Supplementary Material for DORT: Modeling Dynamic Objects in Recurrent for Multi-Camera 3D Object Detection and Tracking

A Evaluation Metrics

Detection Metrics We adopt the official evaluation protocol provided by nuScenes benchmark [1]. The official protocol evaluates 3D detection performance by the metrics of average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), and average attribute error (AAE). Besides, it also measures the mean average precision (mAP) with considering different recall thresholds. Instead of using 3D Intersection over Union (IoU) as the criterion, nuScenes defines the match by 2D center distance d on the ground plane with thresholds $\{0.5, 1, 2, 4\}m$. The above metrics are finally combined into a nuScenes Detection Score (NDS).

Tracking Metrics Regarding the tracking metrics, the nuScenes benchmark mainly measures the average multi-object tracking accuracy (AMOTA), average multi-object tracking precision (AMOTP), and tracking recall. In particular, AMOTA and AMOTP are the averages of multi-object tracking accuracy (MOTA) and multi-object tracking precision (MOTP) under different recall thresholds.

B Implementation Details

In the main paper, we have introduced our overall multi-camera 3D object detection and tracking framework and the details of the proposed components. In this supplemental section, we present the details of the other basic modules.

B.1 Network Architecture

Our framework is built based on BEVDet and BEVDepth, and we follow them to design the basic modules.

2D Feature Extraction. Given N multi-view images $I \in \mathcal{R}^{N\times W\times H\times 3}$in each frame, we use a shared 2D backbone to extract the corresponding features. We adopt the standard ResNet-50 [2] as the backbone and initialize it with ImageNet pre-trained weights. Then we adopt a modified Feature Pyramid Network (FPN) [3] to extract the multiple-level features and the output 2D features are downsampled with the ratio of $\frac{1}{16}$ with channel size 256: $F_{pv} \in \mathcal{R}^{\frac{W}{16}\times\frac{H}{16}\times 256}$.

View Transformation. Our work is the same as BEVDet and BEVDepth that contains a 2D to 3D view transformation module. Specifically, we first leverage a depth prediction head to predict the depth probability for each pixel. Then we lift the 2D features to a 2.5D frustum space via out-product it with the depth probability. The depth probability range is set as $[0m, 60m]$ with grid size $0.5m$. With the 2.5D frustum features, the 3D features for each local volume are obtained via utilizing the camera intrinsic to project the 3D grid back to the frustum and bi-linear sample the corresponding features. As mentioned in the main paper, we aggregate the 3D volume features along the height dimension and obtain the corresponding object-wise BEV features $F_{bev}^{obj} \in \mathcal{R}^{N\times W^{obj}\times H^{obj}\times 256}$, where W^{obj} and H^{obj} are the object features dimension and set as 28 in the main setting.

RefineNet. Given the object-wise features extracted based on the proposal 3D box and motion, RefineNet takes several convolutional neural networks to extract the object-wise features and estimate the bounding box and motion residual. Specifically, we first adopt an average pooling layer to aggregate the 3D features along the height dimension and obtain the BEV features. Then we filter each object-wise BEV features with 6 basic 2D residual blocks, where each residual block consists of two 2D convolution layers and a skip connection module as in ResNet. The channel size of the residual blocks in the first three layers is 256 and decreases to 64 in the last three layers. Then we aggregate the features along the spatial dimension via average pooling and take 4 layers MLP network to estimate the bounding box and motion residuals.

B.2 The Tracking Module

In this section, we provide the details of the tracking module that omit in the paper. Since DORT can estimate tightly coupled object location and motion, object tracking can be easily achieved via nearest center distances association [4, 5, 6]. Hence, our tracking module is mainly adapted from the previous distance-based object tracker [4, 5, 6]. Specifically, the tracking module contains four parts: Pre-processing, Association, Status Update and Life-cycle Management.

Pre-processing. Given the detection results, the pre-processing stages mainly focus on filtering false negative objects. In our work, we first adopt Non-maximum Suppression to remove the duplicated bounding boxes with the threshold of 0.1 in terms of 3D IoU. Then we filter out the bounding boxes that the confidence threshold is lower than 0.25.

Association. This stage associates the detection results in the current frame with tracklets in the past frame. Specifically, we first utilize the estimated object motion (velocity) to warp the detection results back to the past frame and then utilize the L2 distances of object centers to compute the similarity between the detected objects and the tracklets. Then we utilize the linear greedy matching strategy to achieve multi-object matching.

Status Update. This stage updates the status of the tracklets. For the tracklets that do not match with any bounding boxes, we replace it object center location with the corresponding detection results. For the unmatched objects, we utilize the estimated object velocity to update its object center location.

Life-cycle Management. The life-cycle management module controls the "birth" and "depth" of the tracklets (*i.e.* birth, depth). Specifically, for the unmatched bounding boxes, they will be initialized as new tracklets. For the unmatched tracklets, we remove them when they are consecutive unmatched more than 2 times.

C Ablation Studies

In this section, we provide the additional ablation studies that omit in the main paper. We will release the code afterward for providing the details of the methods and reproducing the experimental results.
DORT with Different Proposal Detector. We first show that DORT is agnostic with different proposal detectors (*e.g.* PGD [7], BEVDepth [8]). In Table 1, we display the experimental results of DORT with using PGD and BEVDepth as the proposal detectors. We can observe that the DORT is insensitive to the proposal detector and can consistently improve BEVDepth. We Benefiting from the low computation overhead of BEVDepth in the perspective part and the designed local volume, DORT also can achieve a more lightweight pipeline for dynamic object modeling.

Table 1: Experimental results on the nuScenes validation set. 1 past frame is adopted in the temporal modeling. * denotes the BEV FLOPS from the proposal detector.

Method	mAP	NDS	Flops	
			PV	BEV
BEVDepth	35.1	47.5	120.4	94.5
DORT with PGD	37.9	52.1	238.2	40.2
DORT with BEVDepth	**38.1**	**52.1**	120.4	74.4^*+40.2

Tracking with Semantic Embedding or Geometry Distance. In this work, DORT achieves 3D object tracking via the nearest centerness association. To have a more comprehensive comparison of the tracking pipeline designed, we further provide the comparison of DORT with using semantic embedding to associate objects. Specifically, we follow previous methods [9] and adopt the widelyused quasi-dense similarity learning [10] to learn the tracking embedding. We extract two kinds of embedding features, one is from the perspective-view (PV) and another is from the bird-eye-view (BEV). In Table 2 and 3, we display the tracking results on the nuScenes tracking set. We can observe that DORT with geometry distance association can outperform the embedding-based methods by a large margin. Furthermore, it is also much simpler and more efficient that does not need to maintain an extra object embedding. Besides, the PV embedding is worse than the BEV-based embedding, which may be due to the view change in different cameras.

Table 2: Experimental results on the nuScenes validation set. 1 past frame is adopted in the temporal modeling.

Method	AMOTA↑	AMOTP↓	MOTAR↑
PV-Embedding	36.8	1.412	44.2
BEV-Embedding	40.1	1.356	46.7
DORT (Geometry Distance)	**42.4**	**1.264**	**49.2**

Table 3: 3D object tracking results on the nuScenes validation set. We adopt ResNet-50 as the backbone and set the input resolution as 704×256.

Method	AMOTA↑	AMOTP↓	Recall↑
QD-Track3D [9]	24.2	1.518	39.9
Time3D [11]	21.4	1.360	N/A
TripletTrack [12]	28.5	1.485	N/A
MUTR3D [13]	29.4	1.498	42.7
QTrack [14]	34.7	1.347	46.2
DORT	**42.4**	**1.264**	49.2

D Theoretical Analysis of Ignoring Object Motion

In the main paper, we have shown that when ignoring object motion, the temporal correspondence would derive a biased depth. In this supplementary, we provide the full details of how ignoring object motion introduces a biased depth. We denote the camera intrinsic as K and the ego-motion from frame t_0 to frame t_1 as $T^{ego}_{t_0 \to t_1}$:

$$K = \begin{bmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{bmatrix}, T^{ego}_{t_0 \to t_1} = \begin{bmatrix} 1 & 0 & 0 & x^{ego} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z^{ego} \end{bmatrix}. \tag{1}$$

Here, f is the camera's focal length, and (c_u, c_v) is the camera center coordinates in the image. For simplicity, we assume the ego-motion only contains the translation $(x^{ego}, 0, z^{ego})$ on the horizontal plane. The analysis also can be easily extended to a more complicated case that the motion contains rotation. Given the multiple-view images, temporal-based methods can utilize photometric or featuremetric similarity to find the correspondence of pixel $p_{t_0} = (u_{t_0}, v_{t_0})$ in the past frame t_0 and the pixel $p_{t_1} = (u_{t_1}, v_{t_1})$ in the current frame t_1.

When we ignore the object motion, the depth z_{t_1} of pixel p_{t_1} can be recovered as:

$$\begin{aligned} & T^{ego}_{t_0 \to t_1} \cdot \pi(p_{t_0}, K) = \pi(p_{t_1}, K), \\ & z_{t_1} \frac{u_{t_1} + c_u}{f} - x^{ego} = \frac{u_{t_0} + c_u}{f}(z_{t_1} - z^{ego}), \\ & z_{t_1} = \frac{z^{ego}(u_{t_0} - c_u) - f x^{ego}}{u_{t_0} - u_{t_1}}, \end{aligned} \tag{2}$$

where π denotes the projection from 2D image coordinate to 3D camera coordinate.

But as we showed in the main paper, the moving objects occupy large ratios in the driving scenarios. For example, when the object contains the translation $(x^{obj}, 0, z^{obj})$ in the horizontal plane, the object's motion can be represented as

$$T^{obj}_{i \to j} = \begin{bmatrix} 1 & 0 & 0 & x^{obj} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z^{obj} \end{bmatrix}. \tag{3}$$

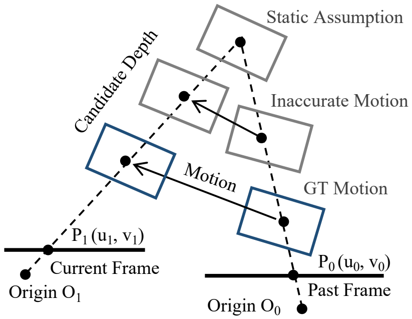

Figure 1: Different object motion can make the same temporal correspondence derive different depth.

With the object motion, the depth z_{t_1} of pixel p_{t_1} is recovered as:

$$\begin{aligned}
&T^{obj}_{t_0\to t_1}T^{ego}_{t_0\to t_1}\cdot\pi(p_{t_0},K)=\pi(p_{t_1},K),\\
&z_{t_1}\frac{u_{t_1}+c_u}{f}-x^{ego}-x^{obj}=\frac{u_{t_0}+c_u}{f}(z_{t_1}-z^{ego}-z^{obj})\\
&\hat{z}_{t_1}=\frac{(z^{ego}+z^{obj})(u_{t_0}-c_u)-f(x^{ego}+x^{obj})}{u_{t_0}-u_{t_1}}.
\end{aligned}\tag{4}$$

From Eq (2) and Eq (4), we can obtain the depth gap for the temporal correspondence with and without considering object motion:

$$\Delta z=\frac{z^{obj}(u_{t_0}-c_u)-fx^{ego}}{u_{t_0}-u_{t_1}}.\tag{5}$$

In Figure 1, we also provide a toy example to illustrate that one temporal correspondence can come from multiple combinations of object depth and motion (*i.e.* inaccurate depth with zero motion and accurate depth and GT motion). This means that if we inaccurately assume that objects are static across frames, the temporal correspondence would derive a misleading depth.

D.1 Ill-posed Problem of Simultaneously Estimating 3D Location and Motion

Although object motion plays a critical role in temporal correspondence, however, it is non-trivial to estimate it from the monocular images. As shown in Figure 1, the one correspondence can come from infinite combinations of location and motion (the location can be the point in the ray $\overrightarrow{O_{t_0}P_{t_0}}$ and $\overrightarrow{O_{t_1}P_{t_1}}$, and the motion can be the line that connects the points.) Hence, it is an ill-posed problem that simultaneously estimates the 3D location and motion from the monocular images. To alleviate this issue, we leverage the rigid-body assumption for the objects in the driving scenarios and elaborate more temporal frames with constant velocity regularization to further constrain the motion.

E More Related Work

Multi-View 3D Perception Leveraging multi-view images to recover 3D information is a fundamental topic, such as structure from motion [15], multi-view stereo [16], simultaneous localization and mapping [17], etc. One line of methods develop neural-network-based cost volumes [16, 18, 19, 20, 21, 22] to construct cross-frame visual cues for 3D perception. Another line of methods [23, 24, 25] constructs geometry constraints and leverage optimization techniques to obtain a tight-coupled 3D structure. However, most of the work assumes the scene and objects are static, making them fail to handle the moving objects in driving scenarios.

References

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[3] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18 (10):3337, 2018.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *ICIP*, 2016.

[5] T. Yin, X. Zhou, and P. Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021.

[6] Z. Pang, Z. Li, and N. Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. In *CVPR*, 2021.

[7] T. Wang, X. Zhu, J. Pang, and D. Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, 2021.

[8] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022.

[9] H.-N. Hu, Y.-H. Yang, T. Fischer, T. Darrell, F. Yu, and M. Sun. Monocular quasi-dense 3d object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[10] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 2021.

[11] P. Li and J. Jin. Time3d: End-to-end joint monocular 3d object detection and tracking for autonomous driving. In *CVPR*, June 2022.

[12] N. Marinello, M. Proesmans, and L. Van Gool. Triplettrack: 3d object tracking using triplet embeddings and lstm. In *CVPRW*, 2022.

[13] T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. *arXiv preprint arXiv:2205.00613*, 2022.

[14] J. Yang, E. Yu, Z. Li, X. Li, and W. Tao. Quality matters: Embracing quality clues for robust 3d multi-object tracking. *arXiv:2208.10976*, 2022.

[15] K. Wang and S. Shen. MVDepthNet: real-time multiview depth estimation neural network. In *3DV*, 2018.

[16] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018.

[17] T. Taketomi, H. Uchiyama, and S. Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9, 12 2017. doi:10.1186/s41074-017-0027-2.

[18] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, 2019.

[19] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[20] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[21] Z. Teed and J. Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *CVPR*, 2021.

[22] J. Sun, L. Chen, Y. Xie, S. Zhang, Q. Jiang, X. Zhou, and H. Bao. Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation. In *CVPR*, 2020.

[23] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019.

[24] C. Tang and P. Tan. BA-net: Dense bundle adjustment networks. In *ICLR*, 2019.

[25] P. Li, S. Liu, and S. Shen. Multi-sensor 3d object box refinement for autonomous driving. *arXiv preprint arXiv:1909.04942*, 2019.