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

Anonymous Author(s)
Affiliation
Address
email

Abstract: Recent multi-camera 3D object detectors usually leverage temporal information to construct multi-view stereo that alleviates the ill-posed depth estimation. However, they typically assume all the objects are static and directly aggregate features across frames. This work begins with a theoretical and empirical analysis to reveal that ignoring the motion of moving objects can result in serious localization bias. Therefore, we propose to model Dynamic Objects in RecurrenT (DORT) to tackle this problem. In contrast to previous global Bird-Eye-View (BEV) methods, DORT extracts object-wise local volumes for motion estimation that also alleviates the heavy computational burden. By iteratively refining the estimated object motion and location, the preceding features can be precisely aggregated to the current frame to mitigate the aforementioned adverse effects. The simple framework has two significant appealing properties. It is flexible and practical that can be plugged into most camera-based 3D object detectors. As there are predictions of object motion in the loop, it can easily track objects across frames according to their nearest center distances. Without bells and whistles, DORT outperforms all the previous methods on the nuScenes detection and tracking benchmarks with 62.8% NDS and 57.6% AMOTA, respectively. The source code will be released.

Keywords: Temporal Modeling, 3D Object Detection

1 Introduction

Multi-camera 3D object detection is critical to robotic systems such as autonomous vehicles, humanoid robots, and *etc*. As object depth estimation from a single image is naturally ill-posed, recent works use large-scale depth pre-trained models [1] and leverage geometric relationships [2, 3, 4, 5] to alleviate the problem. Because stereo correspondence exists in consecutive frames, some works resort to temporal information for accurate depth predictions. For example, BEVDet4D [6] and BEVFormer [7] warp preceding features to the current frame to enrich the single-frame BEV representations. DfM [5] constructs temporal cost volumes that explicitly establish the stereo correspondence. However, these cross-frame feature aggregations do not consider the motion of moving objects and assume all the objects are static, which results in serious 3D localization bias.

In this paper, we first provide a theoretical and empirical analysis to reveal the negative effects of inaccurate object motion to object depth (Fig 1). In particular, if the object is moving, the incorrect temporal correspondence would derive a biased depth. In the driving scenarios, it is critical that a misleading depth is estimated, which might reduce the reaction time of the decision system, leading to catastrophic collision accidents. This motivates us to devise an explicit mechanism to involve object motion estimation in the temporal-based 3D detection pipeline.

Modeling dynamic objects in this context has several challenges: (1) We need a flexible object-wise representation for potential object-wise operations based on motion modeling. (2) Jointly estimating

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

object location and motion is an inherent chicken and egg problem [8]: The temporal correspondence can derive accurate object location only when accurate object motion is given and vice versa. (3) Simultaneously predicting object location and motion from only two frames is also an ill-posed problem theoretically, and thus it is desired to involve right-body assumption and more frames to pose reasonable constraints.

To address these problems, we model **D**ynamic **O**bjects in **R**ecurren**T** (DORT) that simultaneously estimates object motion and location, and then progressively refines them for accurate 3D object detection. It benefits from a local 3D volume representation that not only extracts object-wise 3D features but also alleviates the heavy computational costs of global BEV in previous methods [5, 7, 9]. Based on the object-wise volume, temporal volumes are constructed by warping the volumes from the preceding frame to the current frame according to the object motion. Then the obtained cost volumes act as the features for updating the candidate location and motion. We model this estimation and update pipeline as a recurrent process to alleviate the aforementioned chicken and egg problem. In addition, our framework can take into more than two frames and pose constraints to the object motion. It inherently provides a feasible solution to avoid the ill-posed dilemma of estimating object location and motion from only a single pair of correspondence. As there is object motion prediction in the loop, the framework is naturally capable of joint object detection and tracking by utilizing object motion to align the detection results into the same timestamp. It also can be plugged into most camera-based 3D object detectors for flexible and practical use.

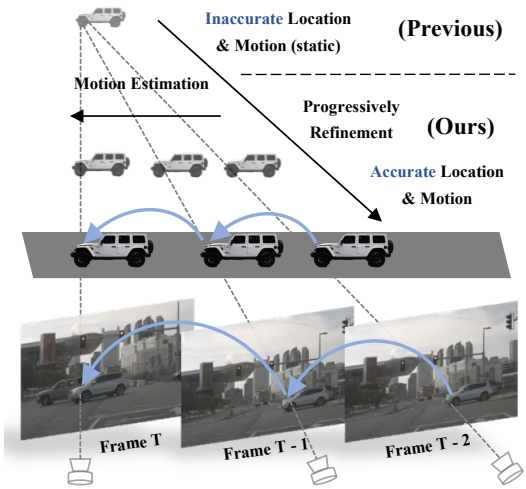


Figure 1: Visualization of object localization from temporal correspondence. Previous works ignore the motion of moving objects, leading to imprecise localization. Our work progressively refines the object's location and motion so that the preceding features can be precisely aggregated.

We validate the effectiveness of our framework on the nuScenes detection and tracking benchmarks. Benefiting from the dynamic objects modeling, DORT outperforms all the previous methods with a large margin, leading to 62.8% nuScenes detection metric (NDS) and 57.6% and average multi-object tracking accuracy (AMOTA), respectively.

2 Related work

Monocular 3D Object Detection Monocular-based 3D object detection was first approached from the single-frame scenario and evolved into multi-frame to alleviate the ill-posed depth estimation.

(a) Methods with A Single Frame The single-frame-based methods [10, 11, 3, 12, 13] first extend 2D object detectors and insert several 3D attribute regression heads to predict 3D bounding boxes. To alleviate the ill-posed depth recovery, several methods improve the model from the perspectives of loss function [11], network architecture [14, 15], regression objective [3, 4], *etc.* Besides directly regressing depth, later approaches [2, 16, 17] further design 2D-3D geometry constraints to better extract visual cues for depth estimation. To align detection features with the output space, another line of methods [18] designs several transformation modules to lift 2D inputs into 3D space. Pseudo-lidar-based methods [19, 20, 21, 21] first predict the per-pixel depth and convert the raw pixel into point cloud for 3D detection. BEV-based methods [18, 22, 9, 23] propose orthographic feature transformation (OFT) to transform the 2D features into 3D voxels and then adopt a LiDAR-based head to localize objects. Later works improve the OFT by explicit depth distribution modeling [22, 9, 23], incorporating deformable attention module [7] or designing 3D position encoding [24, 25].

(b) Methods with Multiple Frames Although many techniques are designed in single-frame-based methods, they still suffer from ill-posed depth recovery, leading to unsatisfactory performance for deployment. To augment the single-view observation, recent works [26, 14, 5, 6, 7, 27] leverage previous frames as additional observations for features augmentation. Kinematic3D [14] leverages 3D Kalman Filter to associate objects across frames and refines 3D boxes. Later studies [5, 6, 7] construct cross-frame cost volumes as another visual cue for 3D detection. The cost volumes are based on the multi-view stereo, which assumes objects are static across frames. However, this assumption does not align with the driving scenario, where the objects can move.

Monocular 3D Object Tracking 3D object tracking associates objects across frames and generates a set of trajectories for motion prediction. Traditional methods adopt a tracking-by-detection paradigm that first detects objects in each frame and then associates them by appearance features [26] or objects' displacement with Kalman filter [28, 29, 30, 31, 32, 33]. Besides the above paradigm, several methods [34, 35] design a two-stage paradigm that first associates objects and then utilizes the temporal motion to improve the detection performance. In this work, we utilize temporal cost volumes to bridge the spatial location and temporal motion and derive a recurrent paradigm that iteratively updates them to obtain tightly coupled results for joint 3D detection and tracking.

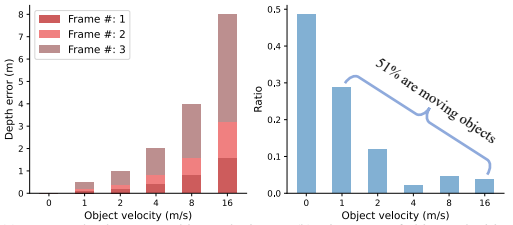
3 Object Motion in Temporal Modeling

In this section, we first conduct analysis to demystify the adverse effects of neglecting object motion in temporal modeling, then discuss the challenges of modeling 3D motion in the monocular setting.

Localization Bias from the Static Assumption In previous methods [5, 6, 7], the object motion is ignored by assuming objects are static and the features are directly aggregated after converting the past frames to the current frame. We first show that the static assumption would derive a biased depth. Without loss of generality, we consider the two-view case, and it can be naturally extended to more than two views. We denote the camera intrinsic as K with focal length and center offset (f, c_u, c_v), and the ego motion and object motion from frame t_0 to frame t_1 as $T^{ego}_{t_0 \to t_1}$ and $T^{obj}_{i \to j}$:

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

For simplicity, we assume the ego and object motion only contain the translation $(x, 0, z)$ on the horizontal plane. The analysis can be easily extended to the case that the motion contains rotation. Given the temporal images, we can utilize photometric or featuremetric similarity to find the correspondence of pixel $p_{t_0} = (u_{t_0}, v_{t_0})$ in the frame t_0 and pixel $p_{t_1} = (u_{t_1}, v_{t_1})$ in the frame t_1. Then, the depth z_{t_1} can be recovered by:



(a) Average depth error vs object velocity (b) Histogram of object velocities

Figure 2: Empirical analysis of the depth bias on the nuScenes dataset if objects are assumed static.

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

where π denotes the 2D to 3D projection. This derivation assumes that the object is static. However, objects can move with a corresponding motion $T^{obj}_{t_0 \to t_1}$. Then, the object depth is revised as follows:

$$T^{obj}_{t_0 \to t_1} T^{ego}_{t_0 \to t_1} \cdot \pi(p_{t_0}, K) = \pi(p_{t_1}, K), \quad \hat{z}_{t_1} = \frac{(z^{ego} + z^{obj})(u_{t_0} - c_u) - f(x^{ego} + x^{obj})}{u_{t_0} - u_{t_1}}. \tag{3}$$

Based on Eq (2) and (3), we can obtain the depth gap:

$$\Delta z = \frac{z^{obj}(u_{t_0} - c_u) - f x^{ego}}{u_{t_0} - u_{t_1}}. \tag{4}$$

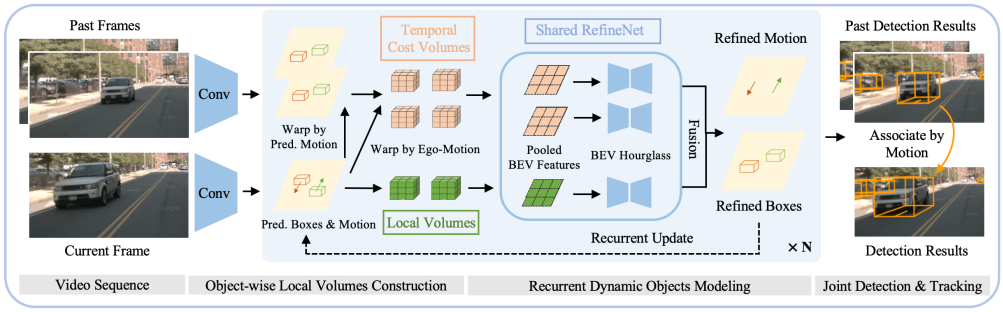


Figure 3: Pipeline overview. Given a video sequence, we first extract the 2D features and generate the candidate boxes and motion by a single-frame detector. Then the boxes and motion are progressively refined from the concurrently updated 3D volume features. A fusion process in the recurrent module combines the estimation from each pair of frames. Based on the tightly coupled modeling of object location and motion, the framework can achieve joint 3D detection and tracking during inference.

From Eq (4), we can observe that the depth bias is linearly correlated with the object motion. In Fig 2, we also display the empirical statistics of object motion and the corresponding depth bias from the nuScenes dataset. We can observe that the empirical depth error is also correlated with the object velocity and increases as the time interval enlarges. Besides, the right part in Fig. 2 also shows that almost 51% of objects are moving across frames, demonstrating the necessity of modeling object motion in the temporal-based framework.

Ill-Posed Problem in Motion Modeling Except for demonstrating the necessity of modeling object motion, we also want to mention that simultaneously estimating object location and motion is nontrivial, especially in the two-frame case. As shown in Fig 1, the correspondence of two points can come from infinite combinations of object location and motion. This illustrates that joint location and motion estimation from only one correspondence is ill-posed. To alleviate this issue, we first simplify the object motion as a right-body movement so that multiple correspondences from the points in the object can be used to solve a shared motion. Furthermore, we leverage more than two frames to constrain the flexibility of object motion. More details can be referred to Sec 4.3.

4 Methodology

This section describes the details of DORT. DORT is a general joint detection and motion prediction module that can estimate coupled object location and motion results across frames. Based on the tightly coupled location and motion results, DORT is also capable of simultaneously 3D object detection and tracking. Basically, it can be based on most temporal 3D detectors [6, 36, 7]. In this work, we select the popular temporal detector BEVDepth [23] as the base detector and extend it to handle both static and moving objects in temporal modeling. We first present an overview of temporal-based frameworks in Sec 4.1 and then introduce our modifications: the local volume for object-wise representation in Sec 4.2, the key recurrent dynamic objects modeling in Sec. 4.3, and the object association for monocular 4D object detection in Sec 4.4.

4.1 Overview of Temporal-Based Frameworks

Current temporal-based methods contain three stages: (1) The 2D features extraction stage extracts the features from the 2D images. (2) The view transformation and stereo matching process that first lift the 2D features to a 3D volume and then warp the features in each frame to an aligned canonical space for matching. Depending on the model design, the order of view transformation and stereo matching may reverse. (3) The detection stage takes the 3D features to estimate 3D bounding boxes.

In this work, we follow previous methods [5, 9, 23] and adopt the widely-used 2D backbone (*e.g.* ResNet [37]) to do the features extraction. For the view transformation stage, we design an **object-wise local volume** that leverages the candidate 3D boxes to obtain the potential foreground regions and only models them with local object-wise 3D volumes. For the stereo matching and detection

stages, we propose a **recurrent dynamic objects modeling** module to progressively refine the detection and motion results for accurate 3D temporal features.

4.2 Object-wise Local Volume

In previous works, the 2D-3D transformation considers each candidate 3D grid point and constructs a global volume for detection. However, there are several limitations: (1) the global volume contains lots of background regions, which is not vital for detection but increases the computation burden. (2) Modeling a global volume needs to pre-define a detection range during training, making the detectors fail to detect objects with arbitrary depth. (3) It is inconvenient to manipulate a global volume with object-wise operations.

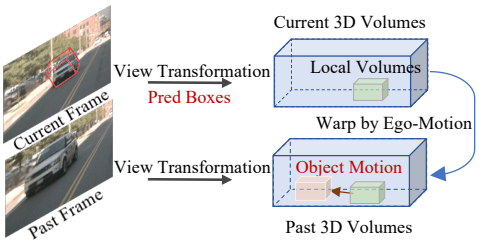


Figure 4: The process of extracting local volumes in the current and past frames according to predicted bounding boxes and object motion.

Hence, we replace the global volume with an object-wise local volume. Specifically, we leverage the candidate boxes to determine the 3D region of interest (RoI) and set the local volume center as the bounding box center. To keep the object ratio and achieve cross-view warping, we assign each 3D RoI volume $V \in \mathcal{R}^{W\times H\times L\times C}$ with the same 3D dimension (W, H, L) and channel size C. Different from 2D detection, the objects' dimension in 3D space has less variance and empirically relies less on the RoI-Align [38] operation. We display the construction of object-wise local volumes in Fig 4. For the 2D to 3D transformation, we first follow LiftSplat [22] and lift the images to a 2.5D frustum by weighting with depth probability. Then we utilize the grid sample operation to warp the features from the 2.5D frustum to each 3D local volume. Benefiting from the accurate 2D detection performance, the local volume features sampled from the 2.5D frustum would have a large overlap with the foreground objects. Hence, if the proposal 3D location is inaccurate, the later refinement module still can use the features for refinement.

4.3 Recurrent Dynamic Objects Modeling

The pipeline of the recurrent framework is illustrated in Fig. 3. Given the candidate 3D bounding boxes and motion as input, each iteration first constructs the temporal cost volumes thereon, and aggregates these cues to refine the proposal boxes and motion. In particular, we adopt a perspective-view based 3D detector (*i.e.* PGD) to generate the initialized candidate 3D boxes and motion and only predict their residuals for refinement in the subsequent recurrent updates.

Cross-Frame Cost Volumes Construction Given the initial predictions of 3D boxes and their motion, we first obtain object-wise volume features following Sec. 4.2. Then we can construct the temporal cost volumes by warping features from past frames to the current frame coordinates based on *ego-motion*. In contrast to previous works [5, 6] assuming objects are static, we further involve the *object motion* into the warping procedure. Specifically, for each point $p \in \mathcal{R}^3$ in the object-wise local volume V, we query the corresponding features in previous frame $t-\Delta t$ with the consideration of ego-motion T^{ego} and the object motion T^{obj} and construct the cost as $\left[V(p), V_{t-\Delta t}(T^{obj}T^{ego}p)\right]$. Note that we simplify the point motion as the object motion with a rigid-body assumption, which can approximate most of the cases in driving scenarios, especially for vehicles [26, 39].

3D Boxes and Motion Residual Estimation Given the object-wise temporal features built from input 3D boxes and motion, we leverage a refinement network to estimate the residual between the input 3D boxes and motion with the ground truth. The refinement network contains several 2D/3D residual-based convolutional layers to extract the 3D volumes and 2D BEV features. The detailed architecture is presented in Supplementary. Formally, the refinement is formulated as the regression of 3D attribute residuals $\mathcal{B}$, including the object's 3D center x, y, z, 3D size w, h, l, rotation θ and velocity v_x, v_y. Since we use the object velocity in the current frame to represent the object motion and assume constant velocity across frame, the supervision for some frames may contain noise(*e.g.* inaccurate labels, violation of rigid-body assumption, etc). Hence, we model the residual

Table 1: Experimental results of monocular 3D object detection and tracking on the nuScenes test set. The input resolution is 1600×900 with using ConvNeXt-Base [40] as the backbone.

(a) 3D detection results.

Method	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
Ego3RT [41]	42.5	0.55	0.26	0.43	1.01	0.14	47.3
UVTR [42]	47.2	0.57	0.25	0.39	0.51	0.12	55.1
BEVFormer [7]	48.1	0.58	0.25	0.37	0.37	0.12	56.9
PETRv2 [43]	51.2	0.55	0.25	0.36	0.40	0.13	58.6
BEVDepth [23]	52.0	0.45	0.24	0.35	0.35	0.13	60.9
BEVStereo [27]	52.5	0.43	0.24	0.36	0.35	0.14	61.0
SOLOFusion [36]	54.0	0.45	0.26	0.37	0.27	0.14	61.9
DORT (Ours)	**55.3**	0.43	0.26	0.42	**0.24**	0.14	**62.8**

(b) 3D tracking results.

Method	AMOTA↑	AMOTP↓	MOTAR↑
QD-3DT [34]	21.7	1.550	56.3
Time3D [35]	21.4	1.360	-
PolarDETR [30]	27.3	1.185	60.7
MUTR3D [44]	27.0	1.494	64.3
SRCN3D [31]	39.8	1.317	70.2
QTTrack [32]	48.0	1.100	74.7
UVTR [42]	51.9	1.125	76.4
DORT(Ours)	**57.6**	**0.951**	**77.1**

as a Laplacian distribution and design the loss function as:

$$\mathcal{L}_{refine} = \sum_{b \in \mathcal{B}} \left(\frac{\sqrt{2}}{\sigma_b} \|\Delta\hat{b} - \Delta b\| + \log \sigma_b \right). \tag{5}$$

Here, Δb, $\Delta\hat{b}$, and σ_b are all the network outputs, and represent the ground truth residual, the estimated residual, and the estimated standard deviation of residual for each 3D attribute, respectively.

Multiple Estimation Fusion Given n frame as inputs, we can obtain n 3D volumes (1 for the local volumes from the current view and $n-1$ for the paired cross-view cost volumes) and obtain n estimated residuals from the residual estimation module. Then we weigh the importance of each residual by the estimated deviation and fuse them to obtain an ensemble result:

$$\hat{b}_{fused} = \sum_{i=1}^{n} \frac{e^{\sigma_{b_i}} b_i}{\sum_{i=1}^{n} e^{\sigma_{b_i}}}, \tag{6}$$

where i denotes the volume index. For simplicity, here we only estimate the velocity measurement for the referenced frame, *i.e.*, the fluctuation of object velocity across different frames would not be considered explicitly. The mechanism for multi-frame fusion is expected to handle this problem adaptively. At the same time, this constraint also provides additional cues when simultaneously estimating object location and motion from more than two frames.

Recurrent Location and Motion Update After each iteration, we can obtain the refined bounding boxes with their motion and thus can derive the updated bounding boxes in different frames. With these updated locations and RoIs, we can further update the volume features and proceed to the next-round refinement. Note that any complex or learnable motion modeling can be integrated into this procedure. Here, to be consistent with the multiple estimation fusion designs, we still keep the constant velocity prediction for simplicity to derive the bounding boxes of previous frames.

In the training stage, we follow the recurrent methods [45, 39] in other tasks and set the loss weight for each iteration as the same. The overall loss is represented as:

$$\mathcal{L} = \mathcal{L}_{pv} + \sum_{i=1}^{k} \mathcal{L}^{i}_{refine}, \tag{7}$$

where $\mathcal{L}_{pv}$ is the loss in the perspective-view detector [4], $\mathcal{L}^{i}_{refine}$ is the refinement loss in each iteration and $k = 3$ is the number of the iterations. In the inference stage, we first take the perspective-view detector (*i.e.* PGD [4]) to generate the initial 3D boxes and their motion and then progressively refine them. In each iteration, we first construct the volume features as discussed in Sec 4.3 and feed them into the refinement module to estimate the 3D boxes and motion residuals for each paired frame input. Then we utilize the multiple estimation fusion module in Sec 4.3 to fuse the estimated results and obtain the refined 3D boxes and motion as the next stage input.

4.4 Monocular 4D Object Detection

So far, we have introduced our recurrent framework for 3D detection from monocular videos. Based on progressive refinement, our model can estimate tightly coupled object location and motion results and thus can easily associate the object detection results across frames, leading to joint 3D detection

Table 2: Experimental results on the nuScenes validation set. The input resolution is 704×256 using ResNet-50 as the backbone. * denotes the re-implementation based on the provided code.

Methods	# frame	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
PGD* [4]	1	28.8	0.75	0.27	0.52	1.13	0.18	37.0
BEVDet [9]		29.8	0.73	0.28	0.59	0.86	0.24	37.9
PETR [25]		31.3	0.77	0.28	0.56	0.92	0.23	38.1
DETR3D [24]		34.9	0.72	0.27	0.38	0.84	0.20	43.4
BEVDet4D [6]	2	32.2	0.70	0.28	0.50	0.35	0.21	45.7
BEVDepth [23]		35.1	0.64	0.27	0.48	0.43	0.20	47.5
DORT (Ours)		**37.9**	0.62	0.27	0.45	**0.31**	0.20	**50.4**
BEVDepth* [6]	8	39.8	0.57	0.27	0.49	0.27	0.18	52.3
DORT (Ours)		**41.8**	0.57	0.26	0.43	**0.25**	0.19	**53.4**
SOLOFusion [36]	16	42.7	0.57	0.27	0.41	0.25	0.18	53.4
DORT (Ours)		**43.6**	0.56	0.26	0.41	**0.24**	0.18	**54.0**

and tracking. Specifically, we follow [28, 46, 29, 47] and associate the detection results by warping current detection to the past frames with object motion. Based on the ego-motion, we first convert the predicted object location to the past frame coordinate and then warp with the estimated object velocity. Then we follow the popular distance-based tracker in 2D task [28, 29] and associate the objects by the closest distance matching. We also utilize the Kalman filter to maintain the trackers' location and velocity. We provide more details of the tracking pipeline in the supplemental materials.

5 Experiments

5.1 Experimental Setup

In this section, we describe the used dataset, the evaluation metrics, and the implementation details.

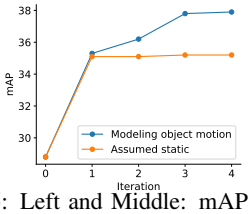
Dataset NuScenes [48] is a large-scale driving dataset, which contains 1,000 video sequences. The official protocol splits the video sequences into 700 for training, 150 for validation, and 150 for testing. Each sequence is annotated with the objects' 3D bounding box, velocity, and tracking id.

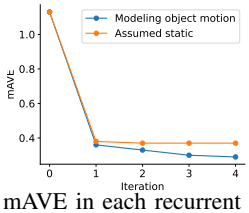
Network Details As discussed in Sec 4.3, the recurrent module requires a proposal detector to generate candidate foreground regions as the 1st stage input. We adopt the popular monocular 3D detector PGD [4] due to its high 2D object detection recall. Following [9, 6, 23], we adopt the ResNet-50 [37] with FPN as the 2D feature extractor and mainly conduct experiments on this setting. The 2D feature extractors in PGD and the recurrent module are shared to save computation time. The grid size of the 3D volume is set as $0.8m$ with the range of $[-5m, 5m]$ in the X and Z (depth) axis and $[-4m, 2m]$ in the Y (height) axis. During 2D to 3D features transformation, we follow [23] and adopt the depth distribution guided 2D to 3D features lifting. Regarding the test set submission, we follow [23, 27] and adopt the ConvNeXt-Base [40] as the image backbone. The image backbone is initialized with ImageNet pre-trained weights, and no other external data is used. We provide more details about the network architecture of the recurrent module and the training configuration in the supplementary materials.

Training Configurations The model is optimized by AdamW optimizer with weight decay 10^{-2}. We first follow [4] to train the proposal detector and refine the recurrent module with 24 epochs, where the initial learning rate is set as 2×10^{-4} and decreases to 2×10^{-5} and 2×10^{-6} at the 18th and 22th epochs. Following [9, 23], we use the class balance sampling strategy (CBGS) to alleviate the class imbalance problem. We adopt the commonly used 2D data augmentation that randomly flips the image, resizes the image with the range of $[0.36, 0.55]$, and crops the image to the resolution of 704×256. Regarding the input video sequences, we follow [6, 23] and sample the preceding keyframes to obtain the past video sequences. Regarding the test set submission, we enlarge the input resolution to 1600×640 and reduce the volume size to $0.4m$.

5.2 Main Results

In Table 1, we first provide the comparison of our framework with existing state-of-the-art methods on the nuScenes test benchmarks. We draw the following observations: (i) Benefiting from dynamic objects modeling, our method displays a significant improvement in both object detection (mAP) and motion estimation (mAVE), and 0.5% and relatively 7.4% better than the previous best method.





Setting	mAP	NDS	mAVE
Assumed static	35.0	47.1	0.37
GT motion	39.3	-	-
Pred motion	37.9	52.1	0.31

Figure 5: Left and Middle: mAP and mAVE in each recurrent iteration step (1 past frame is used). Right: Experimental results of different motion modeling strategies.

These localization and motion estimation improvements also contribute to state-of-the-art results in terms of the nuScenes detection metric (NDS). (ii) With strong localization and motion estimation results, our tracking module can better associate the detected objects in different timestamps, resulting in superior performance over all the other trackers with different metrics. Specifically, we improve the second best tracker [42], another distance-based tracker with 10.9% and 7.5% relative improvements on the AMOTA and AMOTP metrics. Compared with the joint detection and tracking methods QDTrack3D [34] and Time3D [35], the performance gain of our method demonstrates the effectiveness of our dynamic objects modeling framework in jointly modeling object motion and location. (iii) In Table 2, we also report our method on the nuScenes validation set with different settings. For the detection performance, we can draw the same observation as in the test set that our method can outperform previous temporal-based methods [6, 23, 36] in terms of mAP and mAVE. Note that our method is also compatible with the components designed in the current BEV-based frameworks, such as the training techniques in BEVDepth [23] and the depth estimation module in SOLOFusion [36]. Furthermore, the local 3D volume is more friendly to practical applications, which can handle objects with arbitrary depth in the image.

5.3 Ablation Study

Ablation Study of Object Motion We further validate the influence of different dynamic object modeling strategies on the detection performance The first experiment compares the assumed static case with that of using the ground truth object motion. As shown in Table 5, the model with ground truth object motion outperforms the assumed static with 4.3% mAP, demonstrating the necessity of object motion for obtaining accurate temporal correspondence features. When we replace the ground truth object motion with an estimated one, it still can bring 2.9% mAP improvements, illustrating the usefulness of our dynamic objects modeling module.

Experiments with Different Iterations In Figure 5, we provide the comparison of modeling object motion and assumed static with different recurrent iterations. Benefiting from the BEV features modeling, the two configurations display almost 2% mAP improvements in the first iterations. In the later iterations, the improvement in assumed static stops, mainly due to the lack of accurate temporal features. While with more and more accurate temporal features, the model with modeling object motion can progressively improve the detection and motion estimation results.

6 Conclusion and Limitations

This work proposes a novel framework to better leverage temporal information for camera-only 3D detection by modeling dynamic objects. We first design an object-wise local volume to save computation time and maintain an object-wise representation for motion and detection modeling. Then we propose a recurrent module to tackle the challenging motion and location modeling problem. Specifically, we progressively update the motion and location results from the concurrently updated 3D volume features. As the object motion and location results are tightly coupled in the recurrent stage, we also demonstrate the framework can naturally achieve 3D tracking.

Although DORT can better handle dynamic objects, we follow previous methods and simplify the motion with a constant velocity assumption. To handle various kinds of scenarios, this assumption could be further relaxed with modeling acceleration. Additionally, similar to current object-wise feature modeling methods [24, 25, 49] (*i.e.* DETR-based, Two-stage-based), the computation is linearly correlated with the number of instances. To overcome this limitation, point-wise motion modeling and feature extraction with object-wise grouping will be considered in future work.

References

[1] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021.

[2] P. Li, H. Zhao, P. Liu, and F. Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, 2020.

[3] Y. Zhang, J. Lu, and J. Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, 2021.

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

[5] T. Wang, J. Pang, and D. Lin. Monocular 3d object detection with depth from motion. In *ECCV*, 2022.

[6] J. Huang and G. Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022.

[7] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.

[9] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

[10] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.

[11] X. Ma, Y. Zhang, D. Xu, D. Zhou, S. Yi, H. Li, and W. Ouyang. Delving into localization errors for monocular 3d object detection. In *CVPR*, 2021.

[12] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang. Monocular 3d object detection: An extrinsic parameter free approach. In *CVPR*, 2021.

[13] G. Brazil and X. Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 2019.

[14] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, 2020.

[15] T. Wang, X. Zhu, J. Pang, and D. Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *ICCVW*, 2021.

[16] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. In *ICCV*, 2021.

[17] Q. Lian, P. Li, and X. Chen. Monojsg: Joint semantic and geometric cost volume for monocular 3d object detection. In *CVPR*, 2022.

[18] T. Roddick, A. Kendall, and R. Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

[19] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019.

[20] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019.

[21] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019.

[22] J. Philion and S. Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020.

[23] 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.

[24] 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 *CoRL*, 2021.

[25] Y. Liu, T. Wang, X. Zhang, and J. Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022.

[26] P. Li, J. Shi, and S. Shen. Joint spatial-temporal optimization for stereo 3d object tracking. In *CVPR*, 2020.

[27] Y. Li, H. Bao, Z. Ge, J. Yang, J. Sun, and Z. Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022.

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

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

[30] S. Chen, , X. Wang, T. Cheng, Q. Zhang, C. Huang, and W. Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv:2206.10965*, 2022.

[31] Y. Shi, J. Shen, Y. Sun, Y. Wang, J. Li, S. Sun, K. Jiang, and D. Yang. Srcn3d: Sparse r-cnn 3d surround-view camera object detection and tracking for autonomous driving. *arXiv:2206.14451*, 2022.

[32] 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.

[33] T. Fischer, Y.-H. Yang, S. Kumar, M. Sun, and F. Yu. CC-3DT: Panoramic 3d object tracking via cross-camera fusion. In *CORL*, 2022.

[34] 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.

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

[36] J. Park, C. Xu, S. Yang, K. Keutzer, K. M. Kitani, M. Tomizuka, and W. Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In *The Eleventh International Conference on Learning Representations*, 2023.

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

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

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

[40] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. *CVPR*, 2022.

[41] J. Lu, Z. Zhou, X. Zhu, H. Xu, and L. Zhang. Learning ego 3d representation as ray tracing. In *ECCV*, 2022.

[42] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia. Unifying voxel-based representation with transformer for 3d object detection. In *NeurIPS*, 2022.

[43] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, X. Zhang, and J. Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv:2206.-1256*, 2022.

[44] 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.

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

[46] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking objects as points. In *ECCV*, 2020.

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

[48] 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.

[49] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.