Tracklet Proposal Network for Multi-Object Tracking on Point Clouds

Hai Wu[1] , **Qing Li**[1] , **Chenglu Wen**[1*] , **Xin Li**[2] , **Xiaoliang Fan**[1] and **Cheng Wang**[1]
[1]School of Informatics, Xiamen University
[2]School of Electrical Engineering and Computer Science, Louisiana State University
wuhai@stu.xmu.edu.cn, hello.qingli@gmail.com, clwen@xmu.edu.cn, xinli@cct.lsu.edu,
{fanxiaoliang, cwang}@xmu.edu.cn

Abstract

This paper proposes the first tracklet proposal network, named PC-TCNN, for Multi-Object Tracking (MOT) on point clouds. Our pipeline first generates tracklet proposals, then refines these tracklets and associates them to generate long trajectories. Specifically, object proposal generation and motion regression are first performed on a point cloud sequence to generate tracklet candidates. Then, the spatial-temporal features of each tracklet are exploited, and their consistency is used to refine the tracklet proposal. Finally, the refined tracklets across multiple frames are associated to perform MOT on the point cloud sequence. The PC-TCNN significantly improves the MOT performance by introducing the tracklet proposal design. On the KITTI tracking benchmark, it attains an MOTA of 91.75%, outperforming all submitted results on the online leaderboard.

1 Introduction

Multi-Object Tracking (MOT), which is a key technology for extracting dynamic information from the environment, has wide applications in autonomous driving and robotics. Most existing methods explore MOT on images [Hu *et al.*, 2019; Mykheievskyi *et al.*, 2020] or LiDAR point clouds [Simon *et al.*, 2019; Weng *et al.*, 2020a]. These methods usually follow a tracking-by-detection paradigm that associates the detected objects across frames, and their tracking performance is often restricted by inevitable *detection failures* (e.g., due to heavily occluded objects).

To tackle detection failures, in image-based MOT, previous methods [Choi, 2015; Zhang *et al.*, 2020] use *short trajectories* (tracklets), composed by multiple detections in a short time, as a basis for object association to form longer trajectories. Recently, [Sun *et al.*, 2020] directly generates tracklets from image sequences in an end-to-end manner, and achieves state-of-the-art performance in image-based MOT. For point cloud-based MOT, however, such an end-to-end tracklets-based network design is underexplored. The main challenge here is how to produce accurate tracklets from irregular and unordered point clouds, as more accurate tracklets will lead to higher MOT performance. [Luo *et al.*, 2018] proposes an end-to-end network that decodes tracklets by averaging current detections and previous predictions; however, its accuracy is sensitive to the quality of the predictions.

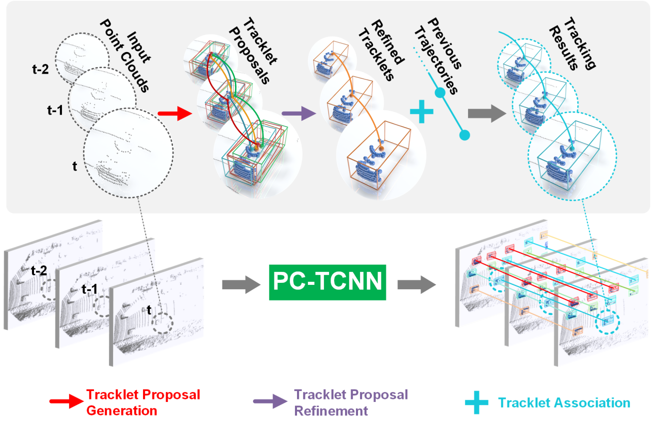

Figure 1: Our PC-TCNN performs MOT on point clouds by generating tracklet proposals, refining the proposals, and associating the refined tracklets to generate long trajectories.

The accuracy of a tracklet relies on the quality of detections inside it. In a tracklet, an object across consecutive frames shares some consistent spatial-temporal features. For example, in point clouds, the width, length, and height of rigid objects (e.g., vehicles) are consistent in consecutive frames. Even for non-rigid objects (e.g., pedestrians), the size, volume, and geometry are often similar in consecutive point cloud frames. Such consistency constraints in spatial-temporal features can improve the quality of detections. We investigate to learn and use such features to improve tracklet accuracy by refining the quality of detections inside tracklets.

The proposal-based methods have been successfully applied in object detection [Shi *et al.*, 2020], single object tracking [Li *et al.*, 2019] and action detection [Hou *et al.*, 2017] by performing object localization and recognition in a coarse-to-fine manner. Through such a proposal-based coarse-to-fine refinement, we could obtain more accurate tracklets, and consequently, better tracking objects in consecutive frames.

[*] Corresponding author

Inspired by this, we develop a novel Tracklet proposal Convolutional Neural Network, named PC-TCNN, for MOT on Point Clouds. Our PC-TCNN performs MOT by (1) generating tracklet proposals, (2) refining tracklet proposals, and then (3) associating the refined tracklets, as shown in Figure 1. Specifically, it first produces a set of high-recall tracklet proposals as regions of interest by performing object proposal generation and motion regression on point cloud sequences. Then, PC-TCNN employs a tracklet proposal refinement scheme to refine the *tracklet detections* (detections inside a tracklet) using spatial-temporal features aggregated from each tracklet proposal, thereby improving tracklet accuracy. Furthermore, to form long trajectories, our PC-TCNN associates the refined tracklets using a greedy assignment. Our contributions are summarized as follows:

- We propose the first tracklet proposal network, which generates tracklet proposals, refines the proposals, and associates the refined tracklets on point cloud sequences, leading to significantly improved MOT performance;
- We introduce a tracklet proposal refinement scheme that uses the consistency constraints in spatial-temporal features of each tracklet proposal to refine the tracklet detections, and this can greatly enhance the accuracy of tracklets;
- Our method achieves the state-of-the-art on the KITTI tracking benchmark [Geiger *et al.*, 2012] and ranks 1^{st} among all submitted methods as of Jan. 20th, 2021.

2 Related Work

Tracking-by-detection on point clouds. Most existing MOT methods on point clouds follow a tracking-by-detection paradigm. They first detect objects from single frame, then associate the objects across frames by applying a filtering algorithm [Simon *et al.*, 2019; Weng *et al.*, 2020a]. Some work constructs deep networks to associate detected objects [Baser *et al.*, 2019; Wang *et al.*, 2020]. Recent studies also investigate MOT using both point clouds and image data [Weng *et al.*, 2020b]. In these methods, the detection is separated from tracking. Hence, their tracking performance is restricted by inevitable detection failures.

Tracklet-based MOT. To tackle detection failures, some previous image-based MOT methods [Bae and Yoon, 2014; Choi, 2015; Zhang *et al.*, 2020] first generate tracklets by independently detecting and matching similar objects in consecutive image frames, then associate the tracklets to generate long trajectories. These methods are less sensitive to detection failure. Recently, [Sun *et al.*, 2020] proposed a DMM-Net that generates tracklets from images in an end-to-end manner. [Luo *et al.*, 2018] proposed an end-to-end FaF network that merges current detections and past predictions to decode tracklets on point clouds; however, its accuracy relies on the prediction quality. Our method directly exploits feature consistency of objects in consecutive frames to improve the detection accuracy, leading to more accurate tracklets.

Proposal-based methods. The proposal-based methods, which usually contain proposal generation and proposal refinement stages, perform through a coarse-to-fine framework. The region proposal-based methods, specifically, generate regions of interest in the spatial domain, and have been successfully applied for 2D object detection [Girshick *et al.*, 2014], single 2D object tracking [Li *et al.*, 2019] and 3D object detection [Shi *et al.*, 2019; Shi *et al.*, 2020]. These methods were designed for single frame data and do not use temporal features. Recently, tube proposal-based methods, which models features by taking into account temporal information, have been used in action detection [Hou *et al.*, 2017] and recognition [Wu *et al.*, 2020]. These models were developed on regular image data rather than irregular point clouds. To our best knowledge, our method is the first tracklet proposal framework on point cloud sequences for MOT.

3 Our Method

We illustrate our tracklet proposal network in Figure 2. The framework consists of three modules: tracklet proposal generation, tracklet proposal refinement, and tracklet association.

3.1 Tracklet Proposal Generation

To localize spatial-temporal region of interest in point clouds, which we denote as *tracklet candidates* or *tracklet proposals*, we propose a tracklet proposal generation module. At each timestamp t, the proposed module takes multiple point cloud frames $\{P_i\}_{i=t-n}^{t}$ (n past frames and one current frame) as inputs, and encodes the input point clouds into multiple bird view feature maps. Then, it performs 3D object proposal generation and motion regression on these feature maps to generate a set of high recall tracklet proposals.

Backbone Network

To efficiently convert point cloud sequences into 3D feature volumes, we employ the 3D sparse convolution [Shi *et al.*, 2020] to obtain the spatial features. Specifically, we first register the input frames to the coordinate system of the last frame using the GPS/IMU data. The registered point cloud frames are denoted by $\{P_i\}_{i=-n}^{0}$. Then, we voxelize multiple point cloud frames with a spatial resolution of $L \times W \times H$ (length, width and height), and obtain voxels $\{P_i'\}_{i=-n}^{0}$. For each voxel, we calculate the raw features using mean of a 4-tuple (3D coordinates + intensities) raw features of all inside points. Then, we apply a series of shared 3D sparse convolution $\mathcal{S}(\cdot)$ on the voxelized point cloud frames to obtain the spatial features $\{F_i^{3D}\}_{i=-n}^{0}$ by

$$F_i^{3D} = \mathcal{S}(P_i'), i = -n, \dots, 0. \qquad (1)$$

Here $\mathcal{S}(\cdot)$ consists of a series of $3{\times}3{\times}3$ 3D sparse convolution kernels, which downsample the spatial features to $1\times$, $2\times$, $4\times$, and eventually a $8\times$ downsampled tensor.

Inspired by the convolutional gated recurrent unit (ConvGRU) [Tokmakov *et al.*, 2017], which has been proven to be effective in modeling temporal features in 2D images, we incorporated it into our tracklet proposal generation module to compute temporal features of point clouds. We first compress the spatial features $\{F_i^{3D}\}_{i=-n}^{0}$ along the H dimension into multiple bird's eye view (BEV) feature maps $\{F_i^{BEV}\}_{i=-n}^{0}$. Then we apply the ConvGRU $\mathcal{G}(\cdot)$ on BEV feature maps to compute the temporal features $\{F_i^{T}\}_{i=-n}^{0}$, by

$$F_i^{T} = \mathcal{G}(F_i^{BEV}, F_{i-1}^{T}), i = -n, \dots, 0, \qquad (2)$$

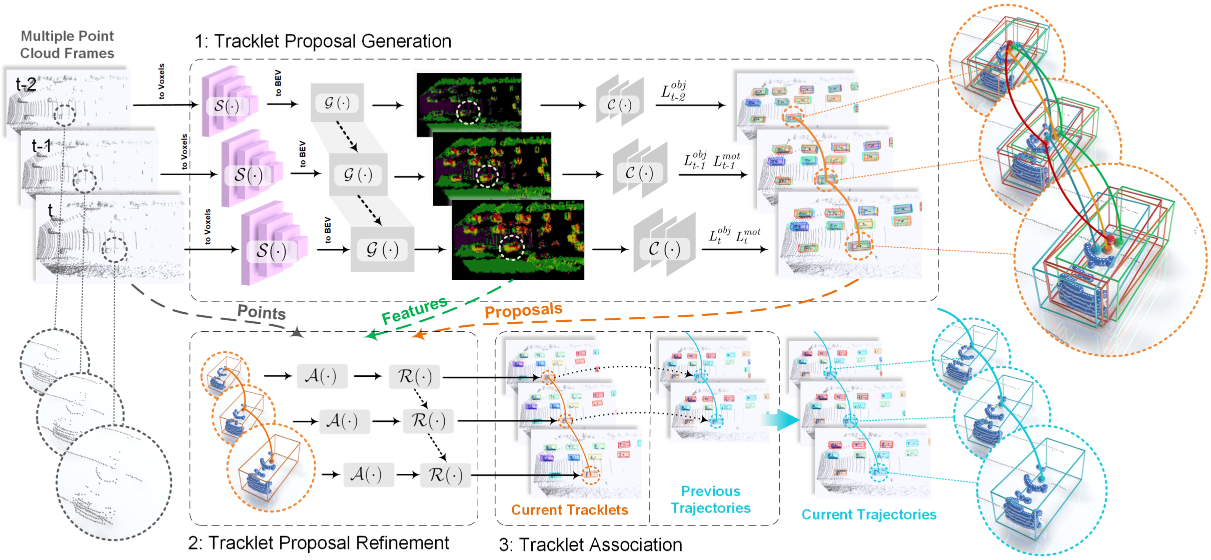

Figure 2: The framework of PC-TCNN. (1) At each timestamp t, PC-TCNN takes a point cloud sequence as inputs, and performs object proposal generation and motion regression on the spatial-temporal features of point clouds to generate high recall tracklet candidates. (2) PC-TCNN extracts and aggregates detailed spatial-temporal features of each proposal to refine the tracklet proposals. (3) PC-TCNN associates the refined tracklets with previous trajectories (initialized by empty set at first timestamp) to generate final tracking results.

where F^T_{-n-1} are initialized to zeros.

The temporal features are further encoded into high-level BEV features $\{F^H_i\}^0_{i=-n}$ (which represent more comprehensive spatial-temporal features of point clouds) by a series of shared 2D convolution $\mathcal{C}(\cdot)$,

$$F^H_i = \mathcal{C}(F^T_i), i = -n, \ldots, 0. \tag{3}$$

Tracklet Proposal Head

A tracklet in point clouds is defined by multiple 3D bounding boxes (BBs), which encode objects' 3D sizes, motions and correspondences during a short period. We denote the 3D BB of j-th tracklet proposal in i-th frame as $B^j_i = [x^j_i, y^j_i, z^j_i, w^j_i, h^j_i, l^j_i, \alpha^j_i]$, which includes object coordinates, width, height, length and orientation angle. Then the j-th tracklet proposal is given by $T^j = \{B^j_i\}^0_{i=-n}$.

The tracklet proposal head takes the BEV feature map sequence $\{F^H_i\}^0_{i=-n}$ as inputs. At each location (pixel) on each feature map, we employ an object proposal head to compute the object confidence and residuals, and construct a motion head to compute a cross-frame-offset of object motion. We employ non-maximum suppression (NMS) on the last frame of BB proposal head, to obtain the seeds of tracklet proposal. Starting from the tracklet proposal seeds, we generate the m tracklet proposals, denoted by $\mathcal{T}_t = \{T^j | j = 0, \ldots, m\}$, based on the cross-frame-offsets.

3.2 Tracklet Proposal Refinement

To improve the quality of tracklet detections, which is essential in attaining more accurate tracklets, we propose a tracklet proposal refinement module. In most proposal-based approaches [Shi *et al.*, 2020; Hou *et al.*, 2017], a main challenge in the refinement stage is how to effectively extracting features from area of interest. Previous feature extraction methods (e.g., region of interest pooling and tube of interest pooling) are mostly designed for single frame [Girshick *et al.*, 2014; Shi *et al.*, 2020] or video data [Hou *et al.*, 2017] and cannot be directly applied on point cloud sequences. To tackle this, we design a tracklet feature aggregation method to capture features from spatial-temporal region of interest on point cloud sequence, leading to more accurate object recognition and localization in tracklets.

Tracklet Features Aggregation

We first extract the points of each tracklet proposal by employing context-aware point cloud pooling [Shi *et al.*, 2019]. For i-th 3D BB B^j_i in the j-th tracklet proposal, we randomly extract b object points from input point cloud frame P_i. These points are denoted by $\mathcal{P}^j_i = \{p^{j,k}_i | k = 0, \ldots, b\}$, where $p \in \mathbb{R}^3$ indicates point coordinates.

The extracted tracklet points contain accurate local object features inside BB, but lack surrounding and global features that are outside. Therefore, we project each point $p^{j,k}_i$ to the coordinate system of backbone features F^T_i, and utilize bilinear interpolation to obtain new backbone features $f^{j,k}_i$. To exploit more accurate local geometry, we transform the extracted points to the canonical coordinate system of the corresponding BB, denoted by $\hat{p}^{j,k}_i$. Thus, for i-th 3D BB in the j-th tracklet proposal, we obtain a set of extracted features $\mathcal{F}^j_i = \{[\hat{p}^{j,k}_i, f^{j,k}_i] | k = 0, \ldots, b\}$. These features are aggregated by set abstraction [Qi *et al.*, 2017] layers $\mathcal{A}(\cdot)$ with single-scale grouping to generate a feature vector $\hat{\mathcal{F}}^j_i = \mathcal{A}(\mathcal{F}^j_i)$.

In a tracklet, an object across consecutive frames usually shares some consistent spatial-temporal features. Especially for the rigid objects, their sizes are the same across frames.

To enhance their relation, we further aggregate the features $\tilde{\mathcal{F}}_i^j$ inside the proposal by employing a gated recurrent unit (GRU) layer $\mathcal{R}(\cdot)$ as

$$\tilde{\mathcal{F}}_i^j = \mathcal{R}(\hat{\mathcal{F}}_i^j, \tilde{\mathcal{F}}_{i-1}^j), \tag{4}$$

where $\tilde{\mathcal{F}}_{-n-1}^j$ are initialized by zeros.

Finally, we obtain a sequence of aggregated features for each j-th tracklet proposal, denoted by $\{\tilde{\mathcal{F}}_i^j\}_{i=-n}^0$, which are used to perform tracklet refinement.

Refinement Head

We refine a tracklet proposal T^j by refining its multiple 3D BBs $\{B_i^j\}_{i=-n}^0$. During training, we assign the ground-truth tracklet to the proposal T^j if the 3D overlap between the BBs in ground-truth tracklet and in proposal is larger than a threshold. For each B_i^j, we conduct box refinement by directly regressing the size, location, and orientation residuals relative to it using the aggregated features $\tilde{\mathcal{F}}_i^j$. For each tracklet proposal, we also compute a confidence using the features $\tilde{F}_0^j \in \{\tilde{\mathcal{F}}_i^j\}_{i=-n}^0$. The training target of the confidence is formulated as $min(1, max(0, 2mIoU - 0.5))$, where the mIoU is calculated by the mean 3D intersection between $\{B_i^j\}_{i=-n}^0$ and their ground-truth. The confidence loss is formulated to minimize the cross-entropy loss between confidence outputs and training targets.

In the inference stage, we perform NMS on the 3D BBs in the last frame to obtain a set of refined tracklets $\hat{\mathcal{T}}_t$, which encodes more accurate object sizes, motions, and correspondences during a short period of time.

3.3 Tracklet Association

Inspired by 2D image-based tracklet association which generates long trajectory by associating the tracklets with previous trajectory set using their 2D IOUs [Sun *et al.*, 2020], we transform this strategy into 3D trackelt association. We first initialize an empty trajectory set at the beginning of tracking. Then, at each timestamp t, we associate our refined tracklets $\hat{\mathcal{T}}_t$ with previous trajectories using a simple greedy matching algorithm based on their 3D IoUs. For successfully associated tracklets, we use them to update the corresponding trajectories. Unsuccessfully associated tracklets will initialize new trajectories for MOT at the next timestamp. This proposed method is able to track objects under both online and global settings. The tracklets encoded by multiple detections can recover missed objects across multiple frames.

3.4 Losses

The proposed PC-TCNN is trained in an end-to-end manner with a tracklet proposal loss L^{tpn} and a proposal refinement loss L^{trn}. Similar to [Shi *et al.*, 2020], we sum the two losses with equal weights to generate a final training loss as $L = L^{tpn} + L^{trn}$. L^{tpn} consists of multiple object losses L_i^{obj} and motion losses L_i^{mot} weighted by β,

$$L^{tpn} = \sum_{i=-n}^{0} L_i^{obj} + \beta \sum_{i=-n+1}^{0} L_i^{mot}. \tag{5}$$

We employ the object loss L_i^{obj} in i-th frame following single frame object detector [Shi *et al.*, 2020],

$$L_i^{obj} = \sum \mathcal{L}^{sl1}(\Delta\widehat{r^p}, \Delta r^p) + \eta L_i^{cls}, \tag{6}$$

where $r \in \{x_i, y_i, z_i, w_i, h_i, l_i, \alpha_i\}$; the smooth L1 loss $\mathcal{L}^{sl1}$ is used for single frame anchor box regression with the network output residual $\Delta\widehat{r^p}$ and regression target Δr^p; focal loss L^{cls} is employed for anchor classification. We also adopt the smooth L1 loss on the predicted motion residuals $\Delta\widehat{r^m}$ and motion targets Δr^m to regress the object motion as

$$L_i^{mot} = \sum_{r \in \{dx_i, dy_i\}} \mathcal{L}^{sl1}(\Delta\widehat{r^m}, \Delta r^m). \tag{7}$$

The L^{trn} loss consists of multiple object refinement residual losses and a confidence loss as

$$L^{trn} = \sum_{i=-n}^{0} \sum \mathcal{L}^{sl1}(\Delta\widehat{r^s}, \Delta r^s) + L^{mIoU}, \tag{8}$$

where $r \in \{x_i, y_i, z_i, w_i, h_i, l_i, \alpha_i\}$; the confidence loss L^{mIoU} is calculated by a cross-entropy loss with confidence outputs and confidence targets as detailed in Section 3.2; The residual loss is formulated by smooth L1 loss with the network output residuals $\Delta\widehat{r^s}$ and the target residuals $\Delta\widehat{r^s}$.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate our method on the KITTI tracking dataset [Geiger *et al.*, 2012], which consists of 21 training sequences and 29 test sequences. Input data of experiments include point clouds and IMU/GPS data. The images are only used to visualize tracking results. Same as most prior works, we report results on the car subset for comparison.

Since the evaluation of the KITTI benchmark is conducted on 2D image plane, we projected our 3D tracking results on image plane for evaluation. We adopted the widely used CLEAR MOT metrics [Bernardin and Stiefelhagen, 2008] (including MOTA, MOTP, IDS, and FRAG) and Mostly Tracked (MT) / Mostly Lost (ML) [Li *et al.*, 2009] metrics for MOT evaluation. We also report the results evaluated by the 3D MOT metrics proposed by [Weng *et al.*, 2020a].

4.2 Implementation Details

In our experiments, the 3D sparse convolution has four layers with feature dimensions (16, 32, 64, 64). Feature dimension of the ConvGRU and 2D convolution is 128 and 256, respectively. We set the tracking range within [0, 70.4]m for the X axis, [-40, 40]m for the Y axis, and [-3, 1]m for the Z axis. The input point clouds are voxelized along X, Y, Z with a size (0.05m, 0.05m, 0.1m), respectively. We employed the set abstraction layers with grouping scales 128, 32 and 1. During training, we performed NMS with a 3D IoU threshold of 0.8 to randomly keep 64 tracklet proposals with 1:1 negative and positive proposals. A tracklet proposal that has mean 3D IoU with ground-truth tracklet beyond 0.5 is treated as a positive proposal. Otherwise, it is treated as a negative proposal. During inference, we performed NMS with a threshold of 0.7 to

Frames	AP@t		MOT Metrics								Speed
	$2D$	$3D$	MOTA	MOTP	Recall	Precision	MT	ML	IDS	FRAG	(ms)
1	91.25%	91.16%	85.45%	**86.23%**	88.58%	97.17%	79.04%	1.81%	48	242	**75**
2	92.78%	93.12%	87.92%	86.14%	91.84%	97.10%	85.71%	2.29%	18	147	158
3	93.59%	94.18%	89.15%	86.19%	92.36%	97.95%	86.67%	**1.80%**	7	63	240
4	**94.03%**	94.48%	**89.44%**	86.10%	92.88%	**98.02%**	**88.10%**	**1.80%**	**2**	33	312
5	93.51%	**94.49%**	88.62%	86.16%	**92.93%**	96.67%	87.14%	2.29%	3	**31**	388

Table 1: Results on the validation set by using the different number of point cloud frames. AP@t refers to the average precision (IoU=0.5) of tracklet detections in the last frame. The $2D$ and $3D$ denote the image plane and LiDAR coordinate system, respectively. Values in bold highlight the best results.

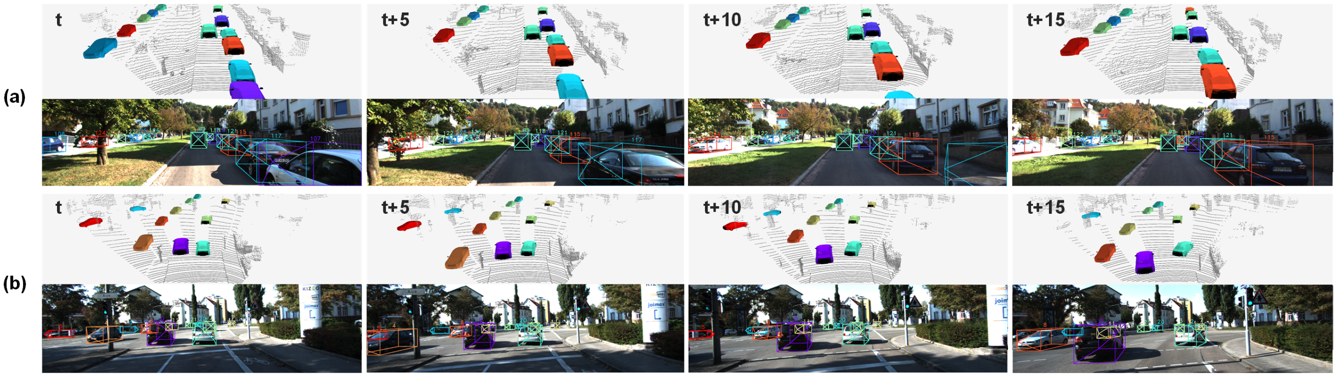

Figure 3: Qualitative results of our method on the sequence 11 (a) and 12 (b) of the KITTI test set. Different objects are in different colors.

Backbone Network	TR	TA	AP@t	MOTA
2D Conv+3D Conv	×	✓	90.21%	83.99%
3D SpConv+ConvGRU	×	✓	91.01%	85.82%
3D SpConv+ConvGRU	✓	×	92.30%	87.30%
3D SpConv+ConvGRU	✓	✓	**94.48%**	**89.44%**

Table 2: Ablation study results on the validation set using different components and 4 input frames. TR is the tracklet proposal refinement scheme; TA is the tracklet augmentation; AP@t refers to average precision of tracklet detection in the last frame.

keep top-100 proposals. Also, we used an NMS threshold of 0.1 to remove the redundant tracklets.

To avoid overfitting, we adopted a series of data augmentation on the registered point cloud frames. It includes random flipping along the X axis, global scaling with a random factor in [0.95,1.05], global rotation around the Z axis with a random angle in $[-\frac{\pi}{2}, \frac{\pi}{2}]$. We aslo performed data augmentation on tracklets by randomly sampling ground-truth tracklets from other training samples to current training samples to simulate objects' locations and motions in various environments. We adopted the ADAM optimizer with batch size 4, learning rate 0.01 and 80 epochs on two RTX 2080 Ti GPUs to train our PC-TCNN. We employed the onecycle learning rate strategy to deal with the learning rate decay.

4.3 Ablation Study

We conducted a number of experiments to examine each component/design of the proposed PC-TCNN. For the ablation study, we split the KITTI training set into two sub-datasets for training and validation. The sub training set consists of 11 sequences, and the validation set consists of 10 sequences.

Number of input frames. We conducted an ablation study on the number of input frames. The results are shown in Table 1. On the validation set, our method achieved the best MOT performance when taking four frames as inputs. Although the false negatives can be further reduced by taking more frames, the false positives dramatically increase with more background noise being introduced. The overall inference time also becomes longer when the number of input frame increases as more computation is required. Some strategy can speed up the network, such as enlarging the voxel size, but may lower tracking performance.

Backbone network. To investigate the settings of the backbone network, we compared our settings of 3D sparse convolution (SpConv)+ConvGRU with a baseline setting of 2D Conv+3D Conv used in FaF [Luo *et al.*, 2018]. The results are shown in Table 2. When both pipelines take in four frames, our method outperforms the baseline by 1.83% on MOTA, suggesting that the 3D SpConv+ConvGRU backbone is an essential setting in our method.

Tracklet augmentation. To analyze the effectiveness of the tracklet augmentation, we removed the tracklet augmentation and kept other components unchanged. The results are shown in Table 2. We can see that without tracklet augmentation, the MOT performance drops significantly (2.14%), indicating the benefit of tracklet augmentation.

Method	Input	**MOTA**	MOTP	Recall	Precision	MT	ML	IDS	FRAG
mono3DT[Hu *et al.*, 2019]	2D	84.52%	85.64%	88.81%	97.95%	73.38%	2.77%	377	847
TuSimple[Choi, 2015]	2D	86.62%	83.97%	90.50%	97.99%	72.46%	6.77%	293	501
CenterTrack[Zhou *et al.*, 2020]	2D	89.44%	85.05%	93.20%	97.73%	82.31%	**2.31%**	116	334
ODESA[Mykheievskyi *et al.*, 2020]	2D	90.03%	84.32%	92.62%	**98.77%**	82.62%	**2.31%**	90	501
GNN3DMOT[Weng *et al.*, 2020b]	2D+3D	82.24%	84.05%	-	-	64.92%	6.00%	142	416
aUToTrack[Burnett *et al.*, 2019]	2D+3D	82.25%	80.52%	89.36%	97.03%	56.77%	7.38%	1025	1402
mmMOT[Zhang *et al.*, 2019]	2D+3D	84.77%	85.21%	88.81%	97.93%	73.23%	2.77%	284	753
JRMOT[Shenoi *et al.*, 2020]	2D+3D	85.70%	85.48%	89.51%	97.81%	71.85%	4.00%	98	372
PointTrackNet[Wang *et al.*, 2020]	3D	68.24%	76.57%	83.56%	88.10%	60.62%	12.31%	111	725
ComplexerYOLO[Simon *et al.*, 2019]	3D	75.70%	78.46%	85.32%	95.18%	58.00%	5.08%	1186	2092
AB3DMOT[Weng *et al.*, 2020a]	3D	83.84%	85.24%	88.32%	96.98%	66.92%	11.38%	**9**	224
PC-TCNN (Ours)	3D	**91.75%**	**86.17%**	**96.08%**	96.45%	**87.54%**	2.92%	26	**118**

Table 3: Car tracking results on the test set of the KITTI tracking benchmark. The 2D and 3D denote 2D images and 3D point clouds, respectively.

Tracklet proposal refinement scheme. We further investigated the effectiveness of the tracklet proposal refinement scheme. As shown in Table 2, with tracklet refinement, the average precision of tracklet detection in the last frame gains a 3.47%, demonstrating that our tracklet proposal refinement improves the tracklet accuracy. As shown in the same table, the tracklet proposal refinement scheme gains a 3.62% on MOTA. This demonstrates the proposed tracklet refinement can further improves MOT on point cloud sequences.

4.4 Results on KITTI Tracking Benchmark

The car tracking results on the leaderboard of the KITTI tracking benchmark are summarized in Table 3 (only recently published methods are reported). The PC-TCNN takes four point cloud frames as inputs, and is trained on 20 sequences of the KITTI training set and validated on the remained one sequence. Our method, achieves an MOTA of 91.75%, currently ranks 1^{st} among all submitted methods as of Jan. 20th, 2021. Our method also achieves a high recall of 96.08%, outperforms the strongest prior model by 2.88%. The outstanding performance is due to that the tracklet proposal design greatly enhances the tracklet accuracy by better modeling spatial-temporal features of point cloud sequences. These tracklets, encoded by accurate detections and motions, significantly reduce the false negatives, ID switches, and fragments in the point cloud-based MOT.

We also show our qualitative results on two sequences of the KITTI test set in Figure 3. For better visualization, we rendered the 3D BBs using CAD car models. We also showed a long-time tracking example in Figure 4, in which the object with ID 2 is tracked over 340 meters, although it is heavily occluded by the object ID 101 in many frames.

4.5 Evaluating in 3D Space

For a broader comparison with 3D MOT methods, we also conducted experiments using the data split as GNN3DMOT [Weng *et al.*, 2020b]. The results on the validation set are shown in Table 4, where the sAMOTA, AMOTA, IDS and FRAG are 3D MOT metrics proposed by AB3DMOT [Weng *et al.*, 2020a]. Our method outperforms the GNN3DMOT 1.76% in the most important sAMOTA metric, and also surpasses the FANTrack [Baser *et al.*, 2019], AB3DMOT [Weng *et al.*, 2020a] and GNN3DMOT [Weng *et al.*, 2020b] in the AMOTA and FRAG metrics, further demonstrating the effectiveness of our method.

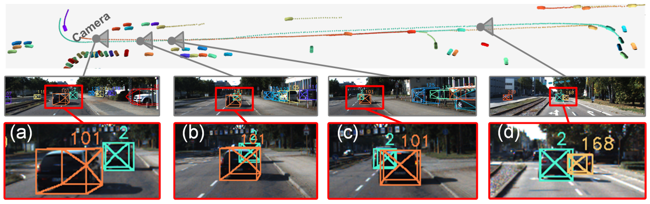

Figure 4: A long-time tracking example: Object ID 2 (cyan box) is tracked by our method over 340 meters, although it is heavily occluded by the object ID 101 in many frames (b, c).

Method	sAMOTA	AMOTA	IDS	FRAG
FANTrack	82.97%	40.03%	35	202
AB3DMOT	91.78%	44.26%	**0**	15
GNN3DMOT	93.68%	45.27%	**0**	10
Ours	**95.44%**	**47.64%**	1	**9**

Table 4: Evaluation results on the validation set. The evaluation is conducted in 3D space using 3D MOT metrics.

5 Conclusion

We present a novel tracklet proposal PC-TCNN network for MOT on point clouds. By innovatively employing a proposal-based framework, our PC-TCNN performs tracklet proposals generation, refinement, and association, leading to significantly improved MOT performance. Meanwhile, we introduced a novel tracklet proposal refinement scheme that extracts and aggregates spatial-temporal features of each proposal to refine the tracklet detections, and this greatly enhances the accuracy of tracklets. With these novel designs, our network achieved better MOT performance over the state-of-the-art methods.

References

[Bae and Yoon, 2014] Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, pages 1218–1225, 2014.

[Baser *et al.*, 2019] Erkan Baser, Venkateshwaran Balasubramanian, Prarthana Bhattacharyya, and Krzysztof Czarnecki. Fantrack: 3d multi-object tracking with feature association network. In *IV*, pages 1426–1433, 2019.

[Bernardin and Stiefelhagen, 2008] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[Burnett *et al.*, 2019] Keenan Burnett, Sepehr Samavi, Steven Waslander, Timothy Barfoot, and Angela Schoellig. autotrack: A lightweight object detection and tracking system for the sae autodrive challenge. In *CRV*, pages 209–216, 2019.

[Choi, 2015] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *CVPR*, pages 3029–3037, 2015.

[Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[Hou *et al.*, 2017] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *CVPR*, pages 5822–5831, 2017.

[Hu *et al.*, 2019] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *CVPR*, pages 5390–5399, 2019.

[Li *et al.*, 2009] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960, 2009.

[Li *et al.*, 2019] Siyuan Li, Zhi Zhang, Ziyu Liu, Anna Wang, Linglong Qiu, and Feng Du. Tlpg-tracker: Joint learning of target localization and proposal generation for visual tracking. In *IJCAI*, pages 708–715, 2019.

[Luo *et al.*, 2018] 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 *CVPR*, pages 3569–3577, 2018.

[Mykheievskyi *et al.*, 2020] Dmytro Mykheievskyi, Dmytro Borysenko, and Viktor Porokhonskyy. Learning local feature descriptors for multiple object tracking. In *ACCV*, pages 558–575, 2020.

[Qi *et al.*, 2017] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, page 5099–5108, 2017.

[Shenoi *et al.*, 2020] Abhijeet Shenoi, Mihir Patel, JunYoung Gwak, Patrick Goebel, Amir Sadeghian, Hamid Rezatofighi, Roberto Martín-Martín, and Silvio Savarese. Jrmot: A real-time 3d multi-object tracker and a new large-scale dataset. In *IROS*, pages 10335–10342, 2020.

[Shi *et al.*, 2019] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019.

[Shi *et al.*, 2020] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, pages 10529–10538, 2020.

[Simon *et al.*, 2019] Martin Simon, Karl Amende, Andrea Kraus, Jens Honer, Timo Samann, Hauke Kaulbersch, Stefan Milz, and Horst Michael Gross. Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In *CVPR Workshops*, pages 1190–1199, 2019.

[Sun *et al.*, 2020] ShiJie Sun, Naveed Akhtar, XiangYu Song, HuanSheng Song, Ajmal Mian, and Mubarak Shah. Simultaneous detection and tracking with motion modelling for multiple object tracking. In *ECCV*, pages 626–643, 2020.

[Tokmakov *et al.*, 2017] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, pages 4481–4490, 2017.

[Wang *et al.*, 2020] Sukai Wang, Yuxiang Sun, Chengju Liu, and Ming Liu. Pointtracknet: An end-to-end network for 3-d object detection and tracking from point clouds. *IEEE Robotics and Automation Letters*, 5(2):3206–3212, 2020.

[Weng *et al.*, 2020a] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *IROS*, pages 10359–10366, 2020.

[Weng *et al.*, 2020b] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *CVPR*, pages 6499–6508, 2020.

[Wu *et al.*, 2020] Haoze Wu, Jiawei Liu, Xierong Zhu, Meng Wang, and Zheng-Jun Zha. Multi-scale spatial-temporal integration convolutional tube for human action recognition. In *IJCAI*, pages 753–759, 2020.

[Zhang *et al.*, 2019] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Loy Chen Change. Robust multi-modality multi-object tracking. In *ICCV*, pages 2365–2374, 2019.

[Zhang *et al.*, 2020] Yang Zhang, Hao Sheng, Yubin Wu, Shuai Wang, Weifeng Lyu, Wei Ke, and Zhang Xiong. Long-term tracking with deep tracklet association. *IEEE Transactions on Image Processing*, 29:6694–6706, 2020.

[Zhou *et al.*, 2020] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490, 2020.