

# 3D Object Detection and Tracking on Streaming Data

Xusen Guo<sup>1</sup> and Kai Huang<sup>2</sup>

**Abstract**—Recent approaches for 3D object detection have made tremendous progress due to the development of deep learning. However, previous researches are mostly single frame based, information between frames is scarcely explored. In this paper, we attempt to leverage the temporal information in streaming data and explore 3D object detection as well as tracking based on keyframes. Towards this goal, we set up a ConvNet architecture that can associate keyframes images and keyframes point clouds to generate accurate 3D detections and trajectories in an end-to-end form. Specifically, a tracking module is introduced to capture objects co-occurrences across time, and a motion based interpolation algorithm is proposed to generate streaming level results given keyframe detections. Our proposed architecture is proven to produce competitive results on the KITTI Object Tracking Benchmark, with 72.21% in MOTA and 82.29% in MOTP respectively.

## I. INTRODUCTION

3D Object detection has received increasing attention over the last few years due to the rapid development of autonomous driving. Compared to 2D image, 3D information can provide accurate localization of targets and characterize their shapes. Current approaches for 3D object detection are mostly carried out in three fronts: image based [1], [2], point clouds based [3], [4], [5], and multi-view fusion based [6], [7]. Most of these approaches have achieved competitive results but are limited to single frame input.

During autonomous driving, data are always generated in a streaming fashion and thus it is more natural to perform object detection with streaming data. Compared to single frame, streaming data can provide consistent temporal correlations between consecutive frames for detected features, which can reduce detection noise over time. In addition, truncated and occluded targets can possibly be compensated by subsequent frames within streaming data. Therefore, exploring 3D object detection methods specifically for streaming data is essential and promising.

Performing 3D object detection in streaming data is however complex. First of all, acquiring consistent 3D information between frames is difficult. On the one hand, camera data provide rich appearance features but lack of depth information. On the other hand, though LiDAR can accurately detect the position of object of interest, it is very sparse and thus difficult to determine the appearance of object purely from its point cloud representation. Second, how to correlate the features between individual frames is not obvious. For example, generating 3D scene flow with temporal feature representation will need to determine the corresponding points between frames, which is not straightforward and challenging. Last but not least, the sheer numbers of frames that streaming data provided introduce unaffordable computational costs for frame level detection.

This paper proposes a **Dual-way Object Detection and Tracking network (DODT)** to tackle the aforementioned problem. The intuition of our approach is that temporal features can be obtained through computing convolutional cross-correlation between adjacent frames, which has been demonstrated in [8]. **DODT** consists of two detection branches, a *shared RPN module* and a *Tracking module*. Its structure is illustrated in *Figure 1*. Detection branches aggregate image features to compensate sparse point cloud data, thus can utilize the strengths if both. RPN module is feed with streaming data segment between two keyframes, and output 3D proposals shared by two detection branches. While *Tracking module* utilizes correlation operation to encode temporal features and predict object displacements over frames, avoiding estimating 3D scene flow directly. Notably, different from [8], [9], our correlation is performed on proposal-level supported by sharing mechanism in RPN, which leads to a much lower computational cost. Note that the network is only performed on keyframes, thus we develop a motion based interpolation algorithm to propagate predictions to non-keyframes. Meanwhile, multi-object tracking can also be accomplished through *tracking by detection* [10]. Moreover, our interpolation algorithm shows a well solution for some common tracking problems such as drift, loss of targets in one frame, etc.

In summary, our contributions are threefold: (i) We set up a dual-way network for 3D streaming-based object detection and multi-object tracking in autonomous driving scenarios. (ii) We introduce a *Tracking module* for temporal object cross-correlation representation, which is much efficient than previous approaches. (iii) We develop a motion based interpolation algorithm for streaming level detection and multi-object tracking, which leads to considerable improvements on both tasks.

## II. RELATED WORK

**3D object detection.** Currently, most approaches in 3D object detection can be divided into three types: image based detectors, point cloud based detectors, and fusion based detectors. Image based approaches such as Mono3D [1] and 3DOP [2] use camera data only. Since image lacks depth information, hand-crafted geometric features are required in these approaches. Point cloud based methods are usually done in two fronts: voxelization based and projection based, according to how point clouds features are represented. Voxelization based methods such as 3D FCN [11], Vote3Deep [12], VoxelNet [3], utilize a voxel grid to encode features. These approaches suffer from the sparsity of point clouds and enormous computation costs in 3D convolution. While

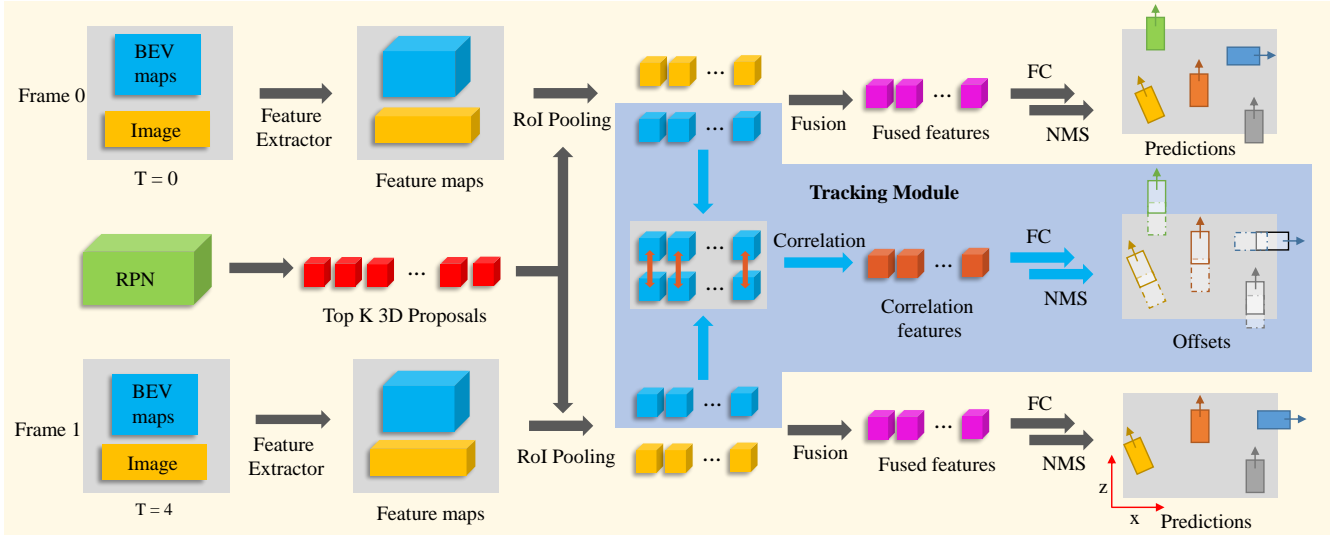


Fig. 1: DODT architecture. Blue area is *Tracking module*, it does not use image features since information in BEV is sufficient for tracking.

projection based methods such as PIXOR [4], ComplexYOLO [5], Complexer-YOLO [13] attempt to project point clouds to a perspective view (e.g. bird eye view) and apply image-based feature extraction techniques. However, due to the sparsity of point cloud, features after projection are insufficient for accurate object detection, especially for small targets. Fusion based approaches such as F-PointNet [14], MV3D [6], AVOD [7], try to fuse point clouds with images to obtain accurate object detection. These methods distinguish from each other mainly on how data is fused. Our detection submodule is constructed based on AVOD, but proposal features are enhanced with temporal information additionally.

**Video object detection.** Nearly all existing methods in video object detection incorporate temporal information on either feature level or final box level. FGFA [15] leverages temporal coherence on feature level, it warps the nearby frames feature maps to a reference frame for feature enhancement according to flow emotion. On the other hand, T-CNN [16], [17] leverages precomputed optical flows to propagate predicted bounding boxes to neighboring frames, Seq-NMS [18] improves NMS algorithm for video by constructing sequences along nearby high-confidence bounding boxes from consecutive frames. They are all utilize temporal information in final box level. There are also a few approaches attempt to learn temporal features between consecutive frames without optical flow. D&T [8] proposes a two branches detection network for object detection and tracking simultaneously in video. The network learns temporal information representation through computing convolutional cross-correlation between frames. Our DODT approach is mainly inspired by D&T, however, we develop this idea to 3D space and restrain correlation operation in proposal-level, which reduce computational costs significantly. Moreover, tracking using two adjacent frames often suffers from drift, loss of targets in one frame, etc, our approach can handle these issues well by performing motion based interpolation algorithm.

**3D multi-object tracking.** Existing 3D multi-object tracking approaches are mostly implemented based on tracking by

detection. For example, FaF [19] jointly reasons about 3D detection, tracking and motion forecasting taking a 4D tensor created from multiple consecutive temporal frames. It can aggregate the detection information for the past  $n$  timestamps to produce accurate tracklets. DSM [20] first predicts 3D bounding boxes in continuous frames and then associates detections using a *Matching net* and a *Scoring net*, which is similar to our approach. However, their 3D detector is directly single frame based approach MV3D [6], temporal features between frames are mostly ignored. Moreover, their bounding boxes association is done by solving a linear program and is an offline version, while our tracking algorithm is an near online approach.

### III. METHODOLOGY

In this section, we first give an overview of our DODT approach (Sec. A) that generates 3D object detection and tracking results given two adjacent keyframes as inputs. We then introduce the *shared RPN module* (Sec. B) that predicts 3D proposals shared by two detection branches. Sec. C shows how *Tracking module* encodes object co-occurrences and predicts the displacement of corresponding targets in two adjacent keyframes. Sec. D shows how we implement motion based interpolation algorithm to accomplish 3D streaming based object detection and multi-object tracking.

#### A. DODT Model Structure

We aim at performing 3D object detection and tracking on streaming data. To this end, we design DODT architecture into a dual-way network. *Figure 1* illustrates the whole architecture. By doubling its inputs, we can feed two adjacent keyframes data simultaneously. The keyframes data consists of an image and point cloud BEV maps (following the procedure described in MV3D [6]). Keyframe data are first fed to feature extractors to get corresponding feature maps. The feature extractor is designed following the procedure described in AVOD [7], which constructed in an encoder-decoder structure resulting in a full resolution feature map.

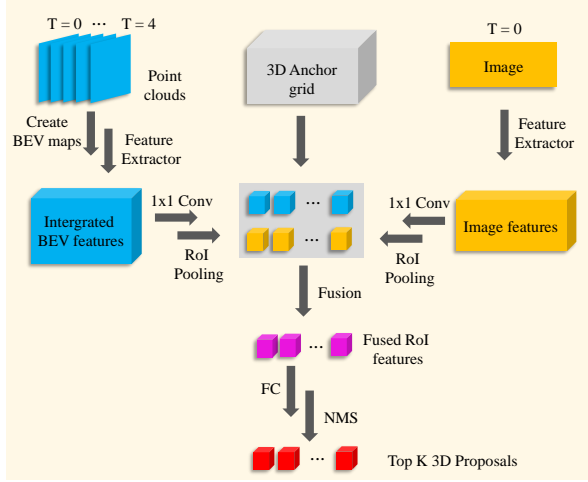


Fig. 2: RPN structure.

To extract feature crops from image and BEV view feature maps, we also follow the idea proposed in AVOD. Given an 3D anchor generated by RPN, two view specific ROIs are obtained by projecting the anchor onto the BEV and image feature maps, the corresponding feature crops are extracted and then RoI pooling is performed from two views. After that, a *early fusion* scheme is used to combine multi-view features in proposal-level. Finally, after fully connected layers and the non-maximum suppression (NMS) algorithm, the final detection outputs are produced. Meanwhile, fed with BEV feature crops of two branches, *Tracking module* performs correlation operation to feature crop pairs to produce correlation features. The correlation features are then used to predict proposal-level offsets for streaming-level detection and tracking.

The whole network is designed in an end-to-end form. The multi-task loss consists of a cross-entropy loss  $L_{cls}$  for classification, a smooth  $L1$  loss  $L_{reg}$  for coordinates regression, and a smooth  $L1$  loss  $L_{corr}$  for displacements regression between corresponding objects across two keyframes.  $L_{reg}$  and  $L_{corr}$  are normalized by the number of proposals while  $L_{cls}$  is normalized by positive proposals.

### B. Shared RPN Module

We transform RPN network in AVOD [7] to our *shared RPN network*, which can generate 3D proposals to both detection branches. The structure is shown in Figure 2. To ensure proposals generated by our RPN are suitable for both keyframes, we create integrated BEV feature maps based on frame segment between two keyframes. In this case is five consecutive frames. Note that point cloud shows object accurate 3D localization, we can simply transform five frames to a same coordinate system to fuse them. Since point cloud is extremely sparse and is encoded by projection, this process does not increase any computational cost but enhance the point cloud features. Due to object movement, the location of the same target in five frames are shift. For training convenient, we replace original labels with new axis-aligned labels that can cover all five original

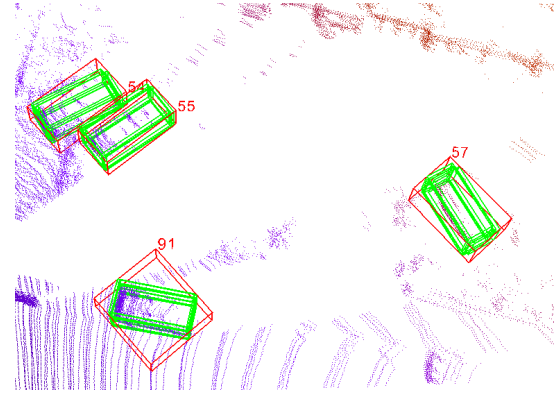


Fig. 3: Five consecutive point clouds in the same coordinate system. Green boxes are default labels of five frames, while red boxes are the axis-aligned new labels for RPN training. The numbers are object id.

labels. Figure 3 illustrates two kind of labels. Though this process enlarges proposal labels, it can ensure that new labels are suitable for proposals generation of both detection branches. To obtain accurate proposals, We fused image data to enhance object appearance features. Since one image contains enough features in a short temporal segment, our module only aggregates the features in first image of a segment. Subsequent processing and operations are similar with RPN module described in AVOD, we refer the reader to [7] for more information.

### C. Tracking Module

The *Tracking Module* is demonstrated in Figure 1. Given two sets of point cloud BEV features crops  $F_t, F_{t+\tau}$ , a set of cross frame feature pairs can be constructed as  $\{(F_t^i, F_{t+\tau}^i) \mid i \in \{0, 1, \dots, N\}\}$ , where  $F_t^i, F_{t+\tau}^i$  are features extracted by  $i$ -th proposal from frame  $t$  and frame  $t + \tau$  respectively,  $\tau$  is temporal stride and  $N$  is the number of 3D proposals. Note that the proposals generated by RPN are shared by two detection branches, thus the set can be obtained easily. After the correspondence of feature crops build, correlation operation is performed on feature pair  $(F_t^i, F_{t+\tau}^i)$  to compute correlation features over frames. Once the correlation features are obtained, FC layers are used to predict the localization and shape offsets.

For a single target we have ground truth box  $B^t = (B_x^t, B_z^t, B_l^t, B_w^t, B_{ry}^t)$  in frame  $t$ , and similarly  $B^{t+\tau}$  for frame  $t + \tau$ , denoting the horizontal and vertical center coordinates and its height, width as well as steering angle in BEV map. The tracking regression values for the target  $\Delta^{t,t+\tau} = (\Delta_x^{t,t+\tau}, \Delta_z^{t,t+\tau}, \Delta_l^{t,t+\tau}, \Delta_w^{t,t+\tau}, \Delta_{ry}^{t,t+\tau})$  are then

$$(\Delta_x^{t,t+\tau}, \Delta_z^{t,t+\tau}, \Delta_{ry}^{t,t+\tau}) = \begin{cases} (\frac{B_x^{t+\tau} - B_x^t}{B_w^t}, \frac{B_z^{t+\tau} - B_z^t}{B_l^t}, \frac{B_{ry}^{t+\tau} - B_{ry}^t}{B_{ry}^t}) & B \in F_t \cap F_{t+\tau} \\ (0.0, 0.0, 0.0) & \text{else} \end{cases} \quad (1)$$

$$(\Delta_l^{t,t+\tau}, \Delta_w^{t,t+\tau}) = \begin{cases} (0.0, 0.0) & B \in F_t \cap F_{t+\tau} \\ (1.0, 1.0) & B \notin F_t \cap B \in F_{t+\tau} \\ (-1.0, -1.0) & B \in F_t \cap B \notin F_{t+\tau} \end{cases} \quad (2)$$

If the target exists in both keyframes, the shape offsets  $(\Delta_l^{t,t+\tau}, \Delta_w^{t,t+\tau})$  are zero since 3D shape of vehicle dose

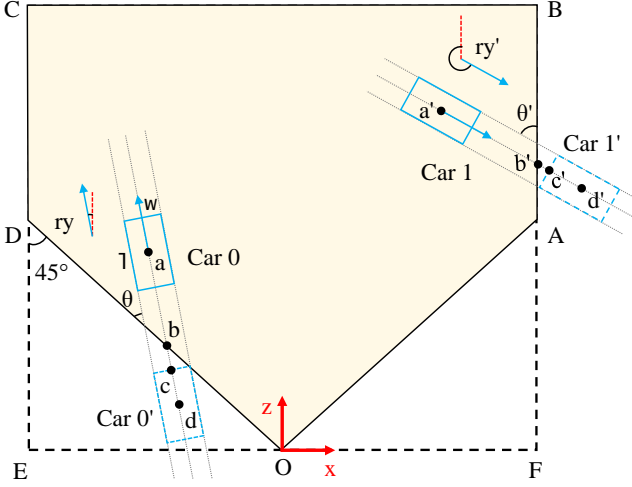


Fig. 4: Motion model of objects that in the start or end of a trajectory.

not change over time. *Tracking Module* only need to predict location offsets ( $\Delta_x^{t,t+\tau}$ ,  $\Delta_z^{t,t+\tau}$ ,  $\Delta_{ry}^{t,t+\tau}$ ). If the target exists only in one keyframe, we set location offsets to zero but shape offsets to 1 for target appearance and -1 for disappearance. The encoding of object existence is the key technology to determine a birth or death of a trajectory, more details are available in Sec. D.

#### D. 3D Streaming Object Detection and Tracking

Since streaming data possesses a lot of redundant information and objects typically move smoothly in time, we can perform detection network on keyframes and compute detections of intermediate frames by applying *Algorithm 1*. Fed with two detections lists and a offsets list, the algorithm first utilizes linear interpolation to calculate the detections in intermediate frames if an object exists on both keyframes. The co-occurrence is determined by a data association method named *IsMatched*, which performs a IoU based algorithm to select the best matched detection in second keyframe given a rectified detection ( $d_i^t + \delta_i^t$ ) in first keyframe. For a detection that exists only in one keyframe, if its absolute value of shape offsets ( $|\delta_{i,l}^{t,t+\tau}|$ ,  $|\delta_{i,w}^{t,t+\tau}|$ ) are both less than  $\delta_{max}$  (0.3 in our experiment), we assume a mis-detection in detection module, then we perform linear interpolation to compute detections. Otherwise we assume a trajectory birth or death appearance, then a motion model is leveraged to predict the detections.

The motion model is illustrated in *Figure 4*. Rectangle BCEF is the range of selected point clouds with size of  $[-40, 40] \times [0, 70]$  meters along  $X, Z$  axis respectively. While OABCD is the range contains points within the field of view of the camera. Taking *Car 0* as a example, it's a trajectory birth and is not detected in first keyframe, but is detected in second keyframe. We hold the hypothesis that *Car 0* is just outside OABCD and located in  $d$  in first keyframe and its steering angle  $ry$  is not change, thus the offset can be obtained by Equation (4). Then the detections in intermediate frames can be calculated. Trajectory death case shows in *Car*

#### Algorithm 1: Motion based Interpolation Algorithm

```

1 Input:
    $D^t = [d_0^t, d_1^t, \dots, d_{N_t}^t]$ ,  $D^{t+\tau} = [d_0^{t+\tau}, d_1^{t+\tau}, \dots, d_{N_{t+\tau}}^{t+\tau}]$ ,
    $\Delta^t = [\delta_0^{t,t+\tau}, \delta_1^{t,t+\tau}, \dots, \delta_{N_t}^{t,t+\tau}]$ 
2 Output:  $D = [D^t, D^{t+1}, \dots, D^{t+\tau}]$ 
3 Initialize:  $D_{temp} = D^{t+\tau}$ ,  $D$ ,  $\delta_{max}$ 
4 for  $d_i^t \in D^t$  do
5    $d' = IsMatched(d_i^t + \delta_i^t, D_{temp})$ 
6   if  $d'$  then
7      $d_i^{t+1}, \dots, d_i^{t+\tau-1} = Interpolate(d_i^t, d')$ 
8     remove  $d'$  from  $D_{temp}$ 
9   else if  $|\delta_{i,l}^{t,t+\tau}| < \delta_{max}$  and  $|\delta_{i,w}^{t,t+\tau}| < \delta_{max}$  then
10     $d_i^{t+1}, \dots, d_i^{t+\tau-1} = Interpolate(d_i^t, \delta_i^t)$ 
11   else
12    predict  $(d_i^{t+1}, \dots, d_i^{t+\tau-1})$  by motion model
13 if  $D_{temp}$  is not empty then
14   for  $d_j^{t+\tau} \in D_{temp}$  do
15     if  $|\delta_{j,l}^{t,t+\tau}| < \delta_{max}$  and  $|\delta_{j,w}^{t,t+\tau}| < \delta_{max}$  then
16        $d_j^{t+1}, \dots, d_j^{t+\tau-1} = Interpolate(d_j^{t+\tau}, \delta_j^{t+\tau})$ 
17     else
18       predict  $(d_j^{t+1}, \dots, d_j^{t+\tau-1})$  by motion model

```

1 can be handled in the similar way.

Multi-object tracking can be accomplished with streaming level object detection simultaneously. Note that after performing interpolation algorithm on a keyframe pairs, data association between keyframes detections is also obtained. We only need to associate keyframes detections to link tracklets in time and build long-term object tubes. This shows an near online tracking approach.

$$\begin{aligned}
 |ad| &= |ab| + |bc| + |cd| \\
 &= |ab| + \frac{w}{2 \tan(\frac{\pi}{4} - ry)} + \frac{l}{2}
 \end{aligned} \tag{3}$$

$$\{\Delta_x, \Delta_z\} = \{|ad| \sin(ry), |ad| \cos(ry)\} \tag{4}$$

## IV. EXPERIMENTS

### A. Datasets and Training

**Datasets and Preprocessing.** We use KITTI object tracking Benchmark [21] for evaluation. It consists of 21 training sequences and 29 test sequences with vehicles annotated in 3D. We split 21 training sequences into two parts according to their sequence number, odd numbered sequences for training and the rest for evaluation. For multi-object tracking evaluation, we train our model in all 21 training sequences. Similar to the data preprocessing in AVOD [7], we crop point clouds at  $[-40, 40] \times [0, 70] \times [0, 2.5]$  meters along  $X, Z, Y$  axis respectively to contain points within the field of view of the camera. To make the system invariant to the speed of the ego-car, we calculate the displacement of the observer between different frames and translate the coordinates accordingly. Location and velocity information of the ego-car are available in IMU data.

Models	Methods	IoU = 0.5			IoU = 0.7			FPS
		Easy	Moderate	Hard	Easy	Moderate	Hard	
AVOD[7]	-	90.40 / 90.91	71.71 / 72.72	71.33 / 72.72	75.24 / 90.90	55.11 / 72.69	48.58 / 72.66	12.5
DODT ( $\tau = 0$ )	-	90.13 / 90.91	80.00 / 81.79	71.61 / 81.79	76.00 / 90.90	57.23 / 81.73	56.13 / 72.69	12.5
DODT*( $\tau = 0$ )	-	98.04 / 99.94	88.77 / 90.89	88.55 / 90.88	87.66 / 90.91	68.98 / 90.85	68.32 / 90.84	12.5
DODT*( $\tau = 1$ )	T	98.70 / 99.96	89.28 / 91.88	89.06 / 91.87	88.16 / 90.91	75.14 / 90.82	73.98 / 90.80	10
DODT*( $\tau = 1$ )	M	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	10
DODT*( $\tau = 1$ )	T + M	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	10
DODT*( $\tau = 2$ )	T + M	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00
DODT*( $\tau = 3$ )	T + M	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00
DODT*( $\tau = 4$ )	T + M	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00
DODT*( $\tau = 5$ )	T + M	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00

TABLE I: We report  $AP_{3D}/AP_{BEV}$  (in %) of the **Car** category on our KITTI tracking evaluation datasets, corresponding to average precision of the birds-eye view and 3D object detection. T is *Tracking Module*, M is *motion based interpolation algorithm*. \* means the network was first pretrained on KITTI object detection datasets, and then fine-tuned on our tracking training datasets.  $\tau$  is temporal stride.

**Training and testing.** We train our network for *Car* category temporarily, following most of the super-parameter settings in AVOD [7] during training and testing. The network is trained for 120K iterations using an ADAM [22] optimizer with an initial learning rate of 0.0001 that is decayed exponentially every 30K iterations with a decay factor of 0.8. The weights of  $L_{cls}, L_{reg}, L_{corr}$  are 1.0, 5.0, 1.0 respectively. During proposal generation, anchors with IoU less than 0.3 are considered background and greater than 0.5 are objects. To remove redundant proposals, 2D NMS is performed at an IoU threshold of 0.8 in BEV to keep the top 1024 proposals during training, while at inference time, the top 300 proposals are kept.

## B. Results

**Shared RPN.** To evaluate the performance of our *Shared RPN*, we implement a non-shared version of RPN. It predicts proposals for each keyframes independently based on each feature maps. A comparison of proposal prediction accuracy between *Non-shared RPN* and *Shared RPN* is shown in Figure II. Result showing *Shared RPN* outperforms *Non-shared RPN* by 0.66% indicates that the shared mechanism in RPN promotes the accuracy of proposal prediction.

Method	Non-shared RPN	Shared RPN
Accuracy(%)	97.81	<b>98.47</b>

TABLE II: Comparison of proposal prediction accuracy.

**3D object detection.** The main results on 3d object detection are summarized in Table I. Several important trends can be observed: **1)** compared to original AVOD [7] model trained on tracking training datasets, our DODT model (without *Tracking Module* and interpolation algorithm) shows improvements in all settings with IoU threshold 0.7. These improvements indicate that the introduction of dual-way structure and *Shared RPN* contribute to detection performance significantly. **2)** Fine-tuned model outperforms model trained from scratch by a large margin, over 10% in almost all settings. Since KITTI tracking datasets are similar with KITTI object detection datasets, transfer learning can improve model performance greatly.

**Streaming level detection.** We first evaluate the effectiveness of *Tracking Module* and our interpolation algorithm with temporal stride  $\tau = 1$ . The results are shown in Table I. It's observed that the introduction of these two modules both contribute to model performance significantly, especially in *Moderate* and *Hard* setting with a overlap of 0.7. *Tracking Module* brings 6.16% and 5.66% gain, while interpolation algorithm xx% and xx% respectively. These significant improvements indicate that

Secondly, We investigate the effect of multi-frame input during testing. Specifically, we focus on the effect of different temporal strides  $\tau$  on inference accuracy and speed. Towards this goal, we train five models with  $\tau = \{1, 2, 3, 4, 5\}$ , and then link the predicted detections over time and generate detections in intermediate frame by box interpolation. Results are shown in Table I. DODT\* ( $\tau = 3$ ) achieves the best result among five models, with xx%  $AP_{3D}$  in *easy* setting, xx%  $AP_{3D}$  in *moderate* setting, xx%  $AP_{3D}$  in *hard* setting. Compared with the based fine-tuned model DODT\* ( $\tau = 0$ ), the  $AP$  scores of DODT\* ( $\tau = 3$ ) can be boosted significantly (e.g. 3D *moderate* setting by xx%, 3D *hard* setting by xx%, BEV *moderate* setting by xx%, BEV *hard* setting by xx%). This gain demonstrates that the detection of truncated and occluded targets can benefit from a large temporal stride. However, there is also a non-ignorable decay on the *easy* setting (by -xx%), we think it is mainly caused by the failed link at both ends of the trajectories (see Sec. 3.4 for detail). Moreover, Table I shows that a too large  $\tau$  leads to a significant decay of accuracy. This is straightforward as a larger temporal stride introduces more failed trajectories link.

We calculate the inference time in streaming level. Results in Table I shows that a larger temporal stride leads to less time cost per frame. Moreover, when  $\tau$  is larger than 3, our Bi-AVOD network can run faster than origin AVOD in streaming level. We chose  $\tau = 3$  for our following experiment, which is a good trade-off between speed and accuracy.

**Multi-object tracking.** We finally validate our approach on multi-object tracking. To investigate the effect of our correlation module, we compare our approach with original



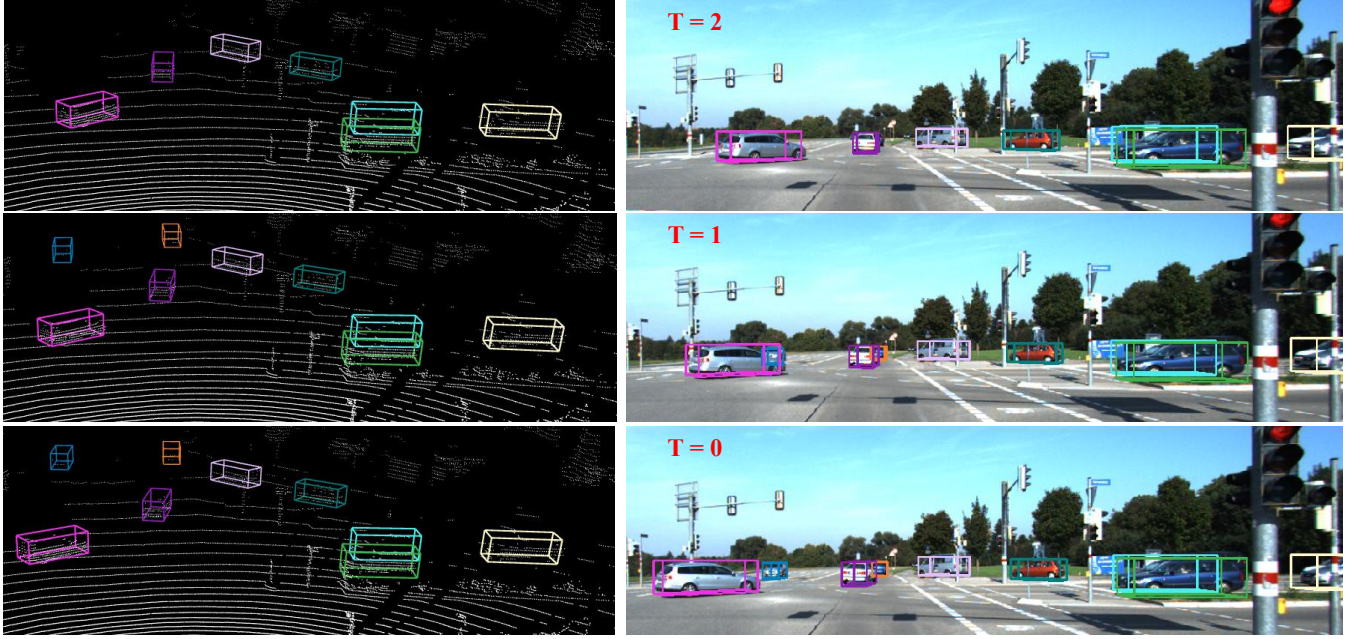


Fig. 5: Visualization of a set of trajectories produced by the tracker. Trajectories are color coded, such that having the same color means it's the same object.

Method	MOTA(%)	MOTP(%)	MT(%)	ML(%)	IDS	FRAG
AVOD[7]	58.59	81.62	42.44	31.51	5	166
DODT(ours)	<b>78.90</b>	<b>84.22</b>	<b>70.59</b>	<b>5.04</b>	31	<b>123</b>

TABLE III: Tracking performance comparison of origin AVOD and our Bi-AVOD on KITTI tracking evaluation datasets.

Method	MOTA(%)	MOTP(%)	MT(%)	ML(%)	IDS	FRAG
CEM[23]	51.94	77.11	20.00	31.54	125	396
RMOT[24]	52.42	75.18	21.69	31.85	50	376
TBD[25]	55.07	78.35	20.46	32.62	31	529
mbodSSP[26]	56.03	77.52	23.23	27.23	0	699
SCEA[27]	57.03	78.84	26.92	26.62	17	461
SSP[26]	57.85	77.64	29.38	24.31	7	704
ODAMOT[28]	59.23	75.45	27.08	15.54	389	1274
NOMT-HM[29]	61.17	78.65	33.85	28.00	28	<b>241</b>
LP-SSVM[30]	61.77	76.93	35.54	21.69	16	422
RMOT*[24]	65.83	75.42	40.15	9.69	209	727
NOMT[29]	66.60	78.17	41.08	25.23	13	150
DCO-X*[31]	68.11	78.85	37.54	14.15	318	959
mbodSSP*[26]	72.69	78.75	48.77	8.77	114	858
SSP*[26]	72.72	78.55	53.85	<b>8.00</b>	185	932
NOMT-HM*[29]	75.20	80.02	50.00	13.54	105	351
SCEA*[27]	75.58	79.39	53.08	11.54	104	448
MDP[32]	<b>76.59</b>	82.10	52.15	13.38	130	387
DODT(ours)	72.21	<b>82.29</b>	<b>54.61</b>	15.38	113	523

TABLE IV: Tracking performance comparison of publicly available methods in the KITTI Tracking Benchmark.

AVOD structure in our evaluation datasets. Performance comparison is shown in Table III. We see that Our Bi-AVOD approach outperforms origin AVOD by a large margin in nearly all tracking metrics (e.g. MOTA by 20.31%, MOTP by 2.6%, MT by 28.15%, ML by 26.47%, FRAG by 43). This indicates that our correlation module can improve the performance of multi-object tracking significantly. We also compare our approach to publicly available methods in KITTI Tracking Benchmark. In Table IV we see that our approach is competitive with the state of the art, outperforming other methods in some of the metrics (MOTP and MT).

Note that KITTI only evaluates the metrics in 2D, which does not fully represent the performance of our 3D approach. We also visualize some trajectories produced by our tracker. An example is shown in Figure 5. It shows that our approach can generate nice trajectories for most targets, even though those truncated and occluded targets. More examples are available in the supplementary materials.

## V. CONCLUSIONS

We propose Bi-AVOD, a unified framework for simultaneous 3D object detection and tracking in streaming data. The network is a dual-way structure and can process two frames at the same time. Embedded with a correlation module to encode the diversity of adjacent frames, our network can perform object detection and tracking in a very efficient way. Our approach achieves accuracy competitive with the state-of-the-art methods in KITTI Tracking Benchmark. In the future, we plan to improve our approach with a more flexible key frame selection algorithm and explore the mismatch problem of trajectory boundaries.

## REFERENCES

- [1] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2147–2156.
- [2] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals using stereo imagery for accurate object class detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1259–1272, 2018.
- [3] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.
- [4] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

- [5] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *European Conference on Computer Vision*. Springer, 2018, pp. 197–209.
- [6] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.
- [7] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.
- [8] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3038–3046.
- [9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.
- [10] P. Lenz, A. Geiger, and R. Urtasun, "Followme: Efficient online min-cost flow tracking with bounded memory and computation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4364–4372.
- [11] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1513–1518.
- [12] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1355–1361.
- [13] M. Simon, K. Amende, A. Kraus, J. Honer, T. Samann, H. Kaulbersch, S. Milz, and H. Michael Gross, "Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- [14] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.
- [15] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 408–417.
- [16] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang *et al.*, "T-cnn: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2018.
- [17] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 817–825.
- [18] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-nms for video object detection," *arXiv preprint arXiv:1602.08465*, 2016.
- [19] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3569–3577.
- [20] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3d tracking by detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 635–642.
- [21] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [23] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE TPAMI*, vol. 36, no. 1, pp. 58–72, 2014.
- [24] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015.
- [25] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *Pattern Analysis and Machine Intelligence (PAMI)*, 2014.
- [26] P. Lenz, A. Geiger, and R. Urtasun, "Followme: Efficient online min-cost flow tracking with bounded memory and computation," in *International Conference on Computer Vision (ICCV)*, 2015.
- [27] J. H. Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, "Online multi-object tracking via structural constraint event aggregation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [28] A. Gaidon and E. Vig, "Online Domain Adaptation for Multi-Object Tracking," in *British Machine Vision Conference (BMVC)*, 2015.
- [29] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," *ICCV*, 2015.
- [30] S. Wang and C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *International Journal of Computer Vision*, 2016.
- [31] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *CVPR*, 2013.
- [32] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.