CRN: Camera Radar Net for Accurate, Robust, Efficient 3D Perception

Youngseok Kim¹ Juyeb Shin¹ Sanmin Kim¹ In-Jae Lee¹ Jun Won Choi² Dongsuk Kum¹
¹KAIST ²Hanyang University
{youngseok.kim, juyebshin, sanmin.kim, injaelee, dskum}@kaist.ac.kr, junwchoi@hanyang.ac.kr

Abstract

Autonomous driving requires an accurate and fast 3D perception system that includes 3D object detection, tracking, and segmentation. Although recent low-cost camera-based approaches have shown promising results, they are susceptible to poor illumination or bad weather conditions and have a large localization error. Hence, fusing camera with low-cost radar, which provides precise long-range measurement and operates reliably in all environments, is promising but has not yet been thoroughly investigated. In this paper, we propose Camera Radar Net (CRN), a novel camera-radar fusion framework that generates a semantically rich and spatially accurate bird’s-eye-view (BEV) feature map for various tasks. To overcome the lack of spatial information in an image, we transform perspective view image features to BEV with the help of sparse but accurate radar points. We further aggregate image and radar feature maps in BEV using multi-modal deformable attention designed to tackle the spatial misalignment between inputs. CRN with real-time setting operates at 20 FPS while achieving comparable performance to LiDAR detectors on nuScenes, and even outperforms at a far distance on 100m setting. Moreover, CRN with offline setting yields 62.4% NDS, 57.5% mAP on nuScenes test set and ranks first among all camera and camera-radar 3D object detectors.

1. Introduction

Accurate and robust 3D perception system is crucial for many applications, such as autonomous driving and mobile robot. For efficient 3D perception, obtaining a reliable bird’s eye view (BEV) feature map from sensor inputs is necessary since various downstream tasks can be operated on BEV space (e.g., object detection & tracking [80], BEV segmentation [82], HD map generation [64], trajectory prediction [17], and motion planning [52]). Another important ingredient for deploying 3D perception to the real world is to build a system that relies less on LiDAR disadvantaged from high-cost, high-maintenance, and low-reliability. Apart from the drawbacks of LiDAR, 3D perception system is required to identify semantic information on the road (e.g., traffic lights, road sign) that can be easily leveraged by camera. In addition to the need for rich semantic information, detecting distant objects is essential, and this can be benefited from radar.

Recently, camera-based 3D perception in BEV [19, 52, 57] has drawn great attention. Thanks to rich semantic information in dense image pixels, camera approaches can distinguish objects even at a far distance. Despite the advantage of cameras, localizing the accurate position of objects from a monocular image is naturally a challenging ill-posed problem. Moreover, cameras can be significantly affected by illumination conditions (e.g., glare, low-contrast, or low-lighting) due to the nature of the passive sensor. To address this, we aim to generate a BEV feature map using a camera with the help of a cost-effective range sensor, radar.

Radar has advantages not only in cost but also in high-reliability, long-range perception (up to 200m for typical automotive radar [8]), robustness in various conditions (e.g., snow, fog, or rain), and providing velocity estimation from a single measurement. However, radar also brings its challenges such as sparsity (typically 180× fewer than LiDAR points per single frame in nuScenes [2]), noisy and ambiguous measurements (false negatives by low resolution,
accuracy, or low radar cross-section, and false positives by multi-path or clutters). As a result, previous camera-radar fusion methods using late fusion strategies that fuse detection-level results [7, 13] fail to fully exploit the complementary information, thus having limited performance and operating environment. Despite the huge potential of learning-based fusion, only a few studies [23, 24, 48] explore camera-radar fusion in autonomous driving scenarios.

To put the aforementioned advantages and disadvantages of camera and radar in perspective, camera-radar fusion should be capable of following properties to fully exploit the complementary characteristics of each sensor. First, camera features should be accurately transformed into BEV space in terms of spatial position. Second, the fusion method should be able to handle the spatial misalignment between feature maps when aggregating two modalities. Last but not least, transformation and fusion should be adaptive in order to tackle noisy and ambiguous radar measurements.

To this end, we design a novel two-stage fusion method for BEV feature encoding, Camera Radar Net (CRN). The key idea of the proposed method is to generate semantically rich and spatially accurate BEV feature map by fusing complementary characteristics of camera and radar sensors. In particular, we first transform image features in perspective view into BEV not solely relying on estimated depth but using radar, named radar-assisted view transformation (RVT). Since transformed image features in BEV is not completely accurate, following multi-modal feature aggregation (MFA) layers consecutively encodes the multi-modal feature maps into a unified feature map using an attention mechanism. We conduct extensive experiments on nuScenes and demonstrate that our proposed method can generate a fine-grained BEV feature map to set the new state-of-the-art on various tasks while maintaining high efficiency, as shown in Fig 1.

The main contributions of our works are three-fold:

- **Accuracy.** CRN achieves LiDAR-level performance using camera and radar on 3D object detection, tracking, and BEV segmentation tasks.
- **Robustness.** CRN maintains robust performance even if one of the single sensor inputs is entirely unavailable, which allows the fault-tolerant system.
- **Efficiency.** CRN requires marginal extra cost for significant performance improvement, which enables long-range perception in real-time.

2. Related Work

**Camera-based 3D Perception.** Thanks to well-established 2D object detection methods [70, 84] on perspective view images, early approaches extend 2D detector to 3D detector by additionally estimating the distance to objects [65, 74], then transforming object center. DD3D [50] improves detection performance by pre-training depth estimation task on depth dataset [14]. Although a simple and intuitive approach, the view discrepancy between input feature space (perspective view, PV) and output space (bird’s-eye-view, BEV) restricts the network from extending to other tasks.

Recent advances in camera-based perception exploit view transformation. Geometry-based methods [32, 51, 52, 57] explicitly estimate the depth distribution of each image feature on PV and transform them into BEV. BEVDepth [32] empirically shows that training depth distribution with auxiliary pixel-wise depth supervision improves the performance, which corresponds to the results of DD3D [50]. Learning-based methods [20, 33, 45, 82] implicitly model the mapping function from PV to BEV using multi-layer perceptron (MLP) [29, 58] or attention [33, 60].

Obtaining a BEV feature map allows the framework to be easily extended to various downstream tasks performed on BEV space, such as 3D detection and tracking [32], segmentation [82], and prediction [52]. However, camera-only methods have limited localization accuracy due to the absence of distance information in image and are sensitive to lighting or weather conditions. Moreover, achieving high performance only using a camera requires large image input and backbone, which is slow and not applicable for real-time applications.

**Point-based 3D Perception.** LiDAR is the most common and favorable sensor for autonomous driving, while radar point cloud has not yet been thoroughly investigated. LiDAR-only 3D detectors extract features (e.g., PointNet [55, 56]) given irregular and unordered point sets and predict 3D objects on point- [62] or voxelized- [26] feature. Some approaches further utilize point and voxel features together [61], use range view as additional features [75], or filter background points [67].

Although similar data representation of radar point cloud [2, 47] to LiDAR, radar point-based 3D perception is considerably less investigated. Several works [53, 66, 77] examine the radar points for free space detection, but only a few studies [68, 71] attempt 3D object detection in autonomous driving. Radar point-based detection methods adapt PointPillars [26] with graph neural network [63] or KPConv [69] focusing on extracting better local features. However, mostly due to many clutter points and lack of contextual information on radar, the performance of radar-only methods lags significantly behind compared to LiDAR. Considering the high potential of radar having robust measurements regardless of weather conditions and perception range, fusing radar with a camera is promising to supplement the insufficient semantic information.

**Camera-Point 3D Perception.** Fusing complementary information of camera image and range measurement is a
promising and active research topic. However, the view discrepancy between two sensors is regarded as a bottleneck for multi-modal fusion. A line of approach handles discrepancy by projecting 3D information to a 2D image (e.g., points [6, 73, 78], proposals [1, 23, 25], or prediction results [49]) and gathering information around the projected region. Some camera-radar fusion methods [35, 43] attempt to improve depth estimation by projecting radar points to the image.

On the other hand, another line of work lifts 2D image information into 3D. Early studies in 3D detection [24, 48, 54] detect 2D or 2.5D object proposals and lift them into 3D space to fuse with point data; however, this object-level fusion is difficult to be generalized to other tasks in BEV. Thanks to advances in monocular BEV approaches, recent fusion approaches extract image and point feature maps in unified BEV space and then fuse feature maps by element-wise concatenation [42] or summation [31], assuming multi-modal feature maps are spatially well aligned. After, the fused BEV feature map is used in various perception tasks such as 3D detection [10, 31, 34, 81], BEV segmentation [42, 83], or HD map generation [9, 29]. However, despite the unique characteristics of a camera (e.g., inaccurate BEV transformation) and radar (e.g., sparsity and ambiguity), previous camera-radar fusion less considers them. Our proposed CRN focuses on fusing multi-modal feature maps considering the characteristics of each sensor thoroughly to have the best of both worlds.

3. Camera Radar Net

In this paper, we propose a camera radar fusion framework to produce a unified BEV representation given multi-view images and radar points, as illustrated in Fig 2. In Sec. 3.2, we introduce a method to transform image features with radar, then a multi-modal feature aggregation method in Sec. 3.3. Finally, generated BEV feature map is used for downstream tasks in Sec. 3.4.

3.1. Preliminaries

Monocular 3D Approaches. The crux of monocular 3D perception is how to construct accurate 3D (or BEV) information from 2D features, which can be categorized into two groups. Geometry-based approaches [32, 52, 57] predict depth $D$ as an explicit intermediate representation and transform features $F$ in perspective view $(u, v)$ into frustum view $(d, u, v)$ then 3D $(x, y, z)$ by:

$$F_{3D}(x, y, z) = M(F_{2D}(u, v) \otimes D(u, v)),$$

where $M$ denotes view transformation module (e.g., Voxel Pooling [32, 42]) and $\otimes$ denotes outer product. Meanwhile, learning-based approaches [33, 82] implicitly model 3D to 2D projection utilizing mapping networks as:

$$F_{3D}(x, y, z) = f(P_{xyz}, F_{2D}(u, v)),$$

where $f$ denotes mapping function between perspective view and BEV (e.g., multi-layer perceptron (MLP) [58] or cross-attention [33]), and $P_{xyz}$ is voxels in 3D space. Although the approaches are different, the key is to obtain spatially accurate 3D features $F_{3D}(x, y, z)$ through implicit or explicit transformation. We aim to explicitly improve the transformation process using radar measurement.

Radar Characteristics. Radar can have various representations (e.g., 2-D FFT [39], 3D Tensor [22, 46], point cloud [2, 47]). Radar point cloud has a similar representation to LiDAR, but their characteristics are different in terms of resolution and accuracy [8]. Moreover, due to the nature of the operating mechanism of radar [21, 28] and its millimeter scale wavelength, radar measurements are noisy, ambiguous, and do not provide elevation. Therefore, radar measurements are often not returned when objects exist or returned when objects do not exist; hence, naively adopting LiDAR methods to radar shows very limited performance on complex scenarios, as in Tables 7 and 8 (Center-Point [80] with radar input). We exploit radar in an adaptive manner to handle its sparsity and ambiguity.

![Figure 2. The overall architecture of the proposed Camera Radar Net. Given multi-view images and radar points, modality-specific backbones extract features in each view. First, image context features in perspective view are transformed into a bird’s-eye-view with the help of radar measurements by Radar-assisted View Transformation (RVT). After, Multi-modal Feature Aggregation (MFA) adaptively aggregates image and radar feature maps to generate semantically rich and spatially accurate bird’s-eye-view representation.](image-url)
3.2. Radar-assisted View Transformation (RVT)

Image Feature Encoding and Depth Distribution. Given a set of $N$ surrounding images, we use an image backbone (e.g., ResNet [16], ConvNeXt [41]) with a feature pyramid network (FPN) [36] and obtain $16 \times$ downsampled feature map $F_I$ for each image view. Then, additional convolutional layers further extract image context features $C_{FV} \in \mathbb{R}^{N \times C \times H \times W}$ and depth distribution of each pixel $D_I \in \mathbb{R}^{N \times D \times H \times W}$ in perspective view, following LSS [52]:

\begin{equation}
C_{FV} = \text{Conv}(F_I), \quad D_I(u, v) = \text{Softmax}(\text{Conv}(F_I)(u, v)),
\end{equation}

where $(u, v)$ indicates coordinate in the image plane, and $D$ is the number of depth bins.

Radar Feature Encoding and Radar Occupancy. Unlike previous methods [32, 52, 57] that directly “lift” image features into BEV using estimated depth distribution as Eq. 1, we exploit noisy yet accurate radar measurements for view transformation. Radar points are first projected onto each $N$ camera view to find corresponding image pixels while preserving its depth, then voxelized [26] into camera frustum view voxels $V_{FV}^R(d, u, v)$. Note that $u, v$ are pixel units in the image width and height directions, while $d$ is a metric unit in a depth direction. We set $v = 1$ to use pillar-style since radars do not provide reliable elevation measurements. The non-empty radar pillars are encoded into features $F_R \in \mathbb{R}^{N \times C \times D \times W}$ with PointNet [56] and sparse convolution [79]. Similar to Eq. 3, we extract radar context feature $C_{FV}^R \in \mathbb{R}^{N \times C \times D \times W}$ and radar occupancy $O_R \in \mathbb{R}^{N \times 1 \times D \times W}$ in frustum view. Here, convolution is applied to top-view $(d, u)$ coordinate instead of $(u, v)$:

\begin{equation}
C_{FV}^R = \text{Conv}(F_R), \quad O_R(d, u) = \sigma(\text{Conv}(F_R)(d, u)).
\end{equation}

Here, a sigmoid is used instead of softmax since radar occupancy is not necessarily one-hot encoded as a depth distribution.

Frustum View Transformation. Given depth distribution $D_I$ and radar occupancy $O_R$, the image context feature map $C_{FV}^I$ is transformed into a camera frustum view $C_{FV}^I \in \mathbb{R}^{N \times C \times D \times H \times W}$ as:

\begin{equation}
C_{FV}^I = \text{Conv}[C_{FV}^I \otimes D_I; C_{FV}^I \otimes O_R],
\end{equation}

where $[\cdot \; \cdot]$ denotes the concatenation operating along the channel dimension and $\otimes$ is the outer product. Due to the absence of height dimension in radar and for saving memory, we collapse the image context feature by summation along the height axis, as illustrated in Fig. 3.

Bird’s-Eye-View Transformation. Finally, camera and radar context feature maps in $N$ camera frustum views $F_{FV}^I = \{C_{FV}^I, C_{FV}^R \in \mathbb{R}^{N \times C \times D \times H \times W}\}$ are transformed into a single BEV space $\mathbb{R}^{C \times 1 \times X \times Y}$ by view transformation module $M$:

\begin{equation}
F_{BEV} = M((F_{FV}^I)_{i=1}^N).
\end{equation}

Specifically, we adopt CUDA-enabled Voxel Pooling [30] implementation and modify it to aggregate features within each BEV grid using average pooling instead of summation. It helps the network to predict a more consistent BEV feature map regardless of the distance to the ego vehicle since a closer BEV grid is associated with a more frustum grid due to the perspective projection.

3.3. Multi-modal Feature Aggregation (MFA)

Motivation. Combining complementary multi-modal information while avoiding the drawbacks of each is especially crucial in camera radar fusion, as claimed in Sec. 3.1. Image feature has rich semantic cues, but their spatial position is inherently inaccurate; on the other hand, radar feature is spatially accurate, but contextual information is insufficient and noisy. Naive approaches are channel-wise concatenation [42] or summation [31], but these cannot handle neither spatial misalignment nor ambiguity between two modalities, thus less effective, as can be seen in Table 6. To have the best of both worlds, the key motivation of our fusion is to leverage multi-modal features in an adaptive manner, using an attention mechanism [72].

Multi-modal Deformable Cross Attention (MDCA). Cross attention [72] is inherently suitable for multi-modal fusion. However, the computation cost is quadratic to input sequence length $O(N^2)$, where $N = XY$ and $X, Y$ denote the height and width of the BEV feature map. If we assume perception range $R = X/2 = Y/2$, computation complexity becomes biquadratic $O(16R^4)$ to perception range, which is not scalable for a long-range perception; Thus we develop the fusion method based on deformable attention [86], which is of linear complexity with the input size $O(2N + NK)$, where $K$ is the total number of the sampled key ($K \ll N = XY$).
Given flattened BEV context feature maps \( x_m = \{ C_{BEV}^C, C_{BEV}^R \in \mathbb{R}^{C \times XY} \} \), we first project \( x_m \) into \( C \) dimensional query feature after concatenation as \( z_q = W_z[\text{LN}(C_i); \text{LN}(C_p)], \) where \( W_z \in \mathbb{R}^{C \times 2C} \) is a linear projection and LN is layer norm. After, the feature map is aggregated by multi-modal deformable cross attention as

\[
\text{MDCA}(z_q, p_q, x_m) = \sum_h W_h \left[ \sum_m \sum_k A_{hmqk} \cdot W'_hm(x_m(\phi_m(p_q + \Delta p_{hmqk}))) \right],
\]

where \( h, m, k \) indexes the attention head, modality, and sampling point. To better exploit multi-modal information, we separately apply attention weights \( A_{hmqk} \) and sampling offset \( \Delta p_{hmqk} \) to multi-modal feature maps \( x_m \). By doing so, the feature aggregation module can adaptively benefit from image and radar as shown in Fig. 4. We refer the reader to Appendix for details of the notation.

### Sparse Aggregation

Although MDCA has linear complexity with respect to the size of BEV grids, it still can be a bottleneck when the perception range becomes large. Inspired by [59], we propose a method to further reduce the number of input queries from \( N = XY \) to \( N = N_k \ll XY \) by using features with top-k confidence. Given BEV depth distribution \( D_f \) and radar occupancy \( O_p \), \( N_k \) features \( z_q^{N_k} \in \mathbb{R}^{C \times N_k} \) are selected from input queries \( z_q \in \mathbb{R}^{C \times XY} \) using a probability of \( \max(D_f, O_p) \). The complexity of the proposed sparse aggregation is now independent of perception range, which is more efficient for long-range perception.

### 3.4. Training Objectives and Task Heads

For all tasks, we train the depth distribution network with a depth map obtained by projecting LiDAR points into the image view, following BEVDepth [32].

**3D Detection and Tracking.** For the 3D object detection task, we follow CenterPoint [80] to predict the center heatmap with anchor-free and multi-group head [85]. After, we perform 3D tracking by tracking-by-detection using velocity-based closest distance matching [80]. For training sparse aggregation setting, we filter LiDAR points outside of the 3D bounding box when obtaining a ground truth depth map and replace the softmax to sigmoid in Eq. 3; thereby, only feature grids containing foreground objects can have a high probability.

**BEV Segmentation.** For the BEV segmentation task, we attach a convolutional decoder head to obtain the prediction map following CVT [82]. Given a BEV feature map from Multi-modal Feature Aggregation (MFA) layers, the segmentation head encodes to a latent representation and decodes back to the final output segmentation map, followed by a sigmoid layer. Our BEV segmentation network predicts a semantic occupancy grid of vehicles and drivable area, trained with a focal loss [37].

### 4. Experiments

#### 4.1. Experimental Settings

**Dataset and Metrics.** We conduct experiments on nuScenes [2], which provides radar point clouds at scale. For 3D object detection and tracking, we use official metrics: mAP [11], NDS [2], and AMOTA [76], and we follow the settings proposed by LSS [52] for BEV segmentation. We refer the reader to nuScenes [2] and LSS [52] for details of metrics.

**Implementation Details.** For the camera stream, we adopt BEVDepth [32] as a baseline with several modifications. We reduce the number of depth estimation layers and eliminate the depth refinement module, which increases the inference speed without a significant performance drop. For radar, we accumulate six previous radar sweeps and use normalized RCS and Doppler speed as features following GRIF Net [23]. Unless otherwise specified, we follow standard practices [32] for implementation and training details. We accumulate previous three BEV feature maps with an interval of 1 second, similar to BEVFormer [33].

Our models are trained for 24 epochs with AdamW [44] optimizer in an end-to-end manner, unless otherwise specified. In addition to image and BEV data augmentation [32], we randomly drop sweeps and points for radar [27]. Inference time is measured on an Intel Core i9 CPU and RTX 3090 GPU with a single batch and FP16 precision. The full experimental settings are provided in Appendix.
4.2. Main Results

3D Object Detection. For a fair comparison with previous state-of-the-art 3D detection methods, we train our model only on 3D detection task and report val and test set results in Tables 1 and 2. Under various input image sizes and backbone settings, our CRN ranks first place among all camera-only and camera-radar methods with much faster FPS (Sec. 4.4 for inference time analysis). We emphasize that CRN with a small image input and backbone (256 × 704 and R18) already outperforms competitors with a large image input and backbone (BEVFormer [33] and BEVDepth [32]) in terms of mAP while running an order of magnitude faster, showing the effectiveness of using radar over camera-only methods. CRN also outperforms the LiDAR method CenterPoint-P [80], demonstrating the potential of cost-effective camera and radar to replace LiDAR for autonomous driving. Qualitative results are provided in Fig. 5 and Appendix.

3D Object Tracking and BEV Segmentation. We further demonstrate the generalization performance of CRN on 3D object tracking and BEV segmentation tasks. As shown in Table 3, our tracking result outperforms all published camera-only methods on nuScenes test set. Also, ours not only significantly improves AMOTA but also reduces AMOTP and identity switches.

CRN consistently achieves state-of-the-art performance on BEV segmentation task as shown in Table 4. When compared to previous segmentation methods with a small image input and backbone [82, 5], ours performs significantly better while maintaining a real-time inference speed thanks to our semantically rich and spatially accurate BEV feature map from camera and radar. CRN also achieves higher performance at a much faster FPS than a large image input and backbone [33], demonstrating the effectiveness of fusion.
4.3. Ablation Studies

We conduct ablation studies on val set with a 3D detection task. Unless otherwise specified, models use two frames of $256 \times 704$ image, R50 backbone, and are trained for 24 epochs without CBGS [85]. For thorough comparison, we additionally build three baseline detectors for camera – BEVDepth [32], point – CenterPoint [80], and camera-point – BEVFusion [42]. Details of baselines and additional ablation studies are provided in Appendix.

**View Transformation.** In Table 5, we study how the radar-assisted feature transformation affects performance. View transformation solely relying on estimated depth suffers from inaccurate localization due to the inherent low accuracy of depth distribution. If we naively replace depth distribution to radar (1 if radar point exists inside the voxel, 0 else), performance is severely degraded. This is because image features in perspective view cannot be properly transformed due to the ambiguity and sparsity of radar. With the proposed RVT, the model can benefit from both dense depth and sparse range measurement to significantly improve performance (+8.2% NDS, +11.6% mAP) over depth-only transformation. Moreover, we find consistent performance improvement on LiDAR input, showing the effectiveness of RVT.

**Feature Aggregation.** Table 6 shows the comparison between different feature aggregation methods. BEVFusion [42] fuses multi-modal feature maps in BEV using a single convolutional layer, which is not adaptive and has a small receptive field ($3 \times 3$). Simply adding two additional convolutional layers for fusion, which provides a larger receptive field ($7 \times 7$) and bigger capacity, does not improve the performance much. On the other hand, using only MFA already outperforms deeper BEVFusion with RVT, showing the effectiveness of the proposed multi-modal deformable cross attention. We find that the performance gain of RVT is less significant on MFA than BEVFusion since MFA is already capable of handling spatial misalignment between multi-modal features.

4.4. Analysis

**Scaling Up Perception Range.** In Table 7, we extend the perception range from 51.2 to 102.4 meters and also increase the evaluation range twice correspondingly (see Appendix for details). Although CenterPoint [80] uses 10 LiDAR sweeps, points become extremely sparse as the range increases, and thus performance is significantly degraded at far distances. On the other hand, CRN outperforms LiDAR especially at farther than 30 meters range with a much faster FPS, showing the effectiveness of camera and radar for long-range perception. Moreover, CRN with sparse aggregation further improves the inference speed while preserving comparable performance.

**Robustness.** To systematically analyze the robustness of sensor failure cases, we randomly drop image and radar inputs in Table 8. For fair comparisons, we use single frame input and fix the seed to ensure the same views can be dropped over experiments. We also train both fusion methods with data-level augmentation [6]. CRN not only outperforms BEVFusion when all modalities are available but maintains higher mAP on sensor failure cases. Considering that ours uses radar points at multiple stages (RVT and MFA), each proposed module is trained to be robust to sparse and ambiguous radar points. Especially when radar input is entirely unavailable, BEVFusion suffers from a performance drop over BEVDepth (~15.0%), while CRN still
Figure 5. Qualitative results of long-range model on various weather conditions. Images on the top are the six camera views surrounding the vehicle. Green boxes are ground truths, blue boxes are our prediction results, and black dots are radar points. Perception ranges are set to 200m × 200m, and ground truth maps on the background are used for visualization. Best viewed in color with zoom in.

<table>
<thead>
<tr>
<th>Input</th>
<th>Sunny</th>
<th>Rainy</th>
<th>Day</th>
<th>Night</th>
</tr>
</thead>
<tbody>
<tr>
<td>CenterPoint [80] L</td>
<td>62.9</td>
<td>59.2</td>
<td>62.8</td>
<td>35.4</td>
</tr>
<tr>
<td>RCBEV [83] C+R</td>
<td>36.1</td>
<td>38.5</td>
<td>37.1</td>
<td>13.5</td>
</tr>
<tr>
<td>BEVDepth [32] C</td>
<td>39.0</td>
<td>39.0</td>
<td>39.3</td>
<td>16.8</td>
</tr>
<tr>
<td>CRN C+R</td>
<td>54.8 (+15.8)</td>
<td>57.0 (+18.0)</td>
<td>55.1 (+15.8)</td>
<td>30.4 (+13.6)</td>
</tr>
</tbody>
</table>

Table 9. Analysis of different lighting and weather conditions using mAP metric. CenterPoint [80] results are from BEVFusion [42], and BEVDepth results are reproduced by us.

Weather and Lighting. We analyze the performance under different weather and lightning conditions in Table 9. Note that R101 backbone with 512 × 1408 input is used for BEVDepth and ours for comparable comparisons with LiDAR methods. Sensor noises of LiDAR in rainy conditions or poor illumination of camera at night make object detection challenging for LiDAR-only or camera-only methods. Thanks to fusion with radar, ours shows consistent performance improvement of more than 10 mAP over the camera-only method, demonstrating the effectiveness and robustness of camera and radar sensors in all weather conditions.

Inference Time. We analyze the inference time of each proposed component in Fig. 6. In all analyses, we assume that the BEV feature map of the previous frame \( T - 1 \) can be stored and accessed at the current frame \( T \) since ours does not use temporal information (e.g., temporal stereo methods [30, 51]) when obtaining the BEV feature map. It means that using a multi-frame only increases the latency of the BEV head. Ours requires negligible additional computation for point encoder and fusion modules, but the performance gain over additional latency is substantial (+14.9ms for +12.4 NDS in 256x704 and R50 setting). Moreover, ours with small input can outperform camera-only with larger input in terms of both latency and performance. We expect that inference optimization methods (e.g., TensorRT) can further reduce the latency of large model for long perception range setting to match the real-time.

5. Conclusion

We present CRN, a novel camera-radar fusion method for accurate, robust, and efficient multi-task 3D perception. Our method effectively overcomes the limitation of each modality and fuses multi-modal information to generate contextually rich and spatially accurate BEV scene representation. CRN is also suitable for long-range perception in real-time and achieves state-of-the-art performance on various tasks. We hope that CRN will inspire future research on camera-radar fusion for 3D perception.
Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00236245, Development of Perception/Planning AI SW for Seamless Autonomous Driving in Adverse Weather/Unstructured Environment and 2022R1A2C200494412)

References


[22] Jinhyeong Kim, Youngseok Kim, and Dongsuk Kum. Low-level sensor fusion network for 3d vehicle detection using...


17626