Supplementary Material: ORA3D: Overlap Region Aware Multi-view 3D Object Detection

Wonseok Roh[1]
paulroh@korea.ac.kr
Gyusam Chang[1]
gsjang95@korea.ac.kr
Seokha Moon[3]
shmoon96@korea.ac.kr
Giljoo Nam[2]
namgiljoo@gmail.com
Chanyoung Kim[1]
kochanha@gmail.com
Younghyun Kim[4]
yhkim84@hyundai.com
Sangpil Kim[1*]
spk7@korea.ac.kr
Jinkyu Kim[3*]
jinkyukim@korea.ac.kr

[1] Department of Artificial Intelligence,
Korea University,
Seoul, Republic of Korea

[2] School of Computing,
KAIST,
Daejeon, Republic of Korea

[3] Department of Computer Science and Engineering,
Korea University,
Seoul, Republic of Korea

[4] Autonomous Driving Center,
Hyundai Motor Company R&D Division,
Seoul, Republic of Korea

Overview

We first provide details of obtaining overlap region, which is essential to apply multi-view stereo matching (Section A). Also, we explain implementation details, including training details for stereo disparity estimation (Section B). Next, section C provides additional qualitative results with more diverse scenes. Moreover, we further provide examples in the following project website: `https://kuai-lab.github.io/bmvc2022ora3d`. Our code is provided as a separate file.

A. Obtaining Overlap Region in Multi-view Camera System

In the general architecture of the multi-view camera system, cameras are organized cylindrically on a plane. Cameras are also spaced apart at regular intervals to capture 360-degree surrounding information. Structurally, all adjacent camera pairs have a degree of overlap between the images captured by the two. Although this overlap from adjacent cameras is typically small, it serves as a geometric link between two images. Thus, in order to derive

---

the geometric potential of the overlap region, it is essential to find the accurate overlap region above all else.

Our model finds the precise overlap region using projection matrices of each camera. The overview of this process, which is the basis of our model, is represented in Figure. 1. The whole pipeline starts with a set of six images for $I = \{\mathbf{i}_f, \mathbf{i}_{fr}, \dots, \mathbf{i}_{br}\} \in \mathbb{R}^{H \times W \times 3}$ and combinations of camera intrinsics and extrinsics. 2D image pixels are explicitly positioned according to the scene depth in the 3D domain. Therefore, we adopt the infinity value as the depth to reach the accurate real-world location. First, the process of converting the 2D coordinates $\mathbf{X}_f^{2D} = [u,\ v]$, which is expected to be an overlapping point with the right adjacent camera of the front camera, into 3D coordinates by applying infinite depth d_∞ is described below.

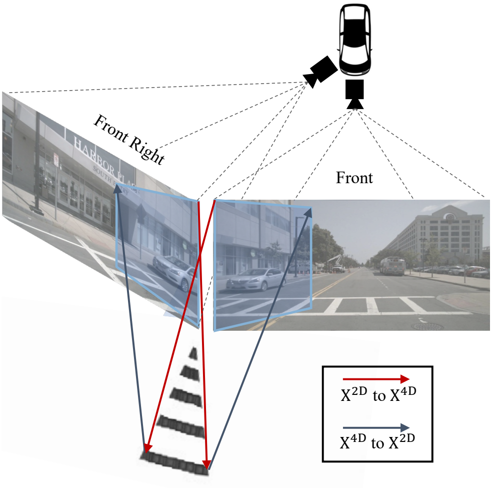

Figure 1: Overview of finding the overlap region between the front and front right camera.

$$\mathbf{X}_f^{3D} = [u\ \times d_\infty,\ v \times\ d_\infty,\ d_\infty] \tag{1}$$

$\mathbf{X}_f^{3D}$ denotes the 3D representation of $\mathbf{X}_f^{2D}$. We convert 4D by concatenating 1 to $\mathbf{X}_f^{3D}$ as below in order to multiply it with a transformation matrix having a 4×4 shape.

$$\mathbf{X}_f^{4D} = \mathbf{X}_f^{3D} \oplus 1 \tag{2}$$

Then multiply the transformation matrices $T = \{\mathbf{T}_f, \mathbf{T}_{fr}, \dots, \mathbf{T}_{br}\} \in \mathbb{R}^{4 \times 4}$ to convert camera coordinates $\mathbf{X}_f^{4D}$ into the egocentric coordinates.

$$\mathbf{X}_f^{\text{ego}} = \mathbf{T}_f\ \cdot\ \mathbf{X}_f^{4D} \tag{3}$$

Here, we need to translate the egocentric coordinates $\mathbf{X}_f^{\text{ego}}$ into 3D points of the camera space neighboring to the right. Specifically, this process goes reverse to the camera-to-egocentric process conducted above, and the target is the front right camera space. To do that, we leverage $\mathbf{T}_{fr}^{-1}$, the inverse matrix of $\mathbf{T}_{fr}$, as shown below.

$$\mathbf{X}^{*4D}_{fr} = \mathbf{T}_{fr}^{-1}\ \cdot\ \mathbf{X}_f^{\text{ego}} \tag{4}$$

Afterwards, we divide $\mathbf{X}^{*4D}_{fr}$ by the pixel depth d_X to project it completely into $\mathbf{X}_{fr}^{4D}$, which corresponds to the homogeneous coordinate system.

$$\mathbf{X}_{fr}^{4D} = \mathbf{X}^{*4D}_{fr}\ /\ d_X \tag{5}$$

We accurately define that $\mathbf{X}_{fr}^{2D} = [u,\ v] \in \mathbb{R}^{H \times W}$ belongs to the overlap domain via the above process. Ultimately, we can reliably determine the boundaries between overlap and non-overlap regions in each image.

B. Implementation Details

Training Details. We implement our model upon DETR3D [3] architecture. Following the default setting used by DETR3D, our model is trained end-to-end using AdamW [2] as an optimizer with the learning rate 3e-5. The whole model is trained for 30 epochs on four NVIDIA GeForce RTX 3090 GPUs, distributing one scene (six images) per GPU. Since the stereo disparity estimation network and the adversarial overlap region discriminator are turned off during inference, the inference time would remain the same as that of DETR3D. We use the publicly available nuScenes as our dataset. The input image size is 1600 × 900. Model evaluation is performed on ten classes: i.e., car, truck, construction vehicle, bus, trailer, barrier, motorcycle, bicycle, pedestrian, and traffic cone.

Stereo Disparity Estimation Details. As shown in Figure 4 in the main paper, our stereo network first obtains 2D features $F \in \mathbb{R}^{B \times C \times H \times W}$ from a general feature extractor for each image. Then, our model utilize the pixel-wise correlation module to learn rich stereo representations and fuse multi-scale hierarchical stereo features to estimate the disparity map densely. The shape of the disparity map output from the module is $[B, C, H/4, W/4]$, which is down-scaled by four compared to the original image shape. Three multi-scale volumes within the network are used to form a cost volume pyramid and are finally output via an inference layer. Note that our disparity estimation module is supervised only for the overlapped areas, but we empirically exhibit that it significantly improves the network's overall 3D object detection performance.

C. Qualitative Results

In this section, we provide additional qualitative examples of ORA3D. The nuScenes [1] dataset is an optimal multi-view dataset for executing tasks such as detection and tracking. Especially, the presence or absence of LiDAR points determines the ground-truth bounding box. For instance, if an object is not visible in the image due to occlusion, but there are LiDAR points for the object, it becomes a ground-truth box. Conversely, if there are no LiDAR points for an object visible in the image, it is not for the ground-truth box.

Figure 2 and Figure 3 show the additional visualized results of 3D bounding boxes predicted by DETR3D [3] (see green boxes) and ORA3D (see magenta boxes). We project the predicted and ground-truth bounding boxes onto images from six different perspectives and Birds'-Eye View. DETR3D predicts 3D coordinates directly from 2D features without using depth cues and achieves promising results in 360-degree. However, DETR3D suffers from ghost boxes (false positive boxes) in the overlap region due to a lack of considering surround-view camera system relationships. In this paper, we enhance the overall performance by developing novel approaches (Stereo Disparity Estimation for Weak Depth Supervision and Adversarial Overlap Region Discriminator) to reasonably use overlapped regions that are small but highly informative.

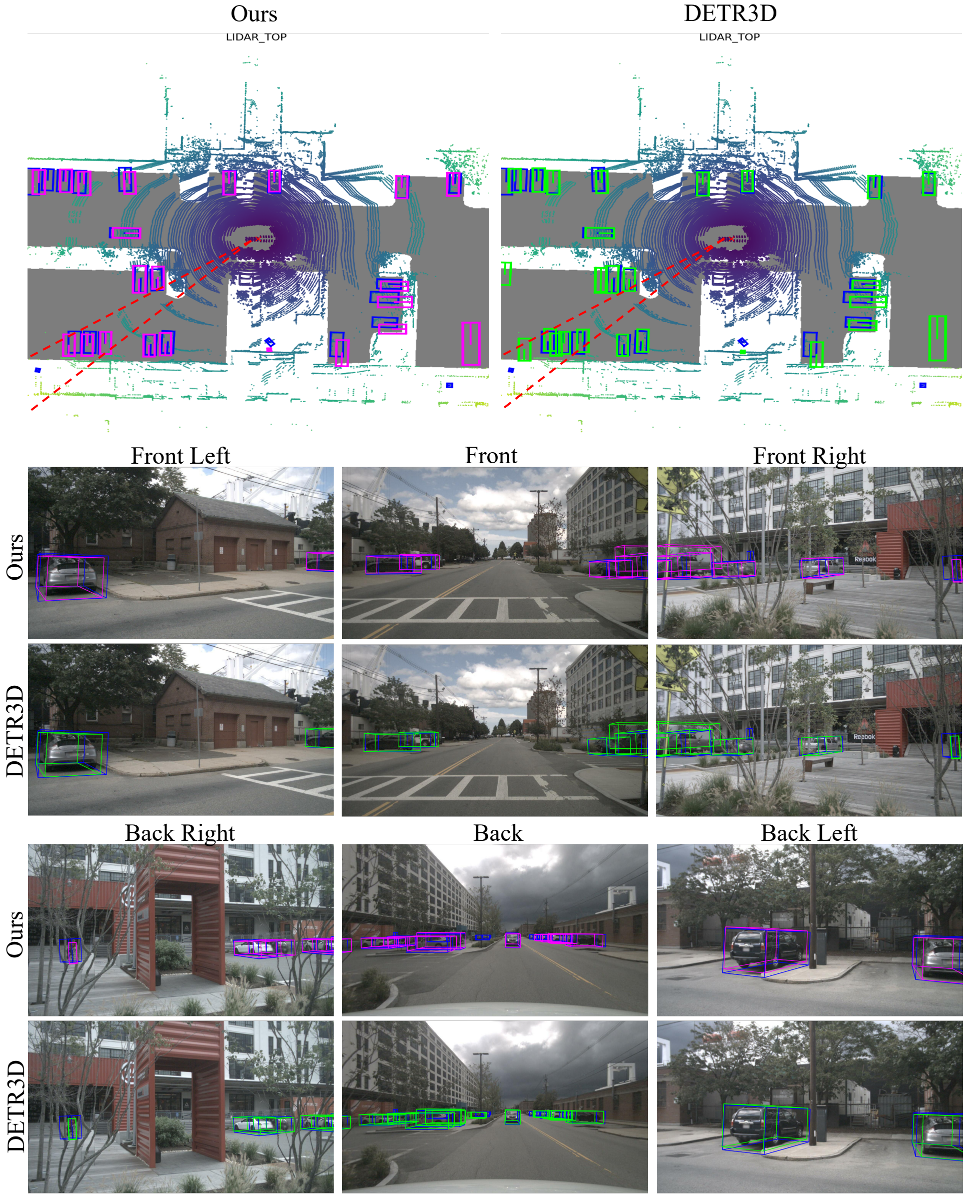

Figure 2: Examples of 3D bounding boxes predictions. The blue, green, and magenta boxes denote ground-truth, DETR3D prediction, and the prediction of ORA3D, respectively. The red dotted lines in the upper BEV images indicates the overlap region between the back right and back cameras of multi-view.

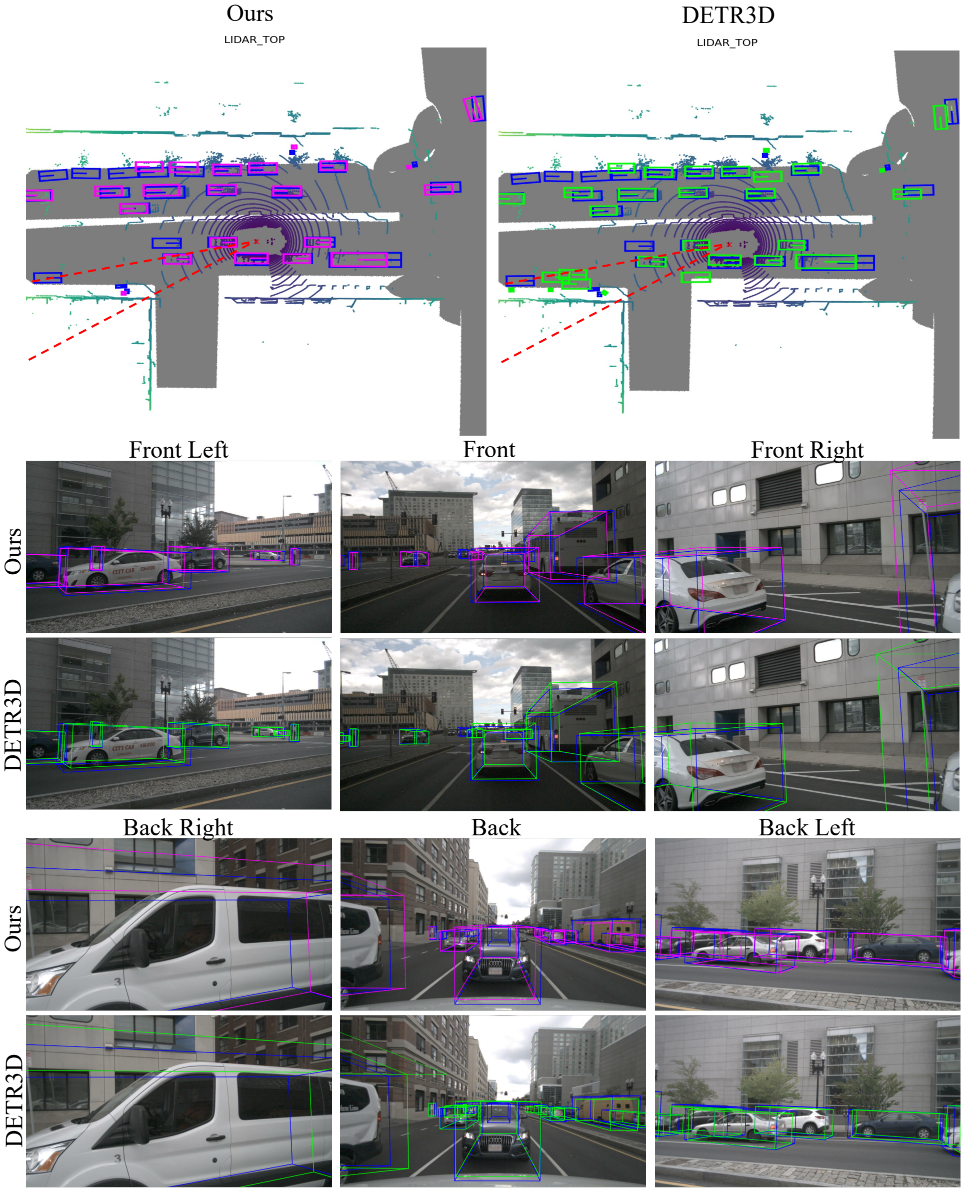

Figure 3: Example of 3D bounding box prediction for a car-crowded scene. Predictions and ground-truth boxes have the identical representation as mentioned in Figure 2.

References

[1] 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] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[3] 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.