2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

PointSiamRCNN: Target-aware Voxel-based Siamese Tracker for Point Clouds

Hao Zou[1], Chujuan Zhang[1], Yong Liu[1,*], Wanlong Li[2], Feng Wen[2], and Hongbo Zhang[2]

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636863

Abstract— **Currently, there have been many kinds of point-based 3D trackers, while voxel-based methods are still under-explored. In this paper, we first propose a voxel-based tracker, named PointSiamRCNN, improving tracking performance by embedding target information into the search region. Our framework is composed of two parts for achieving proposal generation and proposal refinement, which fully releases the potential of the two-stage object tracking. Specifically, it takes advantage of efficient feature learning of the voxel-based Siamese network and high-quality proposal generation of the Siamese region proposal network head. In the search region, the ground-truth annotations are utilized to realize semantic segmentation, which leads to more discriminative feature learning with point-wise supervisions. Furthermore, we propose the Self and Cross Attention Module for embedding target information into the search region. Finally, the multi-scale RoI pooling module is proposed to obtain compact representations from target-aware features for proposal refinement. Exhaustive experiments on the KITTI tracking dataset demonstrate that our framework reaches the competitive performance with the state-of-the-art 3D tracking methods and achieves the state-of-the-art in terms of BEV tracking.**

I. Introduction

With the surging requirement of practical applications such as robotics and autonomous driving, rapid development has been achieved in 3D object tracking [1]–[6]. 3D sensors that can capture the real scene information are essential and critical for autonomous driving vehicles and robots. The most commonly used 3D sensors for real-world applications are LiDAR sensors, which generate point cloud data to provide accurate distance information and be more robust for illumination variation. Due to the sparseness and irregularity of point clouds, well-established visual trackers cannot be directly used for 3D object tracking. Moreover, 3D single object tracking faces the challenge from the enormous search space of 3D object.

Most existing 3D tracking methods can be divided into two categories: the RGBD-based methods and the point-based methods. The performance of the RGBD-based methods [7]–[10] relies heavily on 2D prediction results and cannot utilize point cloud information to generate high-quality proposals. The first point-based method SC3D [4] leverages exhaustive search to generate candidates in the 3D space and introduces the 3D Siamese network based tracker. Nevertheless, it only solves a similarity metric between each candidate and the template. Later works such as [6], [11] improve SC3D by executing the region proposal network (RPN) from birds-eye-view (BEV) or directly predicting the target center to generate proposals. In this way, the inefficient sampling process in SC3D can be avoided. However, the point-based methods have no reference to high-performance 2D tracking experience and cannot effectively embed target information into the search region.

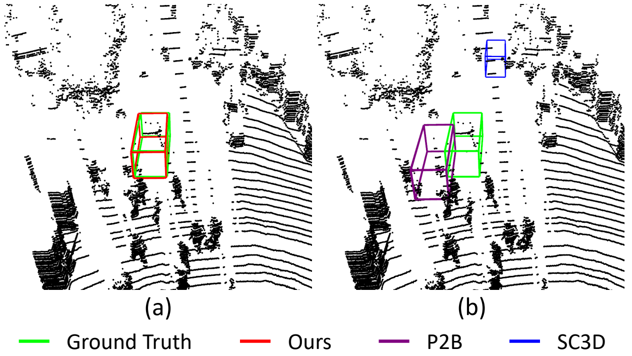

Fig. 1: Predicted tracking results from point clouds by (a) PointSiamRCNN and (b) P2B [6] and SC3D [4]. The proposed PointSiamRCNN can learn point-wise features and achieve better tracking performance.

One of the strategies for tackling the irregularity and disorder of point clouds is converting point clouds to voxel grids. The voxel-based methods have achieved remarkable performance in 3D object detection [12]–[15]. They commonly use 3D voxel Convolution Neural Network (CNN) to abstract features from voxels and reshape the 3D voxel features into the BEV representations for generating proposals by leveraging a 2D detection head. However, the voxel-based method was never explored in 3D object tracking.

Motivated by above observations, we propose the first voxel-based Siamese tracking framework named PointSiamRCNN for 3D object tracking based on 3D voxel CNN. The PointSiamRCNN consists of two stages, the first stage is constructed for generating high-quality proposals and learning target-aware features. We first construct the Siamese network based on 3D voxel CNN for encoding discriminative features from the template and search region, and then reshape the features to BEV representations. The Siamese region proposal network (Siamese-RPN) head generates proposals from BEV representations, which avoids the inefficient sampling method in SC3D and utilizes mature visual tracking experience. Inspired by [16], we observe that the 3D box annotations of tracking can provide the semantic masks and the track-id of each target, which guides the network to segment the intra-target points in the search branch. Besides,

[1] Hao Zou, Chujuan Zhang and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, 310027, China. (Yong Liu* is the corresponding author, email: yongliu@iipc.zju.edu.cn)

[2] Wanlong Li, Feng Wen and Hongbo Zhang are with the Huawei Noah's Ark lab

978-1-6654-1714-3/21/$31.00 ©2021 IEEE

we design the Self and Cross Attention (SCA) Module to learn target information and encode context information for learning the target-aware features in the search branch.

The second stage of PointSiamRCNN is designed for proposal refinement. Given the proposals from the first stage, we propose a multi-scale RoI pooling module to integrate the target-aware features of different spatial resolutions and transform the pooled points to canonical coordinates. Finally, our refinement network is constructed by set abstraction (SA) layers for further downsampling and extracting context features with two heads for confidence prediction and location refinement. By learning the target information and voxel-wise features in the search branch, our method achieves more accurate and robust tracking, as shown in Fig. 1.

The main contributions of our work can be summarized into four-fold.

- To the best of our knowledge, PointSiamRCNN is the first voxel-based Siamese tracker, which utilizes the 2D tracking head for generating a small number of high-quality 3D proposals from the BEV feature map.
- We propose the Self and Cross Attention (SCA) Module to learn the target information and encode strong context information, which enhances the discriminative power and obtains target-aware features in the search branch.
- We propose the multi-scale RoI pooling module to integrate target-aware features of different spatial resolutions, which simply and effectively provides compact representations for proposal refinement.
- Experimental results on the KITTI dataset demonstrate that our PointSiamRCNN outperforms state-of-the-art methods with remarkable margins and achieves 30 FPS inference time.

II. Related Work

In this section, we briefly introduce three tasks most related to our PointSiamRCNN.

A. 2D Object Tracking with Siamese Network based Methods

Recently, the 2D Siamese trackers have attracted widespread attention from the tracking community due to well-balanced tracking accuracy and speed. Many 2D Siamese trackers have achieved the state-of-the-art performance, such as [17]–[26]. SiamFC [27] first proposes a full convolution Siamese network with shared weights, which includes the template branch and the search branch for achieving object tracking. SiamRPN [28] and succeeding works [21]–[24] append the RPN with the classification branch and the regression branch after the Siamese network to further improves the performance of Siamese trackers. Although the Siamese trackers have achieved superior performance in images, especially well-balanced accuracy and speed, they cannot be directly leveraged for 3D tracking.

B. 3D Object Tracking with Siamese Network based Methods

Compared with the 2D Siamese tracking methods, the 3D Siamese tracking methods are still at the primary stage. SC3D [4] first proposes a 3D Siamese tracker based on PointNet [29] and leverages shape completion for regularizing feature learning to further improve tracking performance. The exhaustive search is executed for generating candidates, which makes it difficult to reach real-time speed. FST [5] proposes a double Siamese network and generates proposals from images by adding a 2D Siamese tracker before the 3D Siamese tracker. P2B [6] encodes the template and the search region based on PointNet++ [30], embeds the clues of the template into the search region, and then applies Hough voting [31] for predicting the target center. In this way, it effectively generates proposals but does not make full use of the 3D box annotations and the mature 2D tracking experience. For the first time, our method uses a Siamese-RPN head to generate proposals from the BEV feature map quickly and makes full use of the 3D box annotations for achieving semantic segmentation in the search branch.

C. 3D Object Detection with Voxel-based Methods

One of the strategies for tackling the irregularity of point clouds is converting point clouds to voxel grids. VoxelNet [32] produces regular voxels from the point cloud and encodes them with 3D CNN and 3D sparse convolution [33] is introduced by [12] for processing the voxels. Lang *et al.* [34] produces pseudo image features by stacking the voxels feature along the Z axis. Shi *et al.* [13] utilizes intra-object part information to learn more discriminative features and designs the RoI-aware point cloud pooling to aggregate part features. He *et al.* [14] uses a detachable auxiliary network to learn the structure information of point clouds for achieving accurate detection. The voxel-based method has achieved superior performance in 3D object detection, but it was never explored in 3D object tracking. We propose the first voxel-based tracker for 3D tracking and the experiments prove that it is more effective than the point-based tracker.

III. Method

In this section, we describe our two-stage tracker PointSiamRCNN for 3D single object tracking from point clouds, as illustrated in Fig. 2.

A. Backbone and tracking head

For the first time, we adopt a voxel-based Siamese network as the backbone for learning more discriminative features from point clouds, while previous point cloud Siamese trackers [3]–[6] employ the PointNet-based network as the backbone. For learning the target information and embedding target clues into the search region, we propose the Self and Cross Attention Module. Inspired by [16], due to the fact that 3D targets are independent of each other without overlapping, we achieve semantic segmentation by using free-of-charge semantic masks that directly supplied by 3D ground truth box in the training data. To the best of our knowledge, we propose the first voxel-based Siamese tracker.

1) Network architecture: The 3D Siamese network is constructed by two shared weights encoder with effective 3D sparse convolution. The encoder has four convolution blocks

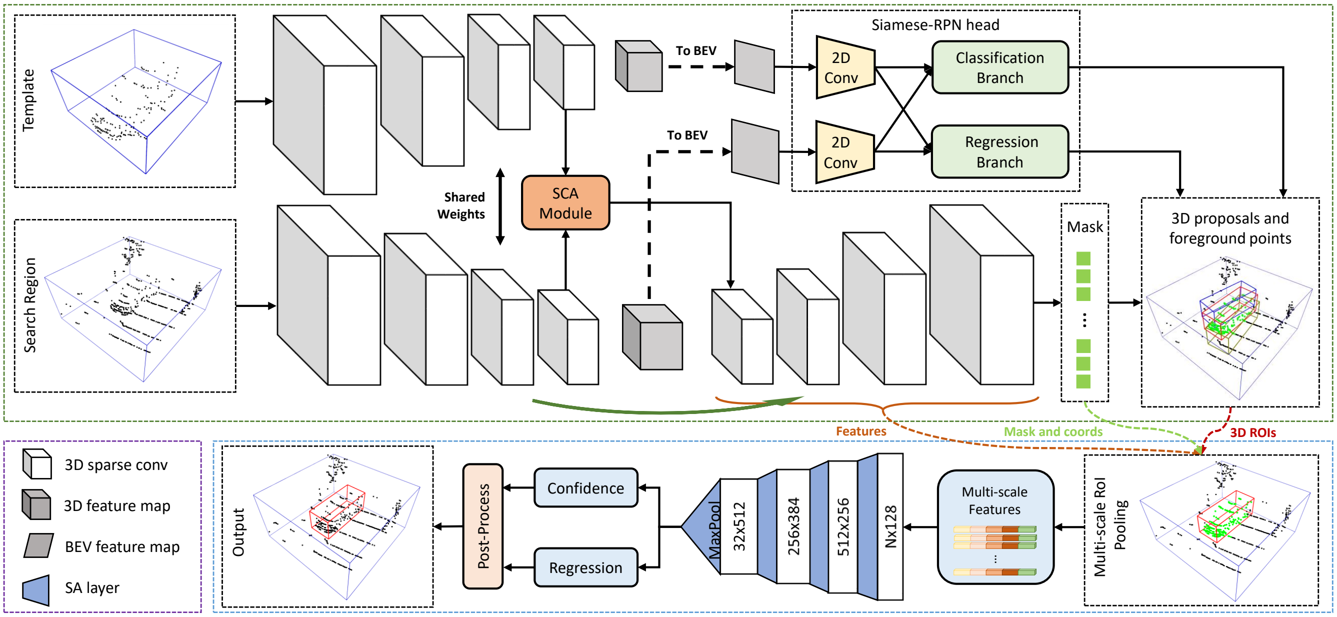

Fig. 2: The overall framework of our PointSiamRCNN. In the first stage, after encoding the template and search region features by the 3D voxel CNN, the Siamese-RPN head is utilized for proposal generation. We achieve the semantic segmentation and design a Self and Cross Attention (SCA) module, which can generate the target-aware features in the search region. In the second stage, the proposed RoI pooling module aggregates the target-aware features of different spatial resolutions to the compact representations for confidence prediction and location refinement.

with the kernel size of 3 and feature dimensions of 16-32-64-64, where the last three convolution blocks with stride 2 for down-sampling the spatial resolution by 8 times. For further learning the discriminative point-wise features from the search region, after the encoder, we append a decoder for semantic segmentation. The backbone composes an encoder-decoder architecture similarly with U-Net [35]. The decoder includes four sparse deconvolution blocks with the kernel size of 3 and feature dimensions 64-32-16-16, respectively. The stride of the last deconvolution block is set to 1, and the stride of the other three deconvolution blocks is set to 2. Each convolution and deconvolution is followed by a batch normalization [36] and ReLU. As a result, 3D feature maps with different spatial resolutions are produced from the search region. Considering that the number of foreground points is usually smaller than the number of background points in outdoor scenes, focal loss [37] is applied for calculating the segmentation loss $\mathcal{L}_{seg}$ to handle the class imbalance issue.

$$\mathcal{L}_{seg}(p_t) = -\alpha_t(1-p_t)^{\gamma}log(p_t),$$
$$\text{where } p_t = \begin{cases} p & \text{for foreground points,} \\ 1-p & \text{otherwise.} \end{cases} \quad (1)$$

2) Siamese-RPN head: In our case, after the sparse convolution based encoder downsamples the voxelized point clouds, we further abstract the features of the Z axis and the point clouds is downsampled on the X, Y, Z axis by 8, 8, 16 times. A 2D tracking head similar to [28] is applied for proposal generation from the BEV representations that are generated by stacking 3D feature maps along the Z axis from the template and search region. The 2D tracking head including 4 convolutions with a kernel size of 3 and the RPN with the classification and regression branch is leveraged to further abstract the BEV representations for achieving box scoring and location refinement (totally K proposals are generated).

3) Self and Cross Attention module: It is greatly important for the search branch to embed the target information for improving feature representation and learning target-aware semantic features. Inspired by [26], we propose the Self and Cross Attention (SCA) Module that consists of two sub-module: the Self Attention sub-module and the Cross Attention sub-module. In the Cross Attention sub-module, as shown in the upper part of Fig. 3, given the template features $\mathbf{Z} \in \mathbb{R}^{C\times h\times w\times d}$, we first reshape it to $\mathbf{Z}^r \in \mathbb{R}^{C\times M}$, where $M = h \times w \times d$ is the number of voxel features from the template. Then we perform matrix multiplication between $\mathbf{Z}^r$ and its transpose matrix, and apply the softmax layer to calculate the cross attention map $\mathbf{A}^c \in \mathbb{R}^{C\times C}$. We feed the search region features $\mathbf{X} \in \mathbb{R}^{C\times H\times W\times D}$ into a convolution layer, and then reshape it to $\mathbf{X}_1 \in \mathbb{R}^{C\times N}$, where $N = H \times W \times D$ is the number of voxel features from the search region. Finally, we perform a matrix multiplication between $\mathbf{A}^c$ and $\mathbf{X}_1$, and reshape the result for obtaining the final output $\mathbf{C} \in \mathbb{R}^{C\times H\times W\times D}$. In the Self Attention sub-module, as shown in the lower part of Fig. 3, we first reshape the search region features $\mathbf{X}$ to $\mathbf{X}^r \in \mathbb{R}^{C\times N}$, perform matrix multiplication between $\mathbf{X}^r$ and its transpose matrix, and then apply the softmax layer to calculate the self attention map $\mathbf{A}^s \in \mathbb{R}^{N\times N}$. Then, we feed the $\mathbf{X}$ into a convolution layer and reshape it to $\mathbf{X}_2 \in \mathbb{R}^{C\times N}$. Finally, we perform a matrix multiplication between $\mathbf{A}^s$ and $\mathbf{X}_2$, and then reshape the

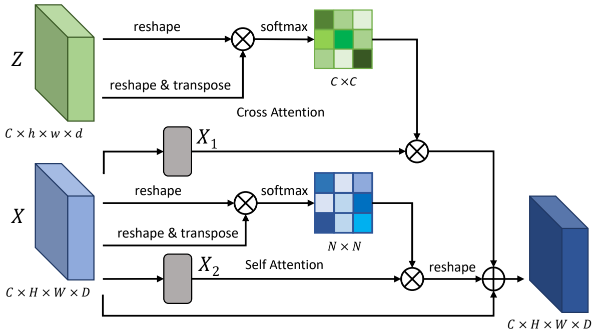

Fig. 3: The details of the Self and Cross Attention Module, where the $\bigoplus$ denotes the element-wise sum and the $\bigotimes$ denotes the matrix multiplication.

result for obtaining the final output $\mathbf{S}\in\mathbb{R}^{C\times H\times W\times D}$. The final output of the SCA module is the element-wise sum of $\mathbf{X}$, $\mathbf{C}$ and $\mathbf{S}$.

B. Multi-scale RoI point cloud pooling

Due to the fact that the object tracking is commonly the long-term tracking in real-world scenarios, the point distribution of the targets changes drastically from the early to late stage of the same tracklet. For handling the problem, we propose the multi-scale RoI point cloud pooling, which aims at aggregating the target-aware semantic feature of different spatial resolutions to obtain the compact representation for proposal refinement. Specifically, we expand each proposal with a constant value τ to obtain a new 3D box for extracting more context information. We denote $F^{(k)}=\left[f_0^{(k)},\ \cdots,\ f_{N^k-1}^{(k)}\right]$ as the feature vectors of each voxel in the k-th level of 3D voxel CNN, and their coordinates are $V^{(k)}=\left[v_0^{(k)},\ \cdots,\ v_{N^k-1}^{(k)}\right]$, where N^k represents the number of voxels in the k-th level. We select N_s^k voxels in the k-th level and retain the features denoted as

$$F_s^{(k)}=\left\{\left[f_j^{(k)};v_j^{(k)}-c_i\right]^T\left|\begin{array}{c}v_j^{(k)}\ in\ b_i,\\ j<N_s^k,\\ \forall v_j^{(k)}\in V^{(k)},\\ \forall f_j^{(k)}\in F^{(k)}.\end{array}\right.\right\},\qquad(2)$$

where the b_i denotes the i-th 3D box with center coordinate c_i. Then we use a multi-layer perceptron (MLP) to further abstract the features to the same dimension ψ of each layer. We perform the above process from different levels of the 3D voxel CNN and concatenate them to obtain multi-scale semantic features. For using shallow features that can provide fine-grained information, we also perform the above operations for the coordinate and mask of each voxel.

C. Refinement network

Given the target-aware semantic features of each proposal, we propose the refinement network for predicting the box location and size residuals between the proposal and their corresponding ground truth boxes and scoring each 3D proposal. Specifically, our refinement network follows [16] to transform the proposal to a local normalized coordinate system. As shown in the lower part of Fig. 2, we adopt Pointnet++ [30] (but not restricted to it), which is a hierarchical network for learning a discriminative feature with a progressive contextual scale for obtaining a discriminative feature vector, and then append two heads for confidence prediction and location refinement. With K proposals generated above, the proposal with the highest proposal-wise score is selected as the final result.

D. Loss functions

Our PointSiamRCNN framework can be trained end-to-end with the Siamese-RPN loss $\mathcal{L}_{srpn}$, the semantic segmentation loss $\mathcal{L}_{seg}$ and the refinement network loss $\mathcal{L}_{rn}$. We adopt the regression targets following [12], [13], [32] and utilize the smooth-L1 loss for anchor box regression. For the confidence prediction, we adopt the binary cross entropy loss. The Siamese-RPN loss $\mathcal{L}_{srpn}$ can be formulated as

$$\mathcal{L}_{srpn}=\mathcal{L}_{cls}+\sum_{res\in\mathcal{B}}\mathcal{L}_{smooth-L1}(\widehat{\Delta res^a},\Delta res^a),\qquad(3)$$

where $\mathcal{B}=\{x,y,z,w,l,h,\theta\}$, $\widehat{\Delta res^a}$ is the predicted result, Δres^a is the corresponding ground-truth target calculated as

$$\begin{array}{c}\Delta x^{(a)}=\frac{x^{(gt)}-x^{(a)}}{d^{(a)}},\Delta y^{(a)}=\frac{y^{(gt)}-y^{(a)}}{d^{(a)}},\\ \Delta z^{(a)}=\frac{z^{(gt)}-z^{(a)}}{h^{(a)}},\Delta w^{(a)}=log(\frac{w^{(gt)}}{w^{(a)}}),\\ \Delta l^{(a)}=log(\frac{l^{(gt)}}{l^{(a)}}),\Delta h^{(a)}=log(\frac{h^{(gt)}}{h^{(a)}}),\\ \Delta\theta^{(a)}=sin(\theta^{(gt)}-\theta^{(a)}),\end{array}\qquad(4)$$

from the candidate $(x^{(a)},y^{(a)},z^{(a)},w^{(a)},l^{(a)},h^{(a)},\theta^{(a)})$, the ground truth $(x^{(gt)},y^{(gt)},z^{(gt)},w^{(gt)},l^{(gt)},h^{(gt)},\theta^{(gt)})$ and $d^{(a)}=\sqrt{(l^{(a)})^2+(w^{(a)})^2}$. The semantic segmentation loss $\mathcal{L}_{seg}$ is the focal loss similar to (1). The refinement network loss $\mathcal{L}_{rn}$ can be formulated as

$$\mathcal{L}_{rn}=\mathcal{L}_{cls}+\sum_{res\in\mathcal{B}}\mathcal{L}_{smooth-L1}(\widehat{\Delta res^r},\Delta res^r),\qquad(5)$$

where $\widehat{\Delta res^r}$ is the predicted result, Δres^r is the corresponding ground-truth target calculated as (4). The overall loss function of our PointSiamRCNN is the sum of the three losses as

$$\mathcal{L}_{total}=\mathcal{L}_{srpn}+\mathcal{L}_{seg}+\mathcal{L}_{rn},\qquad(6)$$

where each loss has equal loss weights.

IV. Experiments

In this section, we introduce the experimental details of our PointSiamRCNN framework and compare it with the state-of-the-art methods [4]–[6] on the KITTI [38] 3D/BEV tracking dataset. At the same time, we conduct detailed ablation experiments to verify the effectiveness of the proposed modules on the most commonly used *car* category.

A. Experimental Setup

1) Dataset: We evaluate our framework on the KITTI [38] tracking dataset. The entire dataset has 21 scenes. Following [4]–[6], we use scenes 0-16 as the training set, 17-18 as the validation set and 19-20 as the test set.

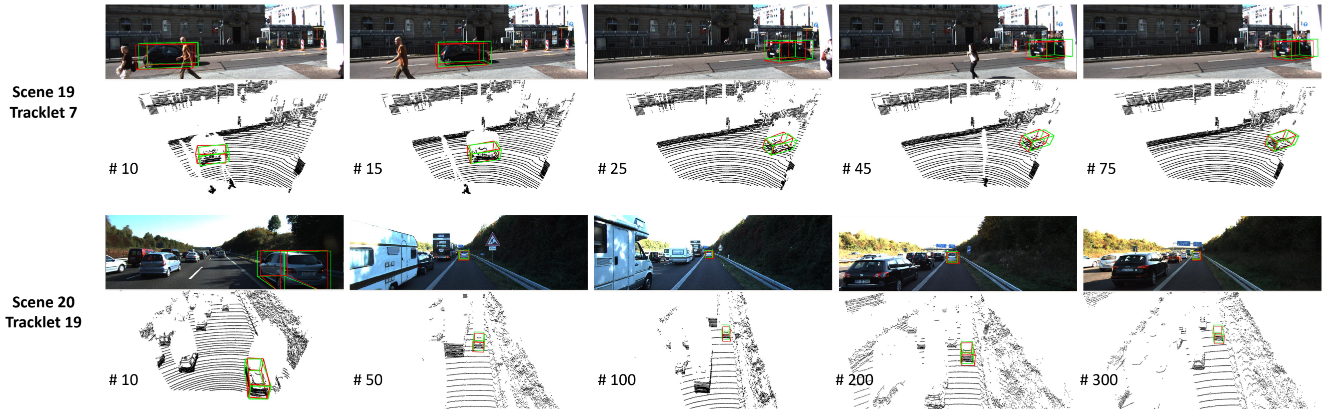

Fig. 4: Qualitative results using the previous result as the search center on the test set. We can observe that our method achieves superior performance even in long distance and object occlusion scenarios.

TABLE I: Performance comparison using 3D and BEV object tracking metric on the *car* class of the test set. Center denotes the different search centers of generating search region. Succ and Prec denote Success and Precision, respectively. The bold value indicates the top performance.

	3D Tracking						BEV Tracking					
Center	Previous result		Previous GT		Current GT		Previous result		Previous GT		Current GT	
Metric	Succ	Prec	Succ	Prec	Succ	Prec	Succ	Prec	Succ	Prec	Succ	Prec
SC3D [4]	41.3	57.9	64.6	74.5	76.9	81.3	39.5	47.3	66.5	75.9	77.0	81.5
FST [5]	37.1	50.6	68.2	77.1	81.6	87.3	43.3	51.5	69.1	78.5	82.4	88.7
P2B [6]	**56.2**	**72.8**	**82.4**	90.1	84.0	90.3	**70.8**	76.9	81.7	89.4	84.7	90.6
PointSiamRCNN	51.5	68.9	80.1	**91.5**	**84.8**	**93.1**	66.4	**77.1**	**82.4**	**91.9**	**85.2**	**93.5**

TABLE II: Extensive comparisons using 3D object tracking metric on the *car* class of the test set. The previous result is used as the center of the search region.

Scene	19		20	
Metric	Success	Precision	Success	Precision
SC3D [4]	30.5	36.0	39.1	56.2
FST [5]	31.3	39.8	40.9	58.7
P2B [6]	46.7	60.3	**57.8**	**73.7**
PointSiamRCNN	**56.2**	**73.3**	50.7	68.1

2) Evaluation metric: We use the One Pass Evaluation (OPE) [39] as the evaluation metric, which defines the overlap that can be calculated as the Intersection-over-Union (IoU) between a bounding box and its corresponding ground truth (GT) box, and the error as the distance between both centers. The Success and Precision metrics are respectively defined by using the overlap and error Area Under Curve.

3) Implementation details: For the template, we reserve the intra-target points that lie between the range (-2m, 2m), (-2m, 2m), (-3m, 1m) along the X, Y, Z axis in the target center. For the search region, we select all the points that lie between the range (-4m, 4m), (-4m, 4m), (-3m, 1m) along the X, Y, Z axis in the target center. Reserved points are voxelized with each voxel size (2cm, 2cm, 4cm) on each axis. The width, length and height of each anchor for the car are (1.6m, 3.9m, 1.56m), respectively. For the multi-scale RoI point cloud pooling module, we set the τ as 0.5m, N_s^k as 256 and ψ as 128 fellow . All anchors that do not contain points are ignored. We utilize the Adam [40] optimizer with a learning rate of 0.001 for the first 50 epochs and then decay it to 0.0001 for the last 50 epochs to train our framework end-to-end. We use the fusion of the first ground truth and previous result as the strategy of template update. To further improve the performance, we apply data augmentation during the training stage, such as randomly translated and rotated.

B. 3D Object Tracking on the KITTI Tracking Dataset

We compare PointSiamRCNN with state-of-the-art methods using the most commonly used *car* category on both validation set and test set of the KITTI 3D/BEV tracking dataset. All the methods are trained on the train set and evaluated on the validation and test set.

1) Evaluation of 3D/BEV tracking: Following [4]–[6], we generate the search region from the center of the previous result, previous GT and current GT, respectively. It is noteworthy that the number of candidates for SC3D, FST, P2B and PointSiamRCNN are 128, 72, 64 and 40, respectively. As illustrated in Table I, our method reaches the best performance in both 3D and BEV tracking tasks with the current GT as the search center. Specifically, our method leads the sate-of-the-art method [6] by (**0.8%/2.8%**) in 3D tracking and (**0.5%/2.9%**) in BEV tracking. In terms of 3D tracking, our method reaches the competitive performance with the state-of-the-art methods and achieves the state-of-the-art performance in BEV tracking. We further adopt extensive comparisons using the 3D object tracking metric with different scenes on the test set, as shown in Table II. In scene 19, we achieve a notable improvement, the Success and Precision increase by 9.5%/13% in 3D tracking. However,

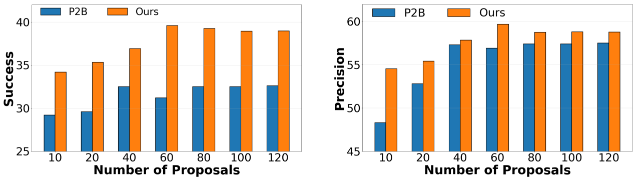

Fig. 5: Illustration the performance of different number of proposals.

TABLE III: Comparison between with and without the SCA module. The bold value indicates the top performance.

Metric	Success	Precision
with SCA module	**39.6**	**59.7**
without SCA module	36.2	55.4

we find that our method fails when the initial template is too sparse to generate effective target information and the voxelization inevitably loses the fine-grained information, which reduces the performance in scene 20. We also illustrate the prediction results using previous results as the search center and project the tracking result into the image for better visualization, as shown in Fig. 4.

2) Evaluation of 3D proposal generation: We evaluate the performance of our method and P2B [6] with different numbers of proposals. As shown in Fig. 5, our method achieves significantly higher performance than P2B. With only 60 proposals, our method obtains **39.6%/59.7%**, which outperforms 31.2%/56.9% of P2B by 8.4%/2.8% at the same number of proposals. When using 60 proposals, our method achieves the best performance, while P2B achieves the best performance using 100 proposals. In summary, our method can be more robust to the number of proposals and achieve a better balance between tracking speed and accuracy.

3) Runtime analysis: We analyze the runtime for each part of our framework separately. PointSiamRCNN achieves 30 FPS, including 8ms for prepossessing point clouds, 19 ms for the stage-one network, 6 ms for the stage-two network and 0.5 ms for post-process on a desktop equipped with an Intel i7 CPU and a 1080Ti GPU, while other methods cannot achieve real-time running speed, except P2B.

C. Ablation studies

In this section, we develop detailed experiments to analyze the effect of the proposed modules. All models are trained in the training set and evaluated in the validation set.

1) Effect of the SCA module: As discussed above, the SCA module learns the discriminative features from the search region and template. In order to verify the effect of the SCA module, we conduct experiments between with and without the SCA module, as shown in Table III. We can observe that our SCA module can learn more discriminative features and make full use of template features and search region features to achieve better tracking performance.

2) Effect of the multi-scale RoI pooling module: As shown in Table IV, we explore the importance of each feature components in multi-scale features. The first row shows that the performance is significantly reduced, where we only use the shallow semantic features. Rows 2 to 5 further improve the performance by aggregating shallow semantic features and high-level semantic features. The last row shows that the performance of only using high-level semantic features is also reduced since the lack of shallow features cannot capture the fine-grained information. The best performance is achieved by aggregating all high-level features and shallow features of different resolutions in 5^{th} row.

TABLE IV: Effects of different feature components for the multi-scale RoI pooling module.

$F_s^{(1)}$	$F_s^{(2)}$	$F_s^{(3)}$	$F_s^{(4)}$	$Coords$	Success	Precision
-	-	-	-	✓	35.6	55.1
-	-	-	✓	✓	36.3	57.0
-	-	✓	✓	✓	38.1	58.0
-	✓	✓	✓	✓	39.4	58.2
✓	✓	✓	✓	✓	**39.6**	**59.7**
✓	✓	✓	✓	-	37.9	57.7

TABLE V: Effects of different strategy of the template update. The "First & Prev" denotes the first ground truth and previous result, "All" denotes all previous result.

Metric	Success			Precision		
Strategy	SC3D	P2B	Ours	SC3D	P2B	Ours
First	20.9	23.3	**34.8**	38.4	37.8	**50.9**
Previous	15.9	26.4	**35.6**	30.3	46.3	**55.5**
First & Prev	25.2	31.0	**39.6**	44.8	55.3	**59.7**
All	28.3	29.5	**37.7**	47.1	49.8	**56.6**

3) Strategy of the template update: As one of the main challenges of object tracking, we evaluate four template update strategies for template generation: using the first GT, using the previous result, using the fusion of the first GT and the previous result and using the fusion of the first GT and all previous results. As shown in Table V, the strategy that uses the fusion of the first GT and previous result as the template achieves the best performance, where the previous result provides real-time target information for the tracker and the first GT provides accurate tracking clues throughout the tracking process. Our method outperforms previous state-of-the-art methods with remarkable margins in all settings.

V. Conclusion

In this paper, we have presented PointSiamRCNN, a two-stage object tracker for 3D tracking from point clouds. The voxel-based Siamese network with the Siamese-RPN head is firstly adopted to generate 3D proposals for 3D tracking. Meanwhile, the free-of-charge 3D tracking annotations are made full use for achieving semantic segmentation. The proposed SCA Module guides the network to be aware of the target information and encodes strong context information. Moreover, the RoI point cloud pooling module is applied to aggregate the target-aware features for generating compact representations. Experimental results on the KITTI tracking dataset demonstrate that our framework significantly improves tracking performance with real-time running speed.

References

[1] K.-H. Lee and J.-N. Hwang, "On-road pedestrian tracking across multiple driving recorders," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1429–1438, 2015.

[2] E. Machida, M. Cao, T. Murao, and H. Hashimoto, "Human motion tracking of mobile robot with kinect 3d sensor," in *2012 Proceedings of SICE Annual Conference (SICE)*. IEEE, 2012, pp. 2207–2211.

[3] Y. Cui, Z. Fang, and S. Zhou, "Point siamese network for person tracking using 3d point clouds," *Sensors*, vol. 20, no. 1, p. 143, 2020.

[4] S. Giancola, J. Zarzar, and B. Ghanem, "Leveraging shape completion for 3D siamese tracking," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 1359–1368, 2019.

[5] H. Zou, J. Cui, X. Kong, C. Zhang, Y. Liu, F. Wen, and W. Li, "F-siamese tracker: A frustum-based double siamese network for 3d single object tracking," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8133–8139.

[6] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, "P2B: Point-to-Box Network for 3D Object Tracking in Point Clouds," 2020. [Online]. Available: http://arxiv.org/abs/2005.13888

[7] A. Asvadi, P. Girão, P. Peixoto, and U. Nunes, "3D object tracking using RGB and LIDAR data," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, no. May 2018, pp. 1255–1260, 2016.

[8] U. Kart, J.-K. Kamarainen, and J. Matas, "How to make an rgbd tracker?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[9] U. Kart, A. Lukezic, M. Kristan, J.-K. Kamarainen, and J. Matas, "Object tracking by reconstruction with view-specific discriminative correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1339–1348.

[10] Y. Liu, X.-Y. Jing, J. Nie, H. Gao, J. Liu, and G.-P. Jiang, "Context-aware three-dimensional mean-shift with occlusion handling for robust object tracking in rgb-d videos," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 664–677, 2018.

[11] J. Zarzar, S. Giancola, and B. Ghanem, "Efficient Tracking Proposals using 2D-3D Siamese Networks on LIDAR," mar 2019. [Online]. Available: http://arxiv.org/abs/1903.10168

[12] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors (Switzerland)*, vol. 18, no. 10, oct 2018.

[13] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 1–1, 2020.

[14] C. He, H. Zeng, J. Huang, X.-s. Hua, and L. Zhang, "Structure Aware Single-stage 3D Object Detection from Point Cloud," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2020.

[15] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," 2019. [Online]. Available: http://arxiv.org/abs/1912.13192

[16] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770–779.

[17] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 1328–1338.

[18] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: residual attentional siamese network for high performance online visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4854–4863.

[19] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4591–4600.

[20] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11213 LNCS, pp. 103–119, 2018.

[21] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[22] Y. Xu, Z. Wang, Z. Li, Y. Ye, and G. Yu, "SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines," 2019. [Online]. Available: http://arxiv.org/abs/1911.06188

[23] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese Box Adaptive Network for Visual Tracking," 2020. [Online]. Available: http://arxiv.org/abs/2003.06761

[24] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking," 2019. [Online]. Available: http://arxiv.org/abs/1911.07241

[25] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual Tracking by Re-Detection," 2019. [Online]. Available: http://arxiv.org/abs/1911.12836

[26] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable siamese attention networks for visual object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6728–6737.

[27] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[28] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[31] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep hough voting for 3D object detection in point clouds," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 9276–9285, 2019.

[32] Z. Liang, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *Computers in Education Journal*, vol. 6, no. 3, pp. 46–48, 1996.

[33] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9224–9232.

[34] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," pp. 12 697–12 705, 2018. [Online]. Available: http://arxiv.org/abs/1812.05784

[35] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[39] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.