CC-3DT: Panoramic 3D Object Tracking via Cross-Camera Fusion

Supplementary Material

In this appendix, we provide additional results on 3D detection performance of CC-3DT on NuScenes [1], an ablation study on the effect of different 3D detectors, technical details about the 3D estimation model, more qualitative results and a runtime analysis.

A 3D detection results on NuScenes

We evaluate the 3D detection results of CC-3DT with DETR3D [2] as 3D detector on the NuScenes validation split. As shown in Table 1, our 3D tracking framework with cross-camera motion model can also improve the overall 3D detection results. Compared to our Kalman Filter baseline (KF3D), the LSTM motion model refines the 3D estimation better and achieves higher mAP and lower mATE (mean Average Translation Error). By learning the cross-camera motion, our framework estimates more accurate object velocity and achieves lower mAVE (mean Average Velocity Error). Overall, CC-3DT improves not only 3D tracking performance but also benefits the 3D detection results.

B Effect of different 3D detectors

We evaluate the overall performance with different 3D detectors with our CC-3DT on the NuScenes dataset. In Table 2, among of all methods, our proposed CC-3DT achieves the best NDS and AMOTA when the most accurate detection results in BEVFormer [3] are used. Thus, we show that our method is robust to the 3D detector being used and that it can benefit from more accurate 3D detections.

C 3D estimation model

The 3D estimation model in CC-3DT consists of a 3D detector and an appearance feature extractor. The appearance features are extracted by the similarity head with the generated 2D bounding boxes as region proposal. We use 4 convolutional layers followed by one fully-connected layer as the similarity head following QDTrack [4, 5], and train the 3D estimation network using the same 3D detector as in QD-3DT [6].

C.1 Similarity head training

We use quasi-dense similarity learning [4, 6, 5] to train our similarity head. Given a key frame at time t, we sample a reference frame within a temporal interval n, where $n \in [-2, 2]$ for NuScenes [1] due to its low sampling frequency and $n \in [-3, 3]$ for Waymo Open [7]. For the estimated 3D bounding boxes $\mathbf{b}_t$, each has its corresponding 2D bounding box $\mathbf{e}_t^d$ on the image $\mathbf{I}_t^m$. We optimize the appearance embedding $\mathbf{f}_t^d$ for each 2D proposal $\mathbf{e}_t^d$ using a multi-positive cross-entropy loss as

$$\mathcal{L}_{\texttt{embed}} = \log[1 + \sum_{\mathbf{p}_{t+n}^d} \sum_{\mathbf{n}_{t+n}^d} \exp(\mathbf{f}_t^d \cdot \mathbf{n}_{t+n}^d - \mathbf{f}_t^d \cdot \mathbf{p}_{t+n}^d)], \tag{1}$$

where the appearance embedding $\mathbf{f}_t^d$ should be similar to its positive reference embeddings $\mathbf{p}_{t+n}^d$, and dissimilar to all its negative reference embeddings $\mathbf{n}_{t+n}^d$. We apply an auxiliary loss based on the

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

Table 1: **Motion model ablation study.** 3D detection results of CC-3DT with different motion models on the NuScenes validation split.

3D Detector	Motion Model	NDS ↑	mAP ↑	mATE ↓	mAVE ↓
DETR3D [2]	-	0.4138	0.3228	0.6803	0.8545
	KF3D	0.4189	0.3244	0.6846	0.8506
	VeloLSTM	**0.4326**	**0.3304**	**0.6594**	**0.7716**

Table 2: **Detector ablation study.** Comparison of 3D tracking performance of CC-3DT using different 3D detectors on the NuScenes validation split.

3D Detector	NDS↑	AMOTA ↑	AMOTP ↓	RECALL ↑	MOTA ↑	IDS ↓
Baseline	0.3868	0.311	1.433	0.472	0.278	2536
DETR3D [2]	0.4326	0.359	1.361	0.498	0.326	**2152**
BEVFormer [3]	**0.4781**	**0.429**	**1.257**	**0.534**	**0.385**	2219

cosine similarity of the appearance embedding $\mathbf{f}_t^d$ in the key frame and its corresponding embedding in the reference frame $\mathbf{f}_{t+n}^d$ as

$$\mathcal{L}_{\text{aux}} = (\frac{\mathbf{f}_t^d \cdot \mathbf{f}_{t+n}^d}{||\mathbf{f}_t^d|| \cdot ||\mathbf{f}_{t+n}^d||} - \mathbb{1}(\mathbf{e}_t^d, \mathbf{e}_{t+n}^d))^2, \tag{2}$$

where $\mathbb{1}(\mathbf{e}_t^d, \mathbf{e}_{t+n}^d)$ is 1 if $\mathbf{e}_t^d$ and $\mathbf{e}_{t+n}^d$ are matched to the same ground truth object and 0 otherwise. The overall loss for similarity head training is

$$\mathcal{L}_{\texttt{similarity}} = \lambda_{\texttt{embed}} \mathcal{L}_{\texttt{embed}} + \mathcal{L}_{\texttt{aux}}, \tag{3}$$

and we use $\lambda_{\texttt{embed}}$ as 0.25 for training on both datasets.

C.2 Baseline 3D detector

We develop the baseline 3D detector based on Faster R-CNN [8] as in [6]. We estimate the 3D center (x, y, z) by regressing a logarithmic depth value, and an offset from the Faster R-CNN generated 2D bounding box center to the projected 3D bounding box center. For the object confidence c, we regress the score by generating the target confidence score as an exponential of negative L1 distance between depth estimation and ground truth. We estimate the object dimensions (l, w, h) by regressing a logarithmic value, and estimate the object orientation θ following Mousavian et al. [9], *i.e.* we learn to classify the angle into 2 bins with binary cross-entropy loss and regress the residual relative to the belonging bin center. We us Huber loss [10] for the 3D box regression losses.

C.3 Different 3D detectors

As shown in Table 2, our proposed CC-3DT works with different existing 3D detectors. To extract appearance embeddings given only 3D bounding boxes, we generate the corresponding 2D bounding boxes by projecting the 3D bounding boxes to the images. We use these 2D bounding boxes to extract the appearance embedding from the image where the 3D bounding box is visible in. In addition, we use the 3D score estimated by the 3D detector as our object confidence for the motion model and only keep the boxes if the score is greater than 0.05.

C.4 3D tracker settings

We use $\mathbf{w}_{\texttt{deep}} = 0.5$ for the affinity matrix $\mathbf{A}$ and 0.5 as the matching score threshold. 0.8 is set as the score threshold to start a new track for the baseline 3D detector, while 0.1 and 0.2 are used for DETR3D [2] and BEVFormer [3] considering the higher quality of the 3D detections. We use 0.5 as the score threshold to continue a track for the baseline 3D detector and 0.05 and 0.1 for DETR3D and BEVFormer. We keep 10 frames for the tracks and 1 frame for the backdrops. Because we use

Table 3: **Runtime analysis.** We analyse the contribution of the different components in our pipeline to the total runtime in seconds on the NuScenes dataset. We measure the runtime of processing a complete sample consisting of 6 images at full resolution (900 × 1600 pixels).

Motion model	3D Detection	Appearance feature	Data association	Motion model	Total
KF3D	0.381	0.002	0.037	0.016	0.436
LSTM	0.381	0.002	0.029	0.072	0.484

lower score threshold for DETR3D and BEVFormer 3D detectors, we do not keep the backdrops for them. Following [4, 6, 5], we apply 0.8 as the momentum to update the appearance features inside the tracks We use 0.7 and 0.3 2D IoU threshold for duplicate removal on each camera for the new detections and the backdrops.

D Runtime analysis

We provide a breakdown of the runtime of our pipeline in Table 3. We measure the runtime using a single RTX 3090 graphics card at batch size 1 on the NuScenes dataset. One batch element consists of 6 images at 900 × 1600 pixel resolution. Note that the setup can differ between datasets and that this may affect the runtime, depending on how many images need to be processed simultaneously. We compare the runtime of our LSTM motion model with the Kalman Filter (KF3D) baseline, and find that the LSTM is slower (0.072s vs. 0.016s). However, we observe that the majority of the total runtime is occupied by the 3D detection method, while our data association and motion modeling is responsible only for a minor part of the total runtime. We ablate the effect of using different detectors with our method in section B. Note that the motion model used affects the association results, which influences the runtime of the data association.

E Qualitative results

We show qualitative results on the Waymo Open and NuScenes datasets in Figure 1, Figure 2 and Figure 3, respectively. We plot the 3D bounding boxes in the image view and depict object identity as box color. For NuScenes, we provide both front and back cameras results and show that our CC-3DT can associate the objects across camera borders.

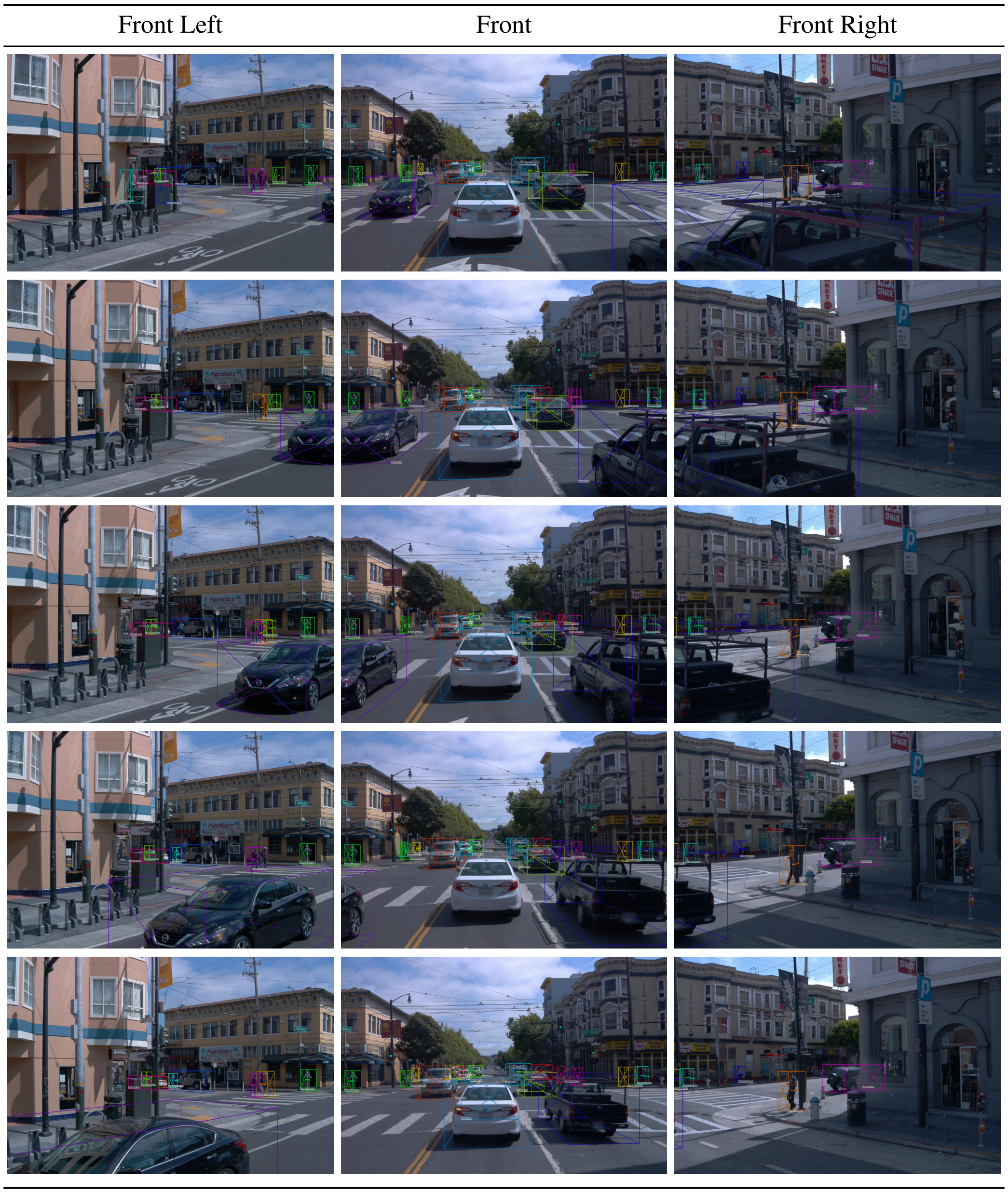

Figure 1: Qualitative results of our tracker on the Waymo Open validation split. Note the consistent identity of objects moving along camera borders.

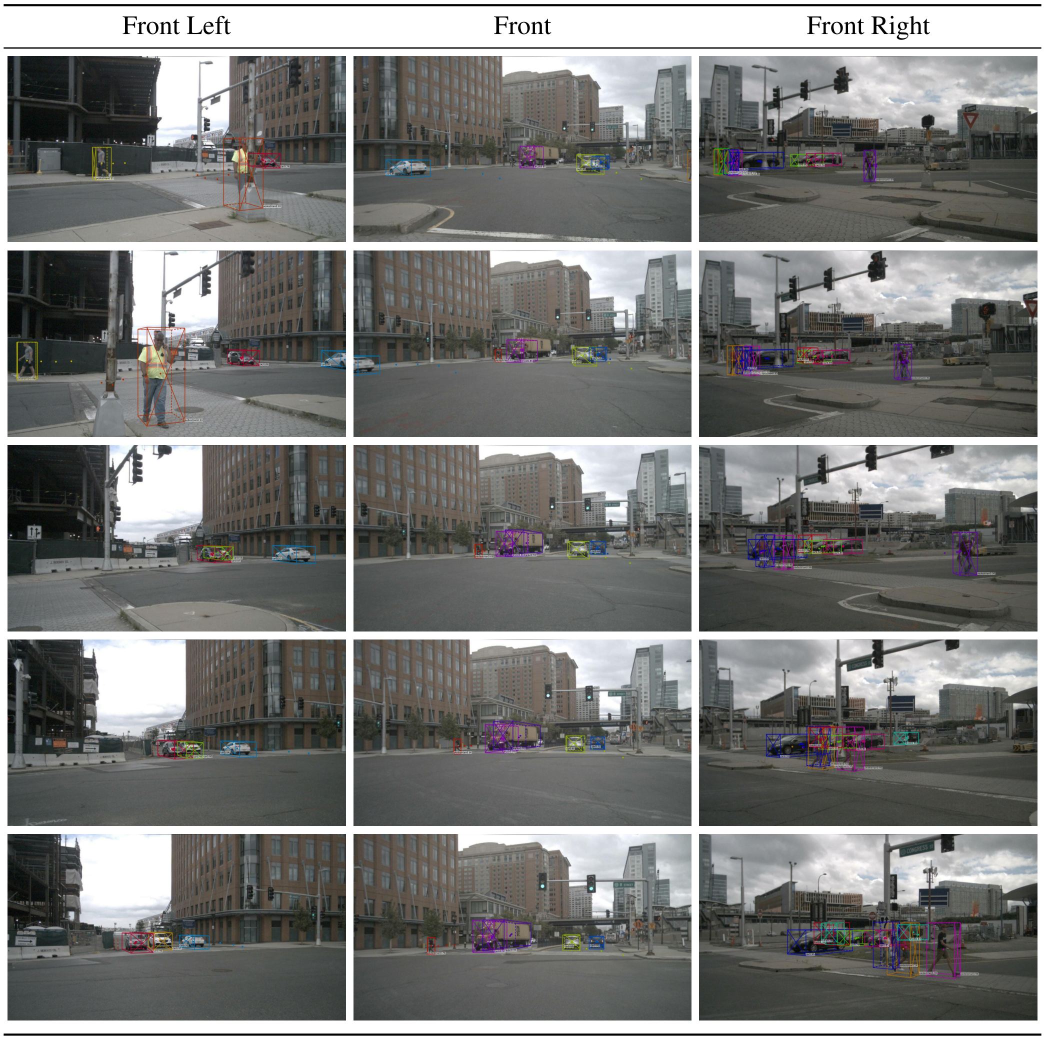

Figure 2: Qualitative results of our tracker on the front camera of the NuScenes validation split. Note the consistent identity of objects moving along camera borders.

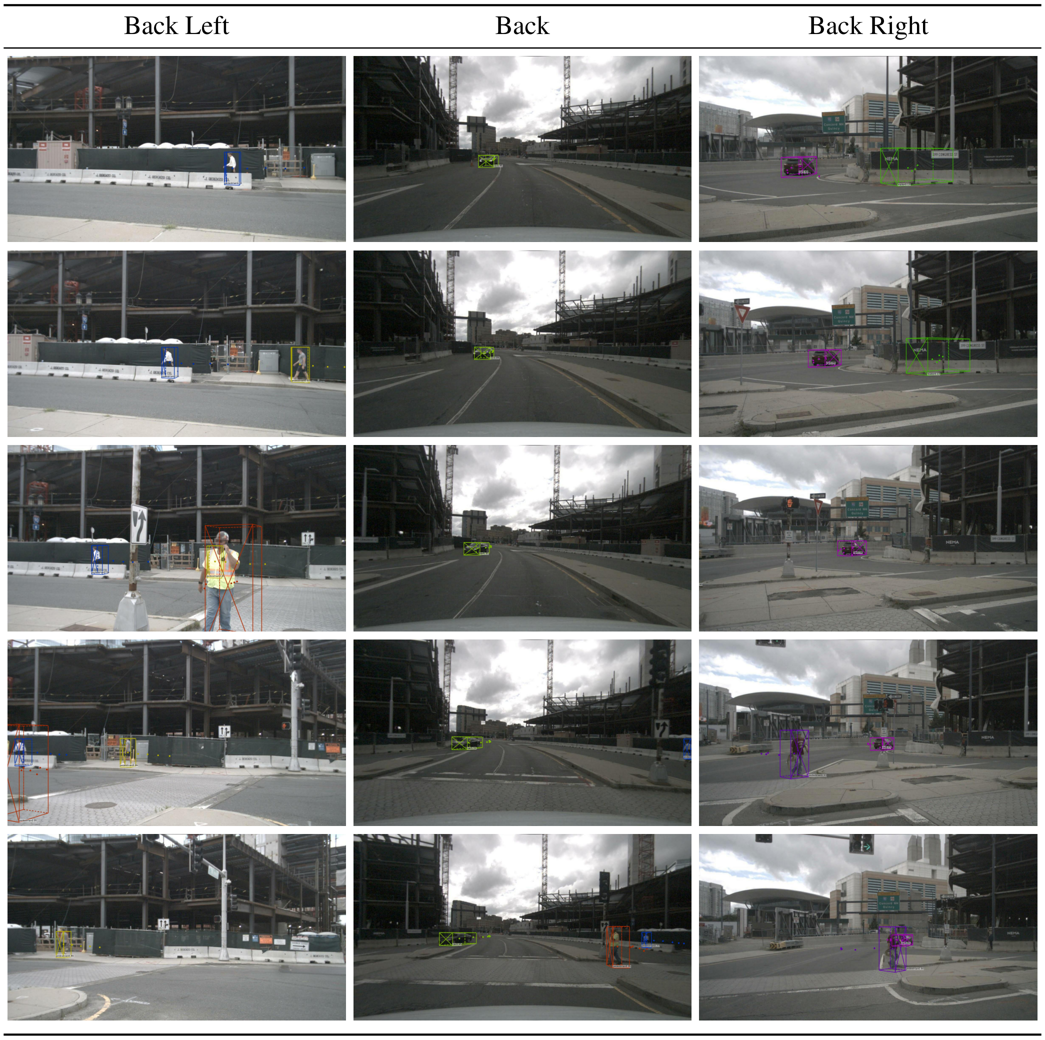

Figure 3: Qualitative results of our tracker on the back cameras of the NuScenes validation split. Note the consistent identity of objects moving along camera borders.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[2] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, , and J. M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *The Conference on Robot Learning (CoRL)*, 2021.

[3] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022.

[4] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu. Quasi-dense similarity learning for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[5] T. Fischer, J. Pang, T. E. Huang, L. Qiu, H. Chen, T. Darrell, and F. Yu. Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking. *arXiv preprint arXiv:2210.06984*, 2022.

[6] 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.

[7] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[9] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3D bounding box estimation using deep learning and geometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[10] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 1964.