3D Object Detection and Tracking on Streaming Data

BMVC 2019 Submission # 536

Abstract

Recent approaches for 3D object detection have made tremendous progress due to the development of neural networks. 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 multiple frames. Towards this goal, we set up a ConvNet architecture that can associate multi-frame images and multi-frame point clouds to generate accurate 3D detections and trajectories in an end-to-end form. Specifically, a correlation module is introduced to capture objects co-occurrences across time, and a multi-task objective for frame-based object detection and across-frame track regression are used. 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.

1 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[3, 5], point clouds based[32, 35, 38], and multi-view fusion based[4, 20]. 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 from streaming data. Compared to single-frame based approaches, 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 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 provide introduces unaffordable computational costs for frame level detection.

In this paper, we propose a dual-way network named Bi-AVOD to tackle the aforementioned problem. Our network is based on an aggregate view object detection architecture AVOD [20] and the structure is illustrated in Figure 1. The network takes two adjacent frames (i.e. an image combined with its corresponding point cloud) as inputs and is capable of fusing different features in image and point cloud, thus utilizing the strengths of both. In order to avoid estimating 3D scene flow directly, a correlation module is aided to our Bi-AVOD for temporal feature encoding. The correlation module computes convolutional cross-correlation between the features response of adjacent frames to estimate local displacements. Moreover, for a fast inference speed, we perform 3D object detection on key frames and propagate predicted bounding boxes to neighboring frames for a streaming-based detection. Furthermore, by linking detections over time with local displacement information, multi-object tracking can be performed through *tracking by detection* [22].

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)* Instead of encoding temporal feature using 3D scene flow, we introduce a correlation module to compute convolutional cross-correlation of adjacent frames for temporal feature representation. *(iii)* We perform our approach to KITTI Object Tracking Benchmark and obtain competitive results, with 72.21% in MOTA and 82.29% in MOTP respectively.

2 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. Images based approaches such as Mono3D [3] and 3DOP [5] 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 information is represented. Voxelization based methods utilize a voxel grid representation to encoder the point cloud and then apply a 3D CNN for feature extraction. These approaches include 3D FCN [23], Vote3Deep [9], VoxelNet [38], *et al*. These approaches suffer from the sparsity of point cloud and enormous computation costs in 3D convolution. While projection based methods attempt to project point cloud to a perspective view (e.g. bird eye view) and apply image-based feature extraction techniques, such as PIXOR [35], FaF [26], Comple-YOLO [32], *et al*. These methods take advantage of the fact that 3D detections in driving scenes are almost on the same plane, thus loss of height information has little influence on performance. 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 [29] first extracts the 3D bounding frustum of an object by extruding 2D bounding boxes from image detectors, then consecutively performs 3D object instance segmentation and amodal extent regression to estimate the amodal 3D bounding box. This method works well for indoor scenes and brightly lit outdoor scenes, but are expected to perform poorly in more extreme outdoor scenarios. MV3D [4] extends the image based RPN of Faster R-CNN[31] to 3D and proposes a 3D RPN, then applies feature fusion of images and point clouds to produces accurate 3D detections. However, due to the insufficient information in feature extraction caused by downsampling, it does not

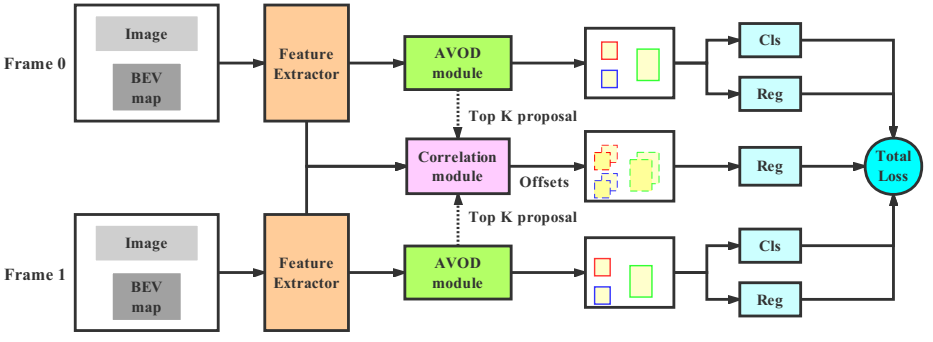

Figure 1: Bi-AVOD architecture. For convenience, AVOD module here does not include feature extractor and loss function.

work well for small targets. AVOD [20] is similar to MV3D in 3D RPN and feature fusion, but with full resolution feature maps produced by a pyramid architecture, which leads to a great improvement in localization accuracy for small targets. In our proposed approach, the AVOD framework is used as a basic 3D object detection module.

Video object detection. Nearly all existing methods in video object detection incorporate temporal information on either feature level or final box level. FGFA [39] leverages temporal coherence on feature level. The network first applies feature extraction network on individual frames to produce per-frame feature maps, and then enhances features at a reference frame by warping the nearby frames feature maps according to flow emotion. On the other hand, final box level approaches usually utilize temporal information in bounding box post processing. T-CNN [17, 18] leverages precomputed optical flows to propagate predicted bounding boxes to neighboring frames, and then generates tubelets by applying tracking algorithms from high-confidence bounding boxes. Seq-NMS [16] improves NMS algorithm for video by constructing sequences along nearby high-confidence bounding boxes from consecutive frames. While boxes of the sequence are then re-scored to the average confidence and other boxes close to this sequence are suppressed.

Other approaches attempt to learn temporal information between consecutive frames to avoid using optical flow. D&T [10] presents an end-to-end fully convolutional architecture, it uses a detection and tracking based loss for simultaneous detection and tracking in video. In order to learn temporal information representation, the network is fed with multiple frames, and a correlation module is embedded for computing convolutional cross-correlation between frames. Our Bi-AVOD approach is mainly inspired by D&T.

3D multi-object tracking. Existing 3D multi-object tracking methods are mostly implemented based on tracking by detection. For example, FaF [26] 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 [11] 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[4], temporal features between frames are mostly ignored. Moreover, their bounding boxes association is done by solving a linear program, while ours by applying a improved IOU tracker algorithm based on [4].

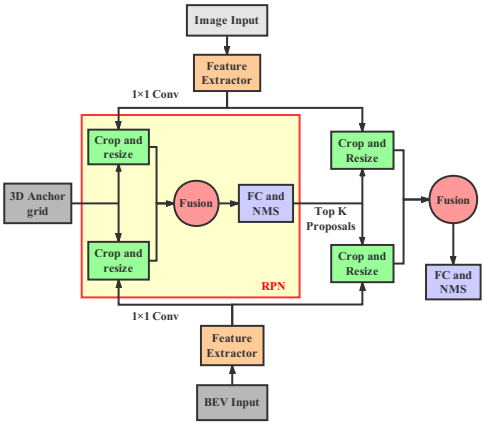

Figure 2: AVOD architecture

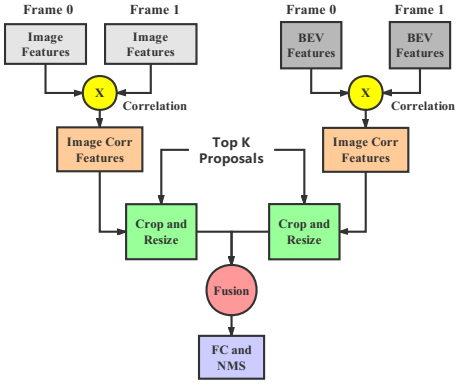

Figure 3: Correlation module

3 Methodology

In this section we first give an overview of our Bi-AVOD approach (Sec. 3.1) that generates object detection results and tracklets given two adjacent key frames as inputs. We then introduce the correlation module (Sec. 3.2) that predicts the displacement of corresponding targets in two adjacent key frames. Sec. 3.3 shows the multi-task objective function. Sec. 3.4 shows how we implement 3D streaming object detection and multi-object tracking with network outputs.

3.1 Bi-AVOD model structure

We aim at performing 3D object detection and tracking on streaming data. To this end, we transform AVOD[20] architecture to a dual-way network. Figure 1 illustrates our Bi-AVOD architecture. By doubling its input, we can feed two adjacent key frames (image and point cloud) simultaneously and obtain corresponding object detection results. Meanwhile, the local displacements can be estimated by computing convolutional cross-correlation between the features responses of adjacent frames using correlation module. With predicted detections of two key frames and their local displacements, interpolation algorithms can be performed to propagate detections to neighboring frames. Also, 3D multi-object tracking can be implemented by applying box association algorithm. Note that two *Feature Extractors* and two *AVOD modules* in Figure 1 are parameter sharing, thus the increased computational costs only come from the correlation module. We will simply review the AVOD architecture in this section, and the correlation module will be left to next section.

AVOD[20] architecture is proposed for 3D object detection in autonomous driving by aggregating front view image and BEV feature maps generated by LiDAR data. It is a two-stage method, the architecture is illustrated in Figure 2. First, pyramid based feature extractors are used to generate full resolution feature maps of both BEV map and RGB image inputs; both feature maps are then fed to a fusion RPN to generate *Top K* non-oriented region proposals after applied a 1×1 convolution for dimensionality reduction. Note that the default anchors in images and BEV feature maps are generated through 3D anchor grid projection, and ROI Pooling is implemented by *Crop and Resize* operations. Finally, the

Top K proposals are passed to the detection network for dimension refinement, orientation estimation and category classification. We refer the reader to [20] for further understanding of the architecture.

3.2 Correlation module

Our 3D correlation module is illustrated in Figure 3. Given feature maps of two adjacent key frames, it first performs correlation operation to compute convolutional cross-correlation. Similar to RPN sub-module in AVOD, the module then extracts feature crops via multiview *Crop and Resize* operations guided by aforementioned *Top K* non-oriented region proposals. The feature crops are then fed to a fusion module to generate multiview aggregated features. The fusion module is identical to the one involved in AVOD, which was first introduced in MV3D[4]. The aggregated features are then passed to a fully connected layer for regression. After that, non-maximum suppression (NMS) is applied to ignore redundant, overlapping bounding boxes for the final loss calculation.

Similar to FlowNet [8], we restrict correlation operation to a local neighborhood instead of all possible circular shifts in a feature map. This restriction helps the module avoid large output dimensionality. The correlation operation performs point-wise feature comparison of two feature maps f_t, $f_{t+\tau}$ by

$$\mathcal{C}^{t,t+\tau}(i,j,p,q) = \Big\langle f_t(i,j), f_{t+\tau}(i+p, j+q) \Big\rangle \tag{1}$$

where $p,q \in [-d,d]$ are offsets to compared features in a local square window defined by the maximum displacement d, and i,j are the location of the window center in a feature map. The output is a correlation feature map of size $\mathcal{C} \in \mathbb{R}^{h\times w\times(2d+1)\times(2d+1)}$, where h,w are the height and the width of the feature map.

After applying (1), two correlation feature maps are obtained, one for point cloud $\mathcal{C}_{pc}^{t,t+\tau}$, and one for RGB image $\mathcal{C}_{img}^{t,t+\tau}$. ROI pooling and feature fusion are then performed to produce aggregate feature map $\mathcal{C}_{fusion}^{t,t+\tau} = fusion(\mathcal{C}_{pc}^{t,t+\tau}, \mathcal{C}_{img}^{t,t+\tau})$, which is then flattened and fed to a fully connected layer to predict the transformation $\Delta^{t,t+\tau} = (\Delta_x^{t,t+\tau}, \Delta_y^{t,t+\tau}, \Delta_z^{t,t+\tau}, \Delta_r^{t,t+\tau})$ of the RoIs from t to $t+\tau$. We hold the prior that the size of a target does not change over time, thus the network only needs to predict the variations in the center coordinates and steering angle.

3.3 Multitask detection and correlation objective

We extend the multi-task loss of object detection, consisting of a classification loss L_{cls} and a regression loss L_{reg}, with an additional term L_{corr} that scores the displacement regression between corresponding objects across two frames. Considering a batch of N RoIs after category balanced sampling, the network predicts softmax probabilities $\{p_i\}_{i=1}^{N}$, bounding box regression offsets $\{b_i\}_{i=1}^{N}$, and cross-frame displacement regression $\{\Delta_i^{t+\tau}\}_{i=1}^{N_{corr}}$, the overall objective function is shown as:

$$\begin{aligned} L(\{p_i\},\{b_i\},\{\Delta_i\}) = \frac{1}{N}\sum_{i=1}^{N} L_{cls}(p_{i,c^*}) + \frac{\alpha}{N_{fg}}\sum_{i=1}^{N}[c_i^* > 0]L_{reg}(b_i, b_i^*) \\ + \frac{\beta}{N_{corr}}\sum_{i=1}^{N_{corr}} L_{corr}(\Delta_i^{t+\tau}, \Delta_i^{*,t+\tau}) \end{aligned} \tag{2}$$

where c_i^* is the ground truth class label of an RoI and p_{i,c_i^*} is corresponding predicted softmax score. b_i^* is the ground truth of bounding box regression target, and $\Delta_i^{*,t+\tau}$ is the ground truth of displacement regression target. The indicator function $[c_i^* > 0]$ is 1 for foreground RoIs and 0 for background RoIs. L_{cls} is cross-entropy loss, and L_{reg}, L_{corr} are smooth L1 loss [15]. α and β are weight coefficients for L_{reg} and L_{corr}, in this work we set 5 and 1 respectively. Note that we only consider the N_{fg} foreground RoIs loss for L_{reg}, and L_{corr} is active only for foreground RoIs which have a track correspondence in both frames. Additionally, b_i^* and $\Delta_i^{*,t+\tau}$ are all encoded following [20].

3.4 3D streaming object detection and tracking

Given a sequence of N frames $\{I_f \mid f = 1,...,N\}$, streaming-based object detection task needs to predict a set of detections D_f for each frame I_f, where $D_f = \{d_f^i \mid i = 1,...N_f\}$, d_f^i is i^{th} target and N_f is the number of targets in frame f. Note that D_f can also be an empty set when no target is detected in a frame. In 3D object detection, each detection d_i is parametrized as $D_i = (x_i, y_i, z_i, w_i, h_i, l_i, \theta_i, s_i)$, where (x_i, y_i, z_i) corresponds to the coordinate of target center, (w_i, h_i, l_i) corresponds to width, height, length of the target, θ_i the rotation angle in yaw axis and s_i prediction confidence of the target.

We only perform object detection in key frames due to the redundant features in streaming data. Detections of the intermediate frame can be determined leveraging the detection results of its adjacent key frames. Suppose we have two predicted detections set $(D_f, D_{f+\tau})$ of two consecutive key frames $(I_f, I_{f+\tau})$, where τ is temporal stride, the detection result d_{f+t}^i of intermediate frame $I_{f+t}(t \in \{1, \tau-1\})$ can be obtained by

$$d_{f+t}^i = \mathcal{F}(W_f d_f^i, W_{f+\tau} d_{f+\tau}^i) \tag{3}$$

where $W_f, W_{f+\tau}$ are the corresponding weight coefficients and $\mathcal{F}$ is interpolation function. Note that we can use Equation (3) to generate d_{f+t}^i only when the target exists in D_f and $D_{f+\tau}$ simultaneously. If a target is the start or end of a trajectory in intermediate frames, this method will fail. Although carefully selecting key frames could fix this problem, it is beyond the scope of this work. In this paper we focus on the targets that always exist between two key frames.

For multi-object tracking, we attempt to assign each detection in each frame to a unique trajectory $T_k = \{d_{f_1}^k, d_{f_2}^k, ..., d_{f_{N_k}}^k\}$, utilizing an improved IOU tracker algorithm[2] (detailed description is available in supplementary material). Where k is the trajectory id and N_k is the length of T_k. Unlike multi-object tracking in 2D image which suffers from boxes overlap, detection in 3D has its unique position, any overlap of two detections in 3D means high probability of the same target. Thus a IOU based data association algorithm can also work well in our approach.

4 Experiments

4.1 Datasets and Training

Datasets. We use KITTI object tracking Benchmark [13] for evaluation. It consists of 21 training sequences and 29 test sequences with vehicles annotated in 3D. Each sequence includes hundreds of point cloud frames captured by Velodyne HDL-64E rotating 3D laser

Method	Runtime (s)	Class	$AP_{3D}(\%)$ Easy	$AP_{3D}(\%)$ Moderate	$AP_{3D}(\%)$ Hard	$AP_{BEV}(\%)$ Easy	$AP_{BEV}(\%)$ Moderate	$AP_{BEV}(\%)$ Hard
AVOD[20]	0.08	Car	75.24	55.11	48.58	89.68	72.68	72.66
Bi-AVOD	0.20		81.17	54.96	53.59	90.87	72.68	72.65
Bi-AVOD*	0.20		**83.77**	58.31	57.12	**90.88**	81.71	72.68
Bi-AVOD* ($\tau = 1$)	0.10		77.07	63.43	63.04	90.86	81.76	81.75
Bi-AVOD* ($\tau = 3$)	0.07		77.89	**63.80**	**63.50**	90.83	**81.77**	**81.76**
Bi-AVOD* ($\tau = 5$)	0.04		77.26	62.30	55.79	90.80	72.69	72.67
Bi-AVOD* ($\tau = 7$)	**0.03**		74.85	54.36	53.60	81.77	72.60	72.57

Table 1: A comparison of the performance between original AVOD, our Bi-AVOD and fine-tuned Bi-AVOD* with different temporal stride τ on KITTI tracking evaluation datasets.

scanner and corresponding RGB images. We split 21 training sequences into two parts according to their sequence number, odd numbered sequences for training datasets and even numbered ones for evaluation datasets. For multi-object tracking evaluation, we train our model in all 21 training sequences.

Datasets preprocessing. Similar to the data preprocessing in [20], we crop point clouds at $[-40,40] \times [0,70] \times [0,2.5]$ meters along Y,X,Z axis respectively to contain points within the field of view of the camera. In KITTI tracking datasets, the observer is an autonomous vehicle, thus the coordinate system of consecutive frames shift due to the movement of the observer over time. Since the location and velocity information of the observer are available in IMU data, one can calculate the displacement of the observer between different frames and translate the coordinates accordingly. In this way both point clouds and objects labels are on the exact same coordinate system. Note that this transform is important to make the system invariant to the speed of the ego-car.

Training and testing. We train our network for *Car* category only. We follow most of the super-parameter settings in [20] during training and testing. The network is trained for 120K iterations using an ADAM[19] optimizer with an initial learning rate of 0.0001 that is decayed exponentially every 30K iterations with a decay factor of 0.8. During proposal generation, anchors with IoU less than 0.3 are considered background and greater than 0.5 are object. 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.

4.2 Results

3D object detection. Both streaming level detection and multi-object tracking require accurate detection results, thus the performance of the network on 3D object detection is significant. We train our Bi-AVOD on our tracking training datasets, and results are evaluated using KITTI object detection evaluation metrics. Since evaluate our model on KITTI tracking testing datasets for 3D object detection is hard, we turn to our evaluation datasets (described in Sec 4.1) instead. To explore the effectiveness of dual-way structure on object detection, we also train original AVOD model on our training datasets. The comparison results on 3D object detection are shown in Table 1. Our Bi-AVOD achieves 81.15% AP_{3D} in *easy* setting and 53.59% AP_{3D} in *hard* setting, outperforms original AVOD by 5.93% and 5.01% respectively. This gain show that the introduction of dual-way structure and correlation module contributes to 3D object detection significantly. For better performance, we train original AVOD model on KITTI object detection datasets, and then transfer relevant parameters to

Method	MOTA(%)	MOTP(%)	MT(%)	ML(%)	IDS	FRAG
AVOD[20]	58.59	81.62	42.44	31.51	**5**	166
Bi-AVOD(ours)	**78.90**	**84.22**	**70.59**	**5.04**	31	**123**

Table 2: Tracking performance comparison of origin AVOD and our Bi-AVOD on KITTI tracking evaluation datasets.

Method	MOTA(%)	MOTP(%)	MT(%)	ML(%)	IDS	FRAG
CEM[27]	51.94	77.11	20.00	31.54	125	396
RMOT[36]	52.42	75.18	21.69	31.85	50	376
TBD[14]	55.07	78.35	20.46	32.62	31	529
mbodSSP[21]	56.03	77.52	23.23	27.23	**0**	699
SCEA[37]	57.03	78.84	26.92	26.62	17	461
SSP[21]	57.85	77.64	29.38	24.31	7	704
ODAMOT[13]	59.23	75.45	27.08	15.54	389	1274
NOMT-HM[8]	61.17	78.65	33.85	28.00	28	**241**
LP-SSVM[33]	61.77	76.93	35.54	21.69	16	422
RMOT*[36]	65.83	75.42	40.15	9.69	209	727
NOMT[8]	66.60	78.17	41.08	25.23	13	150
DCO-X*[28]	68.11	78.85	37.54	14.15	318	959
mbodSSP*[21]	72.69	78.75	48.77	8.77	114	858
SSP*[21]	72.72	78.55	53.85	**8.00**	185	932
NOMT-HM*[8]	75.20	80.02	50.00	13.54	105	351
SCEA*[37]	75.58	79.39	53.08	11.54	104	448
MDP[32]	**76.59**	82.10	52.15	13.38	130	387
Bi-AVOD(ours)	72.21	**82.29**	**54.61**	15.38	113	523

Table 3: Tracking performance comparison of publicly available methods in the KITTI Tracking Benchmark.

our Bi-AVOD model for further fine-tuned learning on KITTI tracking datasets. In this way a better performance is obtained. Results are shown in Table 1, where Bi-AVOD* is the fine-tuned model. Compared to Bi-AVOD which is trained from scratch, the fine-tuned model raises performance substantially to 83.77% AP_{3D} in *easy* setting, 58.31% in *moderate* setting and 57.12% in *hard* setting.

Streaming level detection. With accurate 3D object detection results of adjacent key frames and target displacements, streaming level 3D object detection can be implemented. We investigate the effect of multi-frame input during testing. Specifically, we focus on the effect of different temporal strides τ on inference time and accuracy. Towards this goal, we train five models with $\tau = \{0,1,3,5,7\}$, and then link the predicted detections over time and generate detections in intermediate frame by box interpolation. Results are shown in Table 1. Bi-AVOD* ($\tau = 3$) achieves the best result among five models, with 77.89% AP_{3D} in *easy* setting, 63.80% AP_{3D} in *moderate* setting, 63.50% AP_{3D} in *hard* setting. Compared with the based fine-tuned model Bi-AVOD* ($\tau = 0$), the AP scores of Bi-AVOD* ($\tau = 3$) can be boosted significantly (e.g. 3D *moderate* setting by 5.49%, 3D *hard* setting by 6.38%, BEV *moderate* setting by 9.09%, BEV *hard* setting by 9.11%). 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 -5.88%), we think it is mainly caused by the failed link at both ends of the trajectories (see Sec. 3.4 for detail). Moreover, Table 1 shows that a too large τ leads to a significant decay of accuracy. This is straightforward as a larger temporal stride introduces more failed trajectories link.

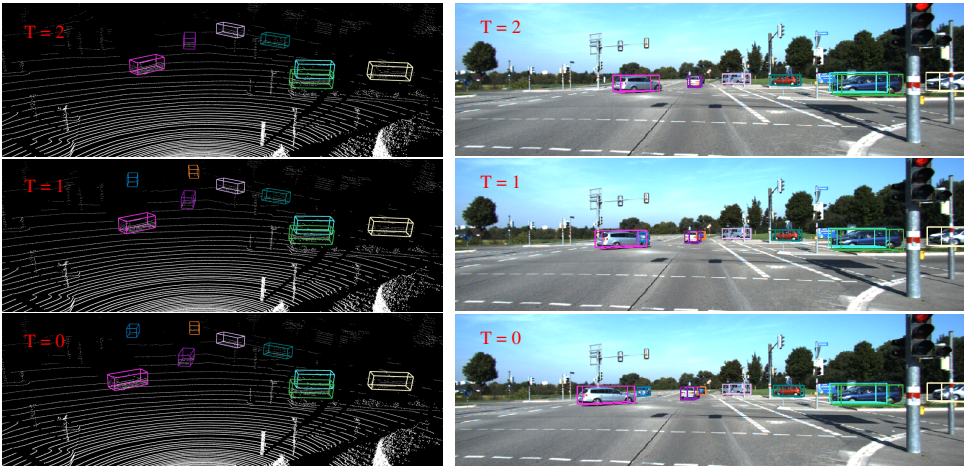

Figure 4: Visualization of a set of trajectories produced by the tracker. Trajectories are color coded, such that having the same color means it's the same object.

We calculate the inference time in streaming level. Results in Table 1 shows that a larger temporal stride leads to less time cost per frame. Moreover, when τ 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 AVOD structure in our evaluation datasets. Performance comparison is shown in Table 2. 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 3 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. A example is shown in Figure 4. 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.

5 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] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for 3d scene flow estimation from point clouds. *arXiv preprint arXiv:1806.02170*, 2018.

[2] Erik Bochinski, Tobias Senst, and Thomas Sikora. Extending iou based multi-object tracking by visual information. *AVSS. IEEE*, 2018.

[3] 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)*, pages 2147–2156, June 2016. doi: 10.1109/CVPR.2016.236.

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[5] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1259–1272, 2018.

[6] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. *ICCV*, 2015.

[7] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[9] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017.

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3038–3046, 2017.

[11] Davi Frossard and Raquel Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642. IEEE, 2018.

[12] A. Gaidon and E. Vig. Online Domain Adaptation for Multi-Object Tracking. In *British Machine Vision Conference (BMVC)*, 2015.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[14] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[16] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016.

[17] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 817–825, 2016.

[18] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2018.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[21] Philip Lenz, Andreas Geiger, and Raquel Urtasun. Followme: Efficient online min-cost flow tracking with bounded memory and computation. In *International Conference on Computer Vision (ICCV)*, 2015.

[22] Philip Lenz, Andreas Geiger, and Raquel Urtasun. Followme: Efficient online min-cost flow tracking with bounded memory and computation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4364–4372, 2015.

[23] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[25] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Learning scene flow in 3d point clouds. *arXiv preprint arXiv:1806.01411*, 2018.

[26] Wenjie Luo, Bin Yang, and Raquel 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*, pages 3569–3577, 2018.

[27] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE TPAMI*, 36(1):58–72, 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.103.

[28] Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *CVPR*, 2013.

[29] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[32] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *European Conference on Computer Vision*, pages 197–209. Springer, 2018.

[33] S. Wang and C. Fowlkes. Learning optimal parameters for multi-target tracking with contextual interactions. *International Journal of Computer Vision*, 2016. ISSN 1573-1405. doi: 10.1007/s11263-016-0960-z.

[34] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[35] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[36] Ju Hong Yoon, Ming-Hsuan Yang, Jongwoo Lim, and Kuk-Jin Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[37] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Online multi-object tracking via structural constraint event aggregation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[38] Yin Zhou and Oncel 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*, pages 4490–4499, 2018.

[39] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017.

[40] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. Towards high performance video object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7210–7218, 2018.