CBILR: Camera Bi-Directional LiDAR-Radar Fusion for Robust Perception in Autonomous Driving

Arthur Nigmatyanov, Gonzalo Ferrer and Dzmitry Tsetserukou
CBILR: Camera Bi-directional LiDAR-Radar Fusion for Robust Perception in Autonomous Driving

Arthur Nigmatzyanov
Skoltech University
Moscow, Russia
Artur.Nigmatzyanov@skoltech.ru

Gonzalo Ferrer
Skoltech University
Moscow, Russia
G.Ferrer@skoltech.ru

Dzmitry Tsetserukou
Skoltech University
Moscow, Russia
D.Tsetserukou@skoltech.ru

Abstract
Safe and reliable autonomous driving hinges on robust perception under challenging environments. Multi-sensor fusion, particularly camera-LiDAR-Radar integration, plays a pivotal role in achieving this goal. Different sensors have specific advantages and disadvantages. Existing pipelines are often constrained by adverse weather conditions, where cameras and LiDAR suffer significant degradation. This paper introduces the Camera Bi-directional LiDAR-Radar (CBILR) fusion pipeline, which leverages the strengths of sensors to enhance LiDAR and Radar point clouds. CBILR innovates with a bi-directional prefusion step between LiDAR and Radar, leading to richer feature representations. First, prefusion combines LiDAR and Radar points to compensate for individual sensor weaknesses. Next, the pipeline fuses the pre-fused features with camera features in the bird’s eye view (BEV) space, resulting in a comprehensive multi-modal representation. Experiments have demonstrated that CBILR outperforms state-of-the-art pipelines, achieving superior robustness in challenging weather scenarios.

Keywords
Fusion, Self-driving, Autonomous vehicle, Camera, LiDAR, Radar, Weather Conditions

ACM Reference Format:

1 Introduction
For self-driving systems, it is crucial to develop a fast and accurate 3D object detector that predicts the bounding boxes and categories of road objects. Nowadays, cameras, LiDARs, and Radars are often used in advanced systems such as drones, robots, and autonomous vehicles. Many authors only use particular sensors to solve perception problems. This can lead to a generalization problem, because there is a high probability that one type of sensor will be more relevant than others for certain real-world scenarios. Each sensor has advantages and disadvantages. From cameras, we can only obtain color and texture information about objects after projective transformation of captured 3D scene into a 2D plane and long stages of post-processing raw images [23], which is the field of color science. For this reason, cameras cannot provide accurate depth information (especially in low light conditions) compared to Radars and LiDARs that operate directly in 3D space [8, 12]. However, researchers continue to develop perception algorithms that rely only on cameras because it is a more cost-effective approach [10, 29].

1.1 Fusion approaches
Sensor fusion is an essential topic in many perception systems. A lot of papers [28, 31] are devoted to LiDAR-camera fusion because LiDARs have higher resolution, are less sparse than Radars and can provide accurate measurements at close range. Since Radar antennas are often installed horizontally, they cannot capture sufficient vertical height information [26]. For voxel representation, a highly sparse point cloud means that some voxels contain too few points for processing.

There are several strategies for sensor fusion. Early fusion directly combines sensor inputs before feeding them into shared feature extractors. Late fusion processes sensor inputs independently and then combines the output results. Mid-level fusion [11] provides an intermediate representation for each sensor before the final fusion step.

1.2 BEV Perception
A unified representation is necessary to make it easier to transfer knowledge and combine features from different modalities [9]. The vast majority of modern perception methods use a bird’s eye view (BEV) representation to describe a 3D scene [22, 33]. BEV is an informal perception standard for autonomous driving scenarios [12]. The BEV coordinate system is a rotation of the camera coordinate system, such that the Z-axis is aligned with the cameras negative Y direction, and is placed a fixed distance below the camera as shown in the Figure 1. Data from different modalities are used to provide complementary knowledge such as precise locations from point clouds and rich context from images. For example, fusion
algorithms translate features from different sensors into the BEV representation and then combine them [13].

Cameras are typically mounted on vehicles parallel to the ground and facing outward. For this reason, images are captured in a Perspective View (PV), which is orthogonal to BEV. Objects of the same shape and size in 3D space can have very different representation in the image plane because of their distance from the camera. The BEV representation does not have scale and occlusion problems compared to PV representation [8]. The transformation from PV to BEV is the inverse perspective map problem, and it can have more than one solution. Before the deep learning era, many works tackled this problem by using a homography transformation matrix because of its computational efficiency. Inverse Perspective Mapping (IPM) has been proposed to address this challenging mapping problem [16, 17]. IPM-based methods assume that all points are on the ground plane, sacrificing height variation. In complex real-world scenarios, 3D objects like vehicles possess height and such transformations can cause noticeable artifacts.

In recent years, data-driven methods have been widely used in complex systems such as self-driving vehicles. Data-driven PV-BEV transformation methods can be divided into three main groups: depth-based, MLP-based, and transformer-based approaches [16]. Depth-based methods estimate the depth distribution of each image pixel along the ray (coming from the camera) that intersects objects in the environment. This allows to elevate the 2D features to 3D, and then obtain the BEV representations from 3D through dimensionality reduction. Depth-based PV-to-BEV methods can be divided into two classes depending on the using representation: point-based and voxel-based methods. Point-based methods are straightforward, they directly utilize depth estimation to convert pixels into point clouds. Examples: Pseudo-LiDAR [24], Pseudo-LiDAR++ [27], AM3D [15], PatchNet [14]. Voxel-based method discretize the 3D space to build a regular structure for feature transformation. The disadvantage of this approach is the loss of detailed local spatial information within each voxel. The advantage is that voxels are more effective at covering large-scale scene structure, they are more efficient for 3D scene understanding.

Another approach is to utilize a variational encoder-decoder or MLP to learn implicit representations of camera calibrations to project PV features to BEV. MLP plays the role of a universal approximator of the mapping function from PV to BEV [16]. MLP-based methods focus primarily on working with a single image. The drawback of MLP-based methods is that the learned weights are fixed and not data dependent:

\[ Y = WX, W \neq X \]

Transformer-based methods employ a top-down strategy constructing BEV queries and searching corresponding features in perspective images through cross-attention mechanism. These methods are more expressive, but hard to train.

### 1.3 BEV representation vs voxel-based

A voxel-based scene representation cannot provide computational efficiency because such representation describes a 3D scene with dense cubic features \( V \in \mathbb{R}^{H \times W \times D \times C} \) where \( H, W, D \) are the spatial resolution of the voxel space and \( C \) is the feature dimension. BEV provides the 3D scene with a 2D feature map \( B \in \mathbb{R}^{H \times W \times C} \) which encodes the top view of the scene. This represents the positional information of the ground plane by accumulating voxel features along the vertical \( z \)-axis Figure 2. The height dimension contains less information than the other two dimensions [5]. It is important to note that some researches do not directly use the BEV representation. In [5] for semantic prediction task due to the lack of \( z \)-axis information authors propose Tri-Perspective View (TPV) representation Figure 2.

### 1.4 Camera-to-BEV View Transform

Transforming from a camera view to a bird’s eye view is complex because the depth associated with each camera feature pixel can be ambiguous. The idea of camera-to-BEV transformation is based on projective geometry (see Figure 3). The process of monocular depth estimation involves generating a unique depth value for each pixel in an image. The state-of-the-art approach involves predicting a categorical distribution of depth for each pixel in the image [13, 20, 21]. This technique is known as feature lifting [20].

---

**Figure 1:** Four main coordinate systems: world, ego - vehicle, camera and bird’s eye view [22].

**Figure 2:** Voxel, BEV and TPV representations. Voxel representation is more informative, but cannot provide computational efficiency [5].

**Figure 3:** The important stage of Camera-to-BEV View Transform is estimation of a categorical depth distribution [20].
In [21], the model utilizes the estimated categorical depth distributions to "lift" an input image in 3D, generating a frustum-shaped point cloud of contextual features. The frustum feature grid is then transformed into a voxel grid using specific camera calibration parameters, and then collapsed into a BEV feature grid. All steps are well-illustrated in the paper [21]. By associating image features with estimated depths, image information can be projected into 3D space using a frustum feature network.

The input to the frustum feature network is an image \( I \in \mathbb{R}^{W_I \times H_I \times 3} \), where \( W_I, H_I \) are the image width and height. The network output is a frustum feature grid \( G \in \mathbb{R}^{W_F \times H_F \times D \times C} \), where \( W_F, H_F \) are the width and height of the image feature representation, \( D \) is the number of discretized depth bins, and \( C \) is the number of feature channels. If we have \( N \) cameras, the full size of the frustum features is \( N \times W_F \times H_F \times D \).

Let’s denote \((u, v, c)\) as a coordinate in image features \( F \) and \((u, v, d_i)\) as a coordinate in categorical depth distributions \( D \), where \((u, v)\) is the location of feature pixel, \( c \) is the channel index, and \( d_i \) is the depth bin index. In order to create a frustum feature grid \( G \), each feature pixel \( F(u, v) \) is weighted by its associated depth bin probabilities in \( D(u, v) \). It adds a new depth axis \( d_i \), as shown in Figure 4. The outer product can be used to weight feature pixels:

\[
G(u, v) = D(u, v) \otimes F(u, v)
\]  

where \( D(u, v) \) is the predicted depth distribution and \( G(u, v) \) is a matrix \( D \times C \). The outer product is calculated for each pixel to generate frustum features \( G \in \mathbb{R}^{W_F \times H_F \times D \times C} \). The next steps are voxel transformation using the camera calibration matrix [21] and collapsing to BEV.

For example, Bevfusion [13] converts camera features into a point cloud, aggregates it with BEV pooling and flattens it along the z-axis. Such algorithms can be related to the Lift-Splat category [20, 21, 32].

### 1.5 Motivation

In [30] authors made a detailed review of how autonomous vehicles perceive the environment under adverse weather conditions. They summarized the strengths and weaknesses of each sensor in the chart 5. As we can see, camera sensors are the most sensitive to environmental conditions. But, not all parts of an image typically contain destructive information. For example, in the Figure 6 certain regions of the image provide crucial details about the objects in the scene.

Recent works [3, 13] have used a mid-level fusion approach to aggregate features from all modalities. Combining the representations of different modalities allows to solve perception problems in adverse weather conditions (see the Table 1).

<table>
<thead>
<tr>
<th>Sensor fusion</th>
<th>Configuration</th>
<th>Target weather</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-LRFusion (2023)</td>
<td>R + L</td>
<td>Fog</td>
</tr>
<tr>
<td>RadarNet (2020)</td>
<td>R + L</td>
<td>Rain</td>
</tr>
<tr>
<td>MVDNet (2021)</td>
<td>R + L</td>
<td>Fog</td>
</tr>
<tr>
<td>Liu (2021)</td>
<td>R + C</td>
<td>Rain, fog, nighttime</td>
</tr>
<tr>
<td>Rawashdeh (2021)</td>
<td>C + L + R</td>
<td>Snow</td>
</tr>
<tr>
<td>SLS-Fusion (2021)</td>
<td>L + C</td>
<td>Fog</td>
</tr>
<tr>
<td>Radecki (2016)</td>
<td>L + R + C</td>
<td>Wet conditions</td>
</tr>
</tbody>
</table>

In last time LiDARs and Radars sensors were significantly improved in terms of spatial resolution, accuracy, velocity measurement and resistance to adverse weather conditions [1].

Because images received from cameras may have artifacts and be overlapped for any reason, using visual transformers may not be as efficient as convolutional neural networks. Transformers take and process every patch of an image, even areas that may not be
Agnostic Feature Sampler (MAFS) creates and aggregate features. \( H \) are updated using self-attention modules and FFN. Features from each sensor feature and fusing them. Object queries are used to encode LiDAR point clouds as multi-scale Bird’s-eye view (BEV) feature maps \( F_{\text{rad}} \) for the \( k \)-th camera. So, after camera backbone there are \( m \) image feature maps for each camera.

A transformer decoder uses queries to predict 3D bounding boxes. The predicted boxes can be repeatedly sent back into the transformer decoder and MAFS to refine the predictions. Modality-Agnostic Feature Sampler (MAFS) creates and aggregate features from each modality based on the 3D reference point (initial position) of each query. The 3D reference point is ground to collect features from multiple sources. The input of detection head is a set of object queries \( Q = \{ q_i \}_{i=1}^{N} \subset \mathbb{R}^C \), and features from all sensors, where \( C \) is the output channel of BEV feature map after processing LiDAR point clouds with VoxelNet. MAFS updates each query by sampling features from each sensor feature and fusing them. Object queries are updated using self-attention modules and FFN.

BEVFusion. BEVFusion [13] is the state-of-the-art fusion pipeline on the nuScenes dataset. It fuses camera and LiDAR sensors in BEV space to perform 3D detection and tracking simultaneously. BEVFusion uses an effective method of transforming camera images into a BEV representation and combining them with LiDAR BEV features using convolutional layers. Like FUTR3D, BEVFusion provides independent camera and LiDAR streams (see Figure 8).

Bi-LRFusion. Radar provides long range detection and velocity hints, while LiDAR is better at capturing the object’s 3D shape [26]. To fully utilize the advantages of combining LiDAR and Radar, the authors enhance the Radar features to make them more powerful before the final fusion.

Fusion uses an effective method of transforming camera images to perform 3D detection and tracking simultaneously. BEV-Fusion [13] is the state-of-the-art fusion pipeline relevant for the specific task. For this reason, authors prefer to use convolutional layers first in neural networks as preprocessing part [3, 12, 13]. Also ViTs require enormous amounts of data and computation to train, and in some cases have longer inference time. For this reason, researchers avoid the unwise use of ViTs [2, 3, 12, 13].
Figure 9: The difference between Bi-LRFusion and standard uni-directional fusion [26].

- the Radar-to-LiDAR (R2L) fusion step: combining LiDAR features with the enhanced Radar features in a unified BEV representation.
- predicting 3D bounding boxes for dynamic objects using the obtained BEV features.

Figure 10: The pseudo height feature formation [26].

Figure 11: The pseudo BEV feature formation [26].

As shown in the Figure 9, the authors propose a bidirectional fusion scheme. L2R Fusion Module consists of two submodules: Query-based Height feature Fusion block and Query-based BEV feature Fusion block, as shown in Figures 10 and 11 respectively. To form pseudo-Radar height features the LiDAR raw points are aggregated, and the LiDAR BEV features are aggregated into pseudo-Radar BEV. Then the pseudo-Radar height and pseudo-Radar BEV are concatenated to the Radar BEV features. The next step is the Radar-to-LiDar (R2L) fusion in a unified BEV.

3 Method

Since both LiDARs and Radars operate in 3D space and they are more reliable than cameras under adverse environmental conditions, we first do their prefusion [26]. The figures 12 and 13 illustrate the concept of our pipeline. We have split the illustration of it into two parts. First of all, we want to make the feature extraction like in [13]. Then, we use a specific transformation for a particular sensor to represent the extracted feature in the BEV. The next steps are bilateral LiDAR-Radar prefusion and image feature concatenation for final fusion.

The LiDAR-to-Radar step enriches the Radar points. Similar to [26], we mix Radar points with LiDAR points before encoding. This eliminates the lack of Radar points per voxel, especially in the height direction. LiDAR points are encoded with SECOND [25], Radar points are encoded with PillarFeatureNet [6]. As a BEVFusion, this pipeline can be used for different tasks such as segmentation and 3D object detection. This article includes a link to GitHub for more information. It is recommended to match the configuration file with the pipeline Figures 12, 13.

4 Experiments

The Nuscenes Dataset is widely used dataset for vision-centric perception with six calibrated cameras covering a 360-degree horizontal FOV, 1 LiDAR and 5 Radars. The camera image resolution is 1600×900. Nuscenes consists of 1000 scenes, each one of them is 20 seconds long. 850 scenes are for training/validation and 150 for testing.
The most commonly used criterion for BEV Detection is average precision (AP) and the mean average precision (mAP) over different classes. For BEV Segmentation, IoU for each class and mIoU over all classes. The Average Precision (AP) metric is extended from 2D to the 3D space:

$$ AP = \int_{0}^{\infty} \max \left\{ p \left( r' \mid r' \geq r \right) \right\} dr $$

where $p(r)$ is the precision-recall curve. The difference between 2D AP and 3D AP is the matching criteria between ground truth and predictions when calculating precision and recall.

Instead of IoU to select TP, NuScenes proposes $A_{\text{center}}$ of a predicted object is matched to a ground truth object if the distance of their center locations on the ground (BEV) plane is below a certain threshold $d$. The $A_{\text{center}}$ is calculated under different distance thresholds: $D = \{0.5, 1, 2, 4\}$ meters. The mAP is computed by averaging the $A_{\text{center}}$ over all matching thresholds and all classes $C$: $mAP = \frac{1}{|D|} \sum_{c \in C} \sum_{d \in D} A_{\text{center}}$. NuScenes Detection Score (NDS) is further proposed to take both $A_{\text{center}}$ and the error of other parameters, i.e. size, heading, velocity, into consideration.

In our experiments we compared BEVFusion [13] and BiFusion [26] with our method (see the Table 2).

Table 2: Results of experiments. "L", "C" and "R" represent LiDAR, Camera, and Radar modalities respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
<th>NDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEVFusion (R + L)</td>
<td>62.3</td>
<td>65.54</td>
</tr>
<tr>
<td>BEVFusion (C + L)</td>
<td>68.57</td>
<td>71.40</td>
</tr>
<tr>
<td>CBILR (C + R + L)</td>
<td>71.09</td>
<td>73.36</td>
</tr>
</tbody>
</table>

Experiments show that it is important to use all modalities in a clever way. Combining different modalities helps to overcome the limitations of individual sensors.

5 Conclusion

This work has demonstrated CBILR, a promising multi-sensor fusion framework that aims to improve perception robustness for autonomous vehicles. It has addressed the critical challenge of limited sensor performance in adverse weather conditions, a significant hurdle on the path to achieving truly autonomous navigation. CBILR aims to overcome the limitations of individual sensors.

Bi-LRFusion module enrich the sparse Radar point cloud with LiDAR original points in two directions (R2L submodule) and then adds enhanced Radar feature representation to Lidar feature representation (L2R submodule). This enriched representation significantly enhances the overall perception accuracy, especially under challenging weather scenarios.

The experiments show that using multiple sensors for fusion increases reliability in challenging weather conditions. Previous works uniformly combine all sensors together. They do not consider the weaknesses of different sensors. By utilizing Bi-LRFusion and promoting a thorough understanding of the environment, CBILR strives to lead the way into a new era of strong and adaptable perception. This effort aims to bring autonomous vehicles closer to the ultimate goal of safe and reliable operation in all conditions.

References