (see cyan boxes) overlaid. We also visualize the predicted and the ground-truth bounding boxes with LiDAR point clouds (see rightmost column). In general, using our model generally improves the overall detection performance by significantly reducing the number of false positives. Our baseline exhibits a relatively large number of false positive detections (see the front left and front view images), while our baseline model with BEVMap is more robust across the entire region. Overall, we observe that our method qualitatively and quantitatively improves detection accuracy.

Further, in Fig. 6, we provide more diverse qualitative examples. For two driving scenarios, we provide (i) the visualized 3D bounding box predictions (yellow) as well as ground-truth (cyan), (ii) visualizations on top of LiDAR point clouds, (iii) BEV object (or vehicle) segmentation results, and (iv) BEV map segmentation results. In (iii), a map is overlaid for better visualization. In (iv), we color-coded differently for each map feature (drivable area with cyan, stop lines with orange, walkways with red, and crosswalks with pink). As we observe in that figure, our model reasonably well predicts 3D object bounding boxes as well as vehicle and map segmentations.

Table 2. Performance comparison for (i) object and (ii) drivable area segmentation tasks over the bird's-eye-view grid. IoU scores are compared with existing approaches, including FISHING [8], OFT [27], and LSS [24].

Model	Modality	IoU scores (%)↑	
		Vehicle	Drivable Area
FISHING [8]	Camera	30.0	-
OFT [27]	Camera	30.1	74.0
LSS [24]	Camera	32.1	75.4
LSS + BEVMap (Ours)	Map-Fused Camera	33.6 (+1.5)	88.8 (+13.4)

Also, as shown in Fig. 7, we observe that a correlation between predicted depth and ground truth depth becomes more improved with our proposed method than that of LSS, i.e. more samples are aligned well with the depth of ground truth.

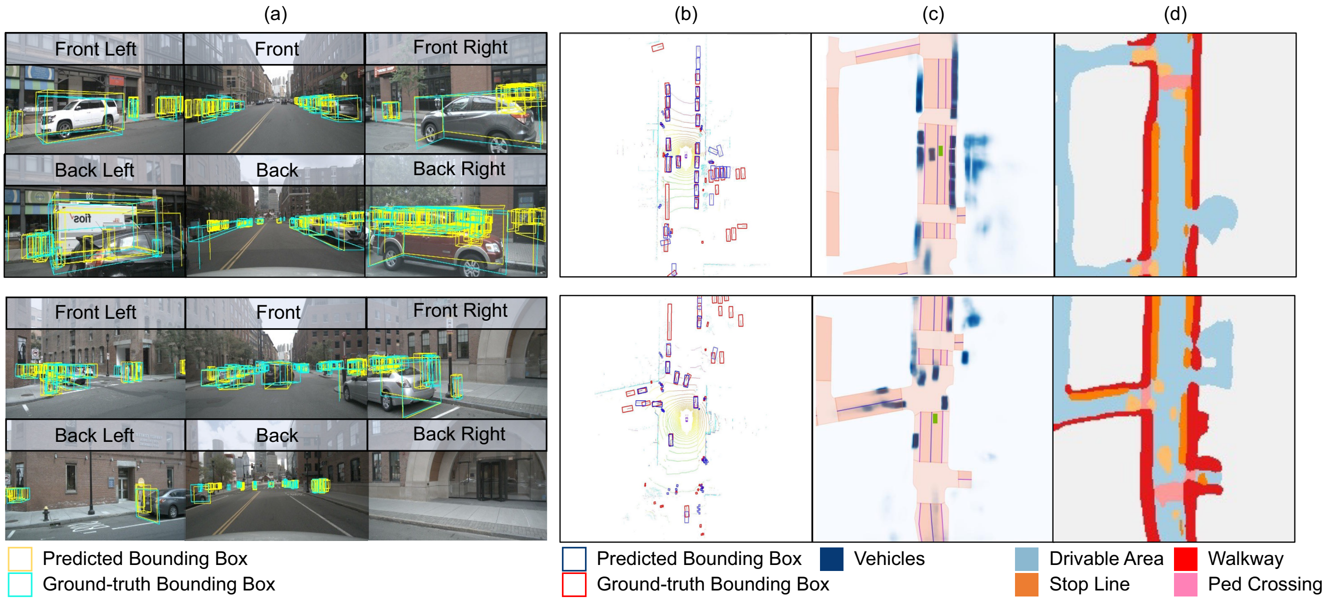

Figure 6. Examples of detected 3D objects visualized on (a) 2D camera images and (b) on LiDAR point clouds. We also provide results of (c) BEV vehicle and (d) map segmentation.

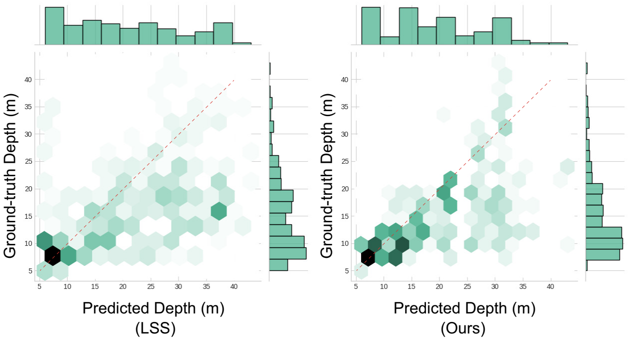

Figure 7. Correlation between LiDAR-based ground truth depth and predicted depth. We compare ours (right) with LSS [24] (left).

4.3. BEV Segmentation Performance

Vehicle and Drivable Area Segmentations. In Table 2, we further demonstrate the ability to learn semantic and geometric BEV representations by evaluating models in the following BEV segmentation tasks: objects and drivable area. We measure IoU scores and compare them with other state-of-the-art approaches: Fishing, OFT, and LSS. Results show that leveraging map-fused camera features largely improves the BEV segmentation tasks, which confirms that map-fused camera features help project pixels in the image plane into the BEV plane, especially for vehicles and drivable areas. BEV Map input is not used for evaluating drivable area segmentation.

Table 3. BEV map segmentation performance comparison. (PVMap Only) IoU scores are reported with existing approaches, including OFT [27], LSS [24], CVT [38], BEVFusion [21], PointPillars [14], CenterPoint [37], and BEVFusion [21]. Note that larger is better. †: Map-fused

	Modality	Drivable	Ped. Cross.	Walkway	Stop Line	Carpark	Mean
PointPillars [14]	L	72.0	43.1	53.1	29.7	27.7	45.1
CenterPoint [37]	L	75.6	48.4	57.5	36.5	31.7	50.0
BEVFusion [21]	C+L	85.5	60.5	67.6	52.0	57.0	64.5
OFT [27]	C	74.0	35.3	45.9	27.5	35.9	43.7
LSS [24]	C	75.4	38.8	46.3	30.3	39.1	46.0
CVT [38]	C	74.3	36.8	39.9	25.8	35.0	42.4
BEVFusion [21]	C	81.7	54.8	58.4	47.4	50.7	58.6
LSS + BEVMap (Ours)	C†	88.8	64.2	74.0	55.9	72.4	71.1

Map Segmentation. We further evaluate the model's ability to learn semantic and geometric information for the BEV map segmentation. In Table 3, we report IoU scores for five map components (i.e. drivable area, pedestrian crosswalk, walkway, stop lines, and car parking area) in the BEV segmentation task. Note that we remove BEV Map input from model design specifically for map segmentation task. We compare ours with other existing approaches, including OFT, LSS, CVT, BEVFusion, PointPillars, CenterPoint, and BEVFusion. The first four methods depend only on camera sensors, while PointPillars and CenterPoint rely on LiDAR sensors. BEVFusion uses a camera and LiDAR sensors together, while ours uses perspective map-fused camera sensors. By incorporating perspective view map encoding with camera features, our approach generally outperforms the other approaches in all categories, confirming that our model learns semantic and geometric information from map inputs. Thus, we believe it can substantially benefit downstream tasks such as 3D perception.

Table 4. Categorical comparison of Object Heading/Velocity Estimation Error on nuScenes *val* set.

Models	Metric ↓	Car	Truck	Bus	Trailer	C.V	Ped.	Mot.	Byc.
BEVDepth	mAOE/mAVE	0.131/0.960	0.140/0.862	0.078/1.726	0.397/0.820	1.030/0.124	1.394/0.887	0.924/1.521	0.985/0.375
BEVDepth + BEVMap	mAOE/mAVE	0.115/0.851	0.122/0.733	0.109/1.818	0.412/0.409	1.026/0.121	1.347/0.882	0.801/1.422	0.987/0.303

Table 5. Ablation Study on 3D Detection Task. We show NDS, mATE, mAOE, mASE metrics, which are related to depth estimation. Lidar supervision option is removed for BEVDepth based models.

Model	NDS ↑	mATE ↓	mAOE ↓	mAVE ↓
BEVDet [11]	0.377	0.734	0.573	0.907
+ Map PointCloud	0.390	0.719	0.575	0.807
+ BEV SPADE	0.395	0.700	0.550	0.788
BEVDepth [16]	0.396	0.666	0.577	0.909
+ Map PointCloud	0.407	0.683	0.551	0.834
+ BEV SPADE	0.408	0.667	0.559	0.817

Table 6. Ablation Study on Vehicle Segmentation Task. Near IoU and Far IoU are measured separately in BEV coordinates.

Model	IoU scores ↑	Near IoU ↑	Far IoU ↑
LSS [24]	32.8	21.1	11.7
+ Perspective-View Map	32.7	21.2	11.5
+ Projected Map Depth	33.1	20.8	12.3
+ BEV SPADE	33.6	21.1	12.5

4.4. Ablation Studies

Extensive ablation study of BEVMap framework is conducted on nuScenes *val* set to demonstrate BEVMap's effectiveness and robustness. We explore two main contributions of BEVMap including perspective view encoding and BEV map spatial normalization in 3D detection and BEV segmentation tasks.

Robust to Different Detectors. We evaluate effectiveness of fusing BEVMap framework on two detectors BEVDet and BEVDepth. Since we attempt to leverage map features on extracting better depth and context features, we choose LSS-based 3D detectors to evaluate BEVMap's effectiveness. Evaluation results on nuScenes *val* set is illustrated in Table 5. BEVMap achieves significant improvement on BEVDet with use of perspective view Map Pointcloud and use of BEV SPADE (approximately 4.77% NDS, 4.63% mATE, 4.01% mAOE and 13.1% mAVE). Huge reduction in error metrics related to distance and direction implies that utilizing map aids in estimating better object depth and heading direction.

In addition, BEVMap brings similar improvements in detection performance in BEVDepth (3.03% NDS). Specifically, both Map PointCloud and BEV SPADE improve mAOE, mAVE, which are related to prediction of correct object heading, by 3.22% mAOE, 11.2% mAVE amount of error reduction. The consistent drop in orientation and velocity error is also illustrated in Table 4, where BEVMap shows better heading/velocity estimation on most of detection categories of nuScenes dataset. Performance gain in mAVE/mAOE demonstrates that BEVMap is capable of leveraging more road contexts in generating BEV features and can predict object orientation based on road features.

Effectiveness of BEVMap in Vehicle Segmentation. We also conduct ablation studies on each component of BEVMap on vehicle segmentation. In Table 6, we separate IoU metric into far and near regions. Near IoU is measured within a 25-meter radius region around the ego vehicle in BEV space, while Far IoU is measured in the rest of region in -51.2m to 51.2m local BEV grid. By using perspective view map and projected distances from ego vehicle (Map Depth) in image plane, IoU in far region improve from 11.5 to 12.3. BEV-SPADE module further boosts IoU in far region. The result implies that fusing map features with camera features provide more accurate depth cues and context in far regions.

5. Conclusion

We propose a novel approach called BEVMap, which augments the camera images with the BEV map to improve perspective depth estimation from 2D multiview camera images. To our best knowledge, this is the first map-aware method proposed for camera-based 3D object detection and BEV segmentation. Our experiments on the large-scale nuScenes dataset demonstrate that our method can produce geometrically- and semantically-robust BEV features and outperforms existing camera-based approaches in the BEV segmentation and detection tasks.

Acknowledgements. We would like to thank Siyeong Lee for his feedback and help with experiment design. This work was supported by Basic Science Research Program (NRF-2021R1A6A1A13044830, 25%), by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2022-0-00043 (50%)), and by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2023-RS-2022-00156295, 25%).

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022. 2

[2] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3d object detection via geometric reasoning on keypoints. *arXiv preprint arXiv:1905.05618*, 2019. 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 5, 6

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 2

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019. 2

[7] Jin Fang, Dingfu Zhou, Xibin Song, and Liangjun Zhang. Mapfusion: A general framework for 3d object detection with hdmaps. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3406–3413. IEEE, 2021. 2

[8] Noureldin Hendy, Cooper Sloan, Feng Tian, Pengfei Duan, Nick Charchut, Yuesong Xie, Chuang Wang, and James Philbin. Fishing net: Future inference of semantic heatmaps in grids. *arXiv preprint arXiv:2006.09917*, 2020. 6

[9] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021. 1

[10] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 2

[11] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2, 3, 5, 6, 8

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[13] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[14] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 7

[15] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022. 2

[16] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022. 2, 3, 5, 6, 8

[17] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17182–17191, 2022. 2

[18] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 1, 2

[19] Juan-Ting Lin, Dengxin Dai, and Luc Van Gool. Depth estimation from monocular images and sparse radar data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10233–10240. IEEE, 2020. 2

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[21] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multitask multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 1, 2, 7

[22] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 1

[23] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. 4

[24] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[25] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for

monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8555–8564, 2021. 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[27] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. 1, 2, 6, 7

[28] Wonseok Roh, Gyusam Chang, Seokha Moon, Giljoo Nam, Chanyoung Kim, Younghyun Kim, Sangpil Kim, and Jinkyu Kim. Ora3d: Overlap region aware multi-view 3d object detection. *arXiv preprint arXiv:2207.00865*, 2022. 2

[29] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5133–5139. IEEE, 2021. 1

[30] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[32] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, pages 1475–1485. PMLR, 2022. 2

[33] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. 1, 2

[34] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 2

[35] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155. PMLR, 2018. 2

[36] Wankou Yang, Ziyu Li, Chao Wang, and Jun Li. A multi-task faster r-cnn method for 3d vehicle detection based on a single image. *Applied Soft Computing*, 95:106533, 2020. 2

[37] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 7

[38] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13760–13769, 2022. 7

[39] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 6