3D Object Detection and Tracking in Streaming Data

BMVC 2019 Submission # ??

Abstract

Recent approaches for 3D object detection have made great progresses due to the evolution in deep learning. However, previous works are mostly based on single frame point cloud or image, information between frames is scarcely explored. In this paper, we try to leverage the temporal information in streaming data and explore 3D object detection and tracking based on multi-frame. Towards this goal, we set up a ConvNet architecture that can associate multi-frame image and LiDAR data to produce accurate 3D detection boxes and trajectories in an end-to-end form. Specially, 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 is used. Therefore, the network can perform 3D object detection and tracking simultaneously. Our proposed architecture is shown to produce competitive results on the KITTI Object tracking datasets. Code and models will be available soon.

1 Introduction

Object detection in 2D images has made tremendous progress due to the emergence of deep learning [13, 15, 24] and their region based descendants [6, 12, 22] recently. However, extending 2D approaches to 3D scenes is extremely hard because of the increased computational cost and the difficulties in acquiring 3D data. Even though, many work has been carried out inspired by the rapid development of autonomous driving industry.

Recent approaches in 3D object detection are usually done in three fronts: image based, point cloud based and fusion of image and point cloud. These methods have achieved decent performance but are limited to single frame input. Note that streaming data is more natural and straightforward for most driving scenarios, thus fast and accurate 3D object detection in streaming data is crucial for autonomous driving. However, applying existing single frame based approaches directly to streaming data will inevitably loss the consistency and diversity between frames, and also introduce unaffordable computational cost and time for most applications. Therefore, exploring 3D object detection network specifically for streaming data is valuable and important.

Similar to extend 2D object detection methods to 3D scenes, we can also try to extend video-based object detection approaches to streaming-based 3D object detection. Most modern computer vision approaches to video object detection based on image and require flow estimation, for example, a series work in [27, 28] associates vision feature and optical flow to build an accurate end-to-end learning framework for video object detection. However, performing video-based approach for 3D streaming object detection in autonomous driving will

be awful, since video suffers from motion blur and partial occlusion, which are conventional in driving scenarios. While another choice is to use LiDAR data, since point cloud provides an accurate spatial representation of the world allowing for precise positioning of objects of interest, thus motion blur and occlusion problem can be avoided in three-dimensional perspective. However, LiDAR does not capture appearance well when compared to the richness of images, additionally, accurate 3D scene flow estimation from point cloud is tremendously tough. Although some approaches such as [1, 19] have been presented to learn 3D motion field in the world, their performance are limited for large and complex scenes.

Inspired by [9], we transform AVOD [16] structure into a dual-way network embedding with a correlation module, named Bi-AVOD, which takes two adjacent key frames as input and predicts location and orientation of object as well as their local displacement. Note that Bi-AVOD is an aggregate view object detection architecture capable of fusing different features in image and point cloud, thus its input includes two adjacent images in front view and two adjacent BEV (bird eye view) from LIDAR data. While the correlation module compute convolutional cross-correlation between the features response of adjacent key frames to estimate displacement of the same objects. With local displacement and object orientation, object location in intermediate frames can be calculated by interpolation. Moreover, object detection boxes can be linked between frames with the help of local displacement, then multiple object tracking can be performed through *tracking by detection* [17].

In summary, our contributions are threefold: *(i)* we set up a two stream architecture based on AVOD for simultaneous 3D streaming object detection and tracking; *(ii)* we introduce correlation module to capture object co-occurrences across time and perform frame-level 3D object detection in a high speed through interpolation; *(iii)* we utilize tracking result to improve detection performance and preliminary explore the scheme of key frame selection.

2 Related Work

2.1 Video object detection

Video object detection in image has received increased attention since ImageNet VID datasets introduced. Most approaches for video object detection utilize optical flows, which present temporal information in videos. Some representative work such as FGFA [27], it leverages temporal coherence on feature level, and improves the pre-frame features by aggregation of nearby features along the motion paths with the help of optical flows. Later a more efficient approach based on [27] has been presented in [28], it introduces three new techniques: sparsely recursive feature aggregation, spatially-adaptive partial feature updating and temporally-adaptive key frame scheduling, which make this unified approach faster, more accurate and more flexible.

There are also some approach try to learn temporal information between consecutive frames. D&T [9] set up a ConvNet architecture for simultaneous detection and tracking in video. In order to capture cross-occurrences across time, it aid a correlation operation in networks. Our Bi-AVOD architecture mainly inspired by this work.

2.2 3D object detection

Currently, most approaches in 3D object detection can be divided into three types: image based detectors, LiDAR based detectors and fusion based detectors.Image based approaches

such as Mono3D[3], 3DOP[5] use camera data only, since image has limited depth information, specific hand-crafted geometric features are required. LiDAR based methods are usually done in two fronts, one is utilizing a voxel grid representation to encoder point cloud and applying 3D CNN for features extracted, these approaches including 3D FCN [13], Vote3Deep [8] and VoxelNet [26] *et al*., these approaches suffer from the sparsity of point cloud and enormous computation cost in 3D convolution; others LIDAR based methods try to project point cloud to bird eye view (BEV) and apply 2D CNN for object detection, such as PIXOR[25], FaF[20] and Comple-YOLO [23] *et al*. These methods take advantage of the fact that objects in autonomous driving almost on the same plane thus loss of height information has little affect to the result, while the depth and Geometric information can be retained and computational complexity reduced significantly, making real-time detection possible. However, due to the sparsity of point cloud, the feature information after projecting is insufficient for accurate object detection especially for the small target.

There are also many multi-modal fusion methods that combine images and LiDAR data to improve detection accuracy. F-PointNet [21] first extracts the 3D bounding frustum of an object by extruding 2D bounding boxes from image detectors, then consecutively perform 3D object instance segmentation and amodal extent regression to estimate the amodal 3D bounding box. MV3D[4] extends the image based RPN of Faster R-CNN[22] to 3D and proposes a 3D RPN targeted at autonomous driving scenarios. MV3D uses every pixel in BEV feature map to multiple 3D anchors and then feeds the anchor to RPN to generate 3D proposals that are used to create view-specific feature crops from BEV feature maps and images. A deep fusion scheme is used to combine information from these feature crops to produce final detection output. However, MV3D does not work well for small targets due to the insufficient data for feature extracting caused by downsampling in convolutional feature extractors. AVOD[16] architecture is similar to MV3D in 3D RPN and feature fusion, however, its feature extract provides full resolution feature maps thus show greatly help in localization accuracy for small targets during the second stage of the detection framework. Our proposed architecture mostly based on AVOD mention above.

2.3 3D object tracking

More and more work has been done in 3D object tracking based on tracking by detection due to the rapidly development in 3D object detection. These approaches usually trend to apply 3D object detection and tracking simultaneously. FaF [20] jointly reasons about 3D detection, tracking and motion forecasting taking a 4D tensor created from multiple consecutive temporal frames. The most similar approach to our work is [10], however, their 3D detector is based on MV3D while ours is AVOD, and their detections association is done by solving a linear program after passing to a matching net and scoring net, while ours use a extending IOU based algorithm [2] by leveraging corresponding displacements over time.

3 Methodology

In this section we first give an overview of the Bi-AVOD approach (Sec. 3.1) that generates detections and tracklets given two adjacent key frames as input. We then introduce the correlation module (Sec. 3.2) that aiding the network in the tracking process. Sec. 3.3 shows the multi-task objective function and Sec. 3.4 shows how we implement 3D streaming detection and tracking using prediction results of Bi-AVOD.

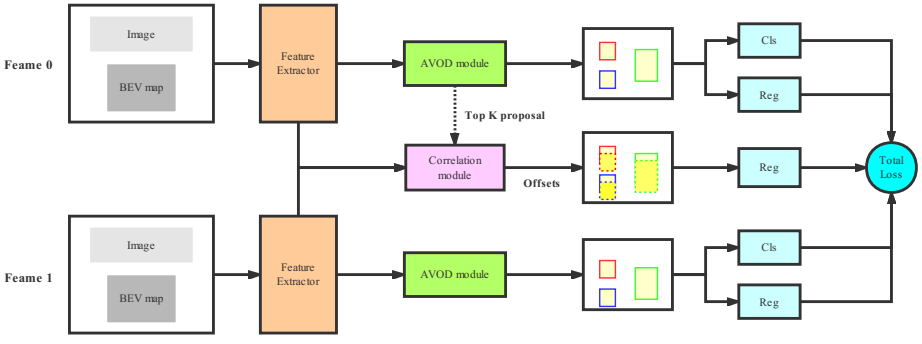

Figure 1: Bi-AVOD architecture

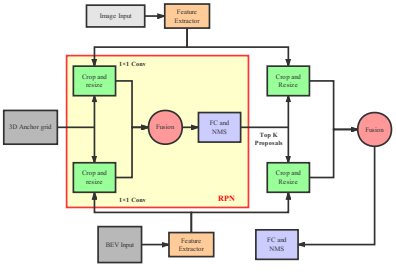

Figure 2: AVOD architecture

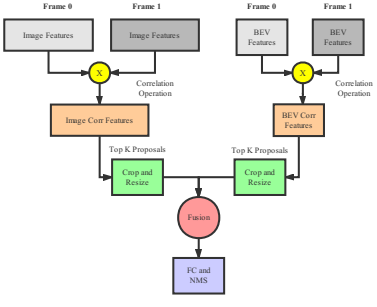

Figure 3: Correlation module

3.1 Bi-AVOD model structure

We aim at performing 3D object detection in point cloud flow, and later apply multi-object tracking using tracking by detection paradigm. We build our Bi-AVOD mainly based on AVOD[16], Figure.1 illustrates our Bi-AVOD architecture. By doubling its input field, we can feed two adjacent key frames simultaneously and obtained corresponding object detection results. Meanwhile, the local displacements can be estimated by computing convolutional cross-correlation between the feature responses of adjacent frames using correlation module. With the predicted bounding boxes of two adjacent key frames and their local displacements, we can implement interpolation algorithms to generate interval bounding boxes for 3D flow detection and develop data association algorithms for 3D object tracking. Note that two *Feature Extractor* and two *AVOD module* in Figure.1 are parameter sharing, thus the increased computational cost comes only from the correlation module. We will simply review the AVOD architecture in this section, and the correlation module will leave to the next section.

AVOD architecture is proposed for 3D object detection in autonomous driving by aggregating front view image and bird eye view (BEV) feature maps (generated by LiDAR data). It is a two stage method and Figure.2 illustrates its architecture. Firstly, it uses feature pyramids based extractors to generate full resolution feature maps from both BEV map and

RGB image. Both feature maps are then feed to fusion RPN to generate *Top K* non-oriented region proposals after applied a 1×1 convolutional layer for dimensionality reduction, note that the default anchors in image and BEV map 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 [16] for an further explanation of the architecture.

3.2 Correlation module

The correlation module is illustrated in Figure.3. It first performs correlation on two image features and two BEV map features respectively. Then similar to the above mentioned RPN, it extracts feature crops via multiview *Crop and Resize* operations guided by *Top K* non-oriented region proposals. The feature crops then feed to fusion module and multiview aggregated features are generated. While the fusion module is identical to the one mentioned in AVOD, which first introduced in MV3D[4], and includes three different fusion schemes, *early fusion*, *late fusion* and *deep fusion*. Finally, the aggregated features are passed to a fully connected layer for regression after NMS.

Similar to FlowNet [7], we restrict correlation to a local neighborhood instead of all possible circular shifts in a feature map, which avoids large output dimensionality and too large displacements. 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 compare features in a local square window defined by the maximum displacement d, and i,j are the location of window center in 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, width of the feature map.

After above operation we get two correlation feature maps, one for point cloud $\mathcal{C}^{t,t+\tau}_{pc}$, and another for RGB image $\mathcal{C}^{t,t+\tau}_{img}$. Later ROI pooling and *early fusion* are performed just as in detection part. Aggregate feature $\mathcal{C}^{t,t+\tau}_{fusion} = \frac{1}{2}(\mathcal{C}^{t,t+\tau}_{pc} + \mathcal{C}^{t,t+\tau}_{img})$ is then flatten and fed to a fully connected layer to predict the transformation $\Delta^{t,t+\tau} = (\Delta^{t,t+\tau}_{x,y,z}, \Delta^{t,t+\tau}_{w,h,l}, \Delta^{t,t+\tau}_{r}) \in \mathbb{R}^7$ of the RoIs from t to $t+\tau$.

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 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:

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

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 bounding box regression target, and $\Delta_i^{*,t+\tau}$ is the 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 [12]. α and β are weight for L_{reg} and L_{corr}. 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 across two key frames. Additionally, For a single target $D_t = (D_t^{x,y,z}, D_t^{w,h,l}, D_t^r) \in \mathbb{R}^7$ in frame t and its track correspondence $D_{t+\tau}$ in frame $t+\tau$, the displacement regression values for the target $\Delta_i^{*,t+\tau}$ are encoded just like the 3D bounding box in [16].

3.4 3D streaming detection and tracking

Give a streaming point cloud of N frames $\{I_f\}$ for $f \in \{1,...N\}$, the streaming object detection task needs to predict a set of detections D_f for each frame I_f. Each detection set D_f consists of object detections $\{D_f^i\}$ while $i \in \{1,...N_f\}$ (N_f is the number of detections in frame f). Note that D_f can also be an empty set when no object is detected in a frame. In 3D object detection, Each detection D_f^i is parametrized as $D_f^i = (x_f^i, y_f^i, z_f^i, w_f^i, h_f^i, l_f^i, \theta_f^i, s_f^i)$, where (x_f^i, y_f^i, z_f^i) corresponds to the center (bottom center in KITTI[11] Datasets) of the detection box in point cloud, (w_f^i, h_f^i, l_f^i) corresponds to width, height, length of the detection box, θ_f^i is the rotation angle in yaw axis and s_f^i is the detectors confidence in the bounding box.

Because of the redundant features in streaming, we can only compute the object detections in key frames, while the detections in intermediate frames can be calculated using the detections in adjacent two key frames. Support we have two predicted object detections set $(D_f, D_{f+\tau})$ in two consecutive key frames $(I_f, I_{f+\tau})$ where τ is temporal stride, the detection results $\{D_{f+t}^i\}$ $(t \in \{1, \tau-1\})$ in intermediate frame D_{f+t} 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 is corresponding weight and $\mathcal{F}$ is generation function to produce D_{f+t}^i, in this paper we use quadratic interpolation function. Note that we can utilize (1) to generate D_{f+t}^i only when the target exists in D_f and $D_{f+\tau}$ simultaneously. if the target is emerged or end in intermediate frames, this method would be failed. Though we can develop the key frames selection function carefully to handle this situation, it is beyond the scope of this article. In this paper we focus on the targets that always exist between two key frames.

With object detections in each frame, multi-object tracking can be implemented by tracking-by-detection paradigm. For each bounding box in each frame, MOT try to associate it to a unique target trajectory $T_k = \{D_{f_1}^k, D_{f_2}^k, ..., D_{f_{N_k}}^k\}$, where k is trajectory id and N_k is the length of T_k, $\{f_1, f_2, ..., f_{N_k}\}$ are corresponding frames id.

4 Experimental evaluation

4.1 Datasets

We use the KITTI object tracking Benchmark [11] for evaluation. It consists of 21 training sequences and 29 test sequences with vehicles annotated in 3D. Each sequence includes hundreds of point clouds frames captured by Velodyne HDL-64E rotating 3D laser scanner and corresponding RGB images. We split 21 training sequences into two parts based on the parity of the sequence number, odd number for training and even number for evaluation. For multi-object tracking evaluation, we train our model in all 21 training sequences.

4.2 Training

Datasets preprocessing. Like other works based on KITTI, the point cloud is cropped 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 between two subsequent frames would be shifted due to the moving of the observer over time. Since the location and velocity information of each frame are available from the IMU data, one can calculate the displacement of the observer between different frames and translate the coordinates accordingly. By this way both point clouds and object labels are on the exact same coordinate system. Note that this approach is important to make the system invariant to the speed of the ego-car.

Training and testing. We train two networks, one for the *Car* class and another for both *Pedestrian* and *Cyclist* classes. We follow most of the super-parameter settings in AVOD[16] during training and testing. To be more specific, the network is trained for 120K iterations using an ADAM[14] 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 anchors for *Car*. For *Pedestrian* and *Cyclist* classes, the IoU thresholds are 0.3 and 0.45 respectively. To remove redundant proposals, we perform 2D non-maximum suppression(NMS) at an IoU threshold of 0.8 in BEV to keep the top 1024 proposals during training, while at inference time, top 300 proposals are used for *Car* class and top 1024 proposals are kept for *Pedestrian* and *Cyclist* class.

4.3 Results

3D object detection. Our work mainly based on 3D object detection, either streaming level detection or multi-object tracking, thus the network performance on 3D object detection is significant important. We train our Bi-AVOD on KITTI tracking datasets, and results are evaluated using KITTI object detection metrics. Since there is no way for us to get object detection performance on KITTI tracking testing datasets, we evaluate our Bi-AVOD on tracking evaluation datasets (described in Sec 4.1) instead. To explore the effectiveness of dual-way structure on object detection, we train original AVOD model on tracking datasets for comparison, results are shown in Table 1. Our Bi-AVOD model is shown to have comparable ability for *easy, moderate, hard* setting on all three classes. This means that the introduction of dual-way model and correlation operation do not reduce model performance on 3D object detection. In pursuit of better network performance, we pre-trained original AVOD model on KITTI object detection datasets, and then migrated relevant parameters to

			$AP_{3D}(\%)$			$AP_{BEV}(\%)$		
Method	Runtime (s)	Class	Easy	Moderate	Hard	Easy	Moderate	Hard
AVOD	0.01	Car	0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD*	0.01		0.90	0.80	0.70	0.80	0.70	0.60
AVOD	0.01	Ped.	0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD*	0.01		0.90	0.80	0.70	0.80	0.70	0.60
AVOD	0.01	Cyc.	0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD*	0.01		0.90	0.80	0.70	0.80	0.70	0.60

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

			$AP_{3D}(\%)$			$AP_{BEV}(\%)$		
Method	Runtime (s)	Class	Easy	Moderate	Hard	Easy	Moderate	Hard
AVOD	0.01	Car	0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 1$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 3$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 5$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
AVOD	0.01	Ped.	0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 1$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 3$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 5$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
AVOD	0.01	Cyc.	0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 1$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 3$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60
Bi-AVOD ($\tau = 5$)	0.01		0.90	0.80	0.70	0.80	0.70	0.60

Table 2: Performance comparison of different temporally strides τ during testing.

our Bi-AVOD model for further fine-tuned learning on KITTI tracking datasets. By this way a better network is obtained, results are shown in Table 1 where Bi-AVOD* is the fine-tuned model. Compared to Bi-AVOD which trained from scratch, the fine-tuned model outperform by 5.20% AP on the *moderate* setting and 4.05% on the *hard* setting respectively.

Streaming level detection. Leveraging accurate 3D object detection results of adjacent key frames and corresponding targets displacements, we can perform streaming level 3D object detection on autonomous driving scenarios. This part we investigate the effect of multi-frame input during testing, specifically, we focus on the influence on time and precise of different temporally strides τ. In Table 2 we see that linking detections to tubes based on corresponding targets displacements results in speed and precision improvements more or less. This gain is mostly for the following reason: if a bounding box is false negative in intermediate frame but is true positive in its two adjacent frames, it will be corrected during linking; on the other hand, a false positive detection will be modified to true positive if it considers as true bounding box in its adjacent frames. Note that a large τ will leads to a remarkable speed but accuracy decay in detection, there is a treat-off between runtime and precision. Our experiments show that $\tau = 3$ is the best choice on KITTI tracking datasets.

Multi-object tracking. We finally validate the efficacy of our Bi-AVOD network on multi-object tracking task. We compare our model to publicly available methods in the

Method	MOTA(%)	MOTP(%)	MT(%)	ML(%)	IDS	FRAG
CEM	51.94	77.11	20.00	31.54	125	396
RMOT	52.42	75.18	21.69	31.85	50	376
TBD	55.07	78.35	20.46	32.62	31	529
mbodSSP	56.03	77.52	23.23	27.23	0	699
SCEA	57.03	78.84	26.92	26.62	17	461
SSP	57.85	77.64	29.38	24.31	7	704
ODAMOT	59.23	75.45	27.08	15.54	389	1274
NOMT-HM	61.17	78.65	33.85	28.00	28	241
LP-SSVM	61.77	76.93	35.54	21.69	16	422
RMOT*	65.83	75.42	40.15	9.69	209	727
NOMT	66.60	78.17	41.08	25.23	13	150
DCO-X*	68.11	78.85	37.54	14.15	318	959
mbodSSP*	72.69	78.75	48.77	8.77	114	858
SSP*	72.72	78.55	53.85	8.00	185	932
NOMT-HM*	75.20	80.02	50.00	13.54	105	351
SCEA*	75.58	79.39	53.08	11.54	104	448
MDP	76.59	82.10	52.15	13.38	130	387
LP-SSVM*	77.63	77.80	56.31	8.46	62	539
NOMT*	78.15	79.46	57.23	13.23	31	207
MCMOT-CPD	78.90	82.13	52.31	11.69	228	536
DSM	76.15	83.42	60.00	8.31	296	868
Bi-AVOD(ours)	80.00	90.00	70.00	8.00	200	500

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

KITTI Tracking Benchmark, the results are listed in Table 3. It's shown that our approach is competitive with the start-of-the-art, outperforming all other methods in some of the metrics(best for MOTP and MT, second best for ML) by a large margin. Note that KITTI only evaluates the metrics in 2D, which does not fully represent the performance of our 3D approach. Also we refer the reader to Figure 5 for an visualization example of the multi-object tracking results, more examples are available in Supplementary material.

5 Conclusions

In this work we proposed Bi-AVOD, a unified framework for simultaneous 3D object detection and tracking in point cloud streaming. The network is a dual-way structure and can process two frames at a same time, which makes multi-frame detection possible. Meanwhile, embedding with a correlation module to encode the similarity and diversity of adjacent frames, our network can do object detection and tracking in a very efficient way. In multi-object tracking evaluation, our approach achieves accuracy competitive state-of-the-art methods while with a higher speed. In the future, we plan to improve our approach with an adaptive key frame selection algorithm.

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] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] 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.

[26] 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.

[27] 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.

[28] 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.