3D Single-Object Tracking with Spatial-Temporal Data Association

Yongchang Zhang[1,3], Hanbing Niu[2], Yue Guo[1], and Wenhao He[1,3]✉

***Abstract*— This paper proposes a novel 3D single-object tracker to more stably, accurately, and faster track objects, even if they are temporarily missed. Our idea is to utilize spatial-temporal data association to achieve object tracking robustly, and it consists of two main parts. We firstly employ a temporal motion model cross frames to estimate the object's temporal information and update the region of interest(ROI). The advanced detector only focuses on ROI rather than the whole scene to generate the spatial position. Second, we introduce a new pairwise evaluation system to exploit spatial-temporal data association in point clouds. The proposed evaluation system considers detection confidence, orientation offset, and objects distance to more stably achieve object matching. Then, we update the predicted state based on the pairwise spatial-temporal data. Finally, we utilize the previous trajectory to enhance the accuracy of static tracking in the refinement scheme. Experiments on the KITTI and nuScenes tracking datasets demonstrate that our method outperforms other state-of-the-art methods by a large margin (a 10% improvement and 280 FPS on a single NVIDIA 1080Ti GPU). Compared with multi-object tracking, our tracker also has superiority.**

I. Introduction

Object tracking, a crucial technology for extracting dynamic information from surroundings, has wide applications in mobile robotics and autonomous driving. Exiting methods are mainly divided into single-object tracking(SOT) and multi-object tracking(MOT). Multi-object tracking methods focus on the data association between detected objects in different frames, and they will stop tracking when the object is not detected. But single-object tracking methods emphasize the continuity of tracking for single objects. We have to predict the position of the object even if the tracked object is missed.

Due to the illumination interference and camera defects, 2D visual tracking based on RGB cameras usually limits the practical applications. In the field of autonomous driving and mobile robots, 3D object tracking methods outperform 2D counterparts. The main reason is that point clouds generated by LIDAR have more accurate distance information and can be widely used in 3D object tracking to follow targets continuously. Meanwhile, with the advantage of laser scanners, methods using point clouds can overcome the illumination change effectively. Hence, we only focus on 3D object tracking in point clouds.

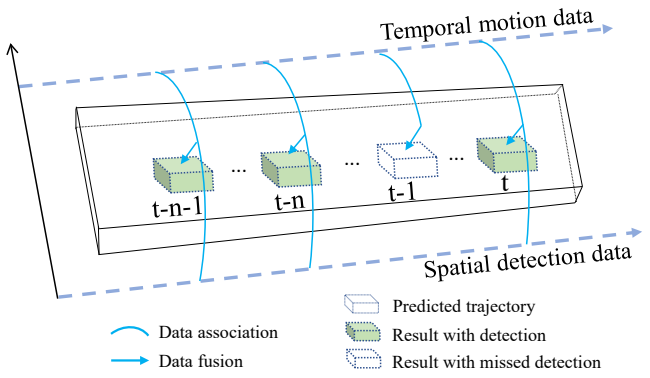

Fig. 1. Exemplified illustration to show how our tracker works, which consists of temporal motion data, spatial detection data, data association and fusion.

Continuous tracking of the heavily occluded or distant objects in raw point clouds remains challenging. Some methods [1] project point clouds into a planar space(*e.g.*, bird's eye view) and inherit 2D works to predict 3D box. Others [2], [3] follow a tracking-by-detection framework and utilize the template similarity to determine the final box. Both of the above methods fail when the initial template in the first frame is too sparse and hence yielded little target information.

Another difficulty in 3D object tracking is taking full advantage of spatial-temporal information. An object across consecutive frames shares some spatial-temporal consistencies. Previous tracking methods [3], [4] only utilize the spatial features to track objects and ignore the temporal features(*e.g.*, velocity and acceleration of an object) in the continuous three-dimensional space. So they always cause significant deviations when object appearances change a lot.

To tackle the above-mentioned difficulties, we introduce spatial-temporal data association into 3D single-object tracking. Compared with the data association in 3D multi-object tracking, our tracker mainly associates the temporal motion data and spatial detection data in the same frame. Based on these, our method can robustly track the objects even if they are distant or occluded. Specifically, we propose a novel 3D single-object tracker, based on spatial-temporal data association, in point clouds to more stably, accurately, and faster track objects. Compared with simple template matching or simple object detection, we turn to address 3D single-object tracking by associating the temporal motion data and spatial detection data. An illustration is shown in

*This work is supported by National Key R&D Program of China (2018YFB1306302, 2018YFB1306300, and 2018YFB1306500).

[1] Yongchang Zhang, Yue Guo, and Wenhao He are with Institute of Automation, Chinese Academy of Sciences, Beijing, China. zhangyongchang2020,guoyue2013@ia.ac.cn

[2] Hanbing Niu is with University of Electronic Science and Technology of China, Chengdu, China.

[3] Yongchang Zhang, and Wenhao He are also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. wenhao.he@ia.ac.cn

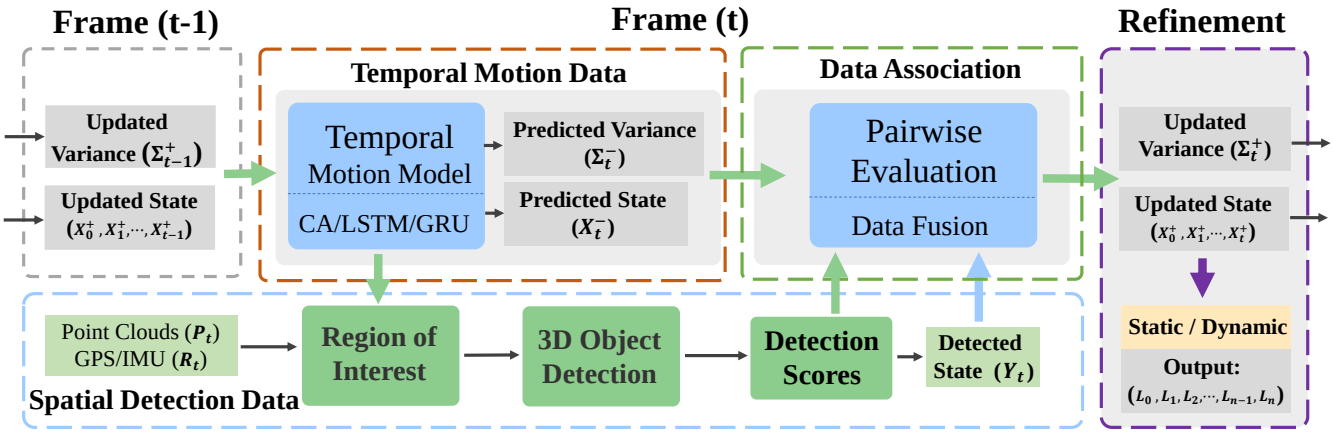

Fig. 2. The main framework of our tracker. The temporal motion data comes from the temporal model based on previous state, and the spatial detection data outputs from advanced 3D object detector. Then, we build a pairwise evaluation system to achieve data association between temporal data and spatial data and fuse the most like object and temporal data to the updated result. We finally refine the updated results based on the previous trajectory.

Figure 1. We first use temporal motion data to estimate the object's position and determine the region of interest. Second, we feed the ROI into an advanced detector to generate the spatial detection data. Then, a new pairwise evaluation system is built to achieve temporal-spatial data association, and we fuse the associated output and motion position to generate the predicted result. Finally, we utilize the previous trajectory to enhance the accuracy of static tracking in the refinement scheme.

Experiments on KITTI [5] and nuScenes [6] tracking dataset demonstrate that our STRNet significantly outperforms state-of-the-art methods(about a 10% improvement and 280 FPS on a single NVIDIA 1080Ti GPU). Compared with MOT, our tracker also has superiority.

Overall, the main contributions of this paper include:

- We propose a novel tracker that associates the spatial-temporal data to track the missed object robustly.
- We build a new pairwise evaluation system to exploit data association about motion model and 3D detector.
- We utilize historical tracking information to update the region of interest, which achieves faster tracking.
- We present a position refinement scheme that uses the previous trajectory to enhance the tracking accuracy.

II. Related Work

Our work takes insights from 3D single-object tracking and data association in 3D multi-object tracking.

A. 3D object detection in Point clouds

The tracking-by-detection framework is popular in most 3D object tracking methods. Plenty of ideas from object detection advance the 3D objecting tracking. We first focus on the 3D object detection. Traditionally, 3D detectors are categorized into two types: *1) single-stage detectors*, such as [7], [8], [9], [10], regress bounding boxes directly from features without proposals. VoxelNet [11] extract features from point clouds by Voxel layers. TANet [12] considers feature-wise relation in the feature extraction. PointPillar [13] divides a point cloud into pillars for efficient object detection. *2) two-stage detectors*, including [14], [15], [16], [17], use region-proposal-aligned features to regress results. PointRCNN [18] proposes a region proposals network to refine the detection of PointNet [19]. PV-RCNN [20] combines both point-based and voxel-based networks to extract features from raw point clouds.

B. 3D object tracking in Point clouds.

3D object tracking methods localize objects over a continuous period of time. Early methods generate object proposals on projected point clouds by 2D experience, such as the bird's-eye views [21], or foreground images [22], [23]. These works introduce errors into the 3D boxes because of losing the depth information on plane. Since the above issues limit some real-world applications, SC3D [4] and P2B [3] addressed such concerns from a pure geometric perspective. SC3D [4] executes 3D template matching randomly to generate bunches of 3D object proposals. P2B [3] localizes potential object centers in a 3D search area embedded with target information. However, they ignored the temporal features(e.g., velocity and acceleration) and fail when spatial features disappear.

C. Data association in 3D multi-object tracking.

Data association is a key part in 3D multi-object tracking, and it mainly is used for object matching in consecutive frames. Previous 3D MOT methods [2], [24] achieved data association based on the distance or overlap between detected bounding boxes(BBs) in adjacent frames. PC3T [25] and PC-TCNN [26] adds the geometric affinity, appearance offset and motion cost to build the data association matrix and utilizes the greedy algorithm to associate the same object. Different from the data association for objects in consecutive frames, our tracker mainly associates the temporal motion data and spatial detection data in the same frame. Meanwhile, we also consider the considers the orientation offset and detection score in our pairwise evaluation system. The orientation

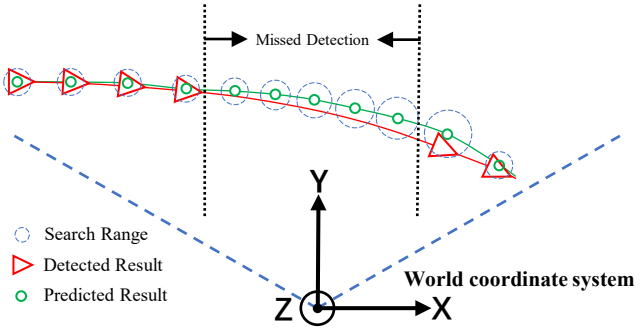

Fig. 3. Illustration of RoI. RoI is the search range of 3D detector. The detected result comes from the 3D detector, and the predict result outputs from motion model.

offset part is used to distinguish objects that are in the opposite direction, while the detection score is used to measure the degree of trust for the detector.

III. Proposed Method

With spatial-temporal data association, our method can systematically track a 3D object in practical applications. Figure 2 illustrates the main framework of our tracker. The temporal motion data comes from the temporal model based on the previous state, and the spatial detection data outputs from an advanced 3D object detector. Note that the detector only focuses on the region of interest to improve efficiency. Then, we build a pairwise evaluation system to achieve data association between temporal data and spatial data and fuse the most like object and temporal data to the updated result. We finally refine the updated results based on the previous trajectory.

A. Temporal motion model

Apart from the tracking-by-detection framework, we introduce a temporal motion model to estimate the object's state in every frame. The motion state can be represented as $X = [x, y, z, \theta, v^x, v^y, v^z, v^\theta, a^x, a^y, a^z, a^\theta, l, w, h]^T$ from BBs. Here, $\{x, y, z\}$ are the predicted position of the box; $\{l, w, h\}$ are length, width and height of the predicted box; $\{\theta\}$ is the angle of yaw; $\{v, a\}$ represent the velocity and acceleration. Note that X_t^- and X_t^+ represent the predicted state and updated state at the t frame, respectively.

Our temporal model is flexible and changeable and we can easily replace it with the advanced motion model. In this article, we have chosen three common models, including constant acceleration, LSTM [27] and GRU [26], to estimate the object's temporal motion state. We analyze these motion model in Ablation Study.

B. Determining of the region of interest

Instead of the whole scene, we only detect the object in the region of interest(RoI) to speed up our tracking. At frame t, the RoI is determined by predicted position in the current motion state X_t^-. We choose a two-meter range centered on the predicted position as our RoI. Note that we update the RoI with the current motion state.

When we determine the RoI, the predicted position is not always accurate, especially for the objects missed by the detector in many consecutive frames. To tackle this problem, we enlarge the RoI based on the number of consecutive missed frames(see Figure 3). This strategy is formulated by:

$$D_t = \begin{cases} D_0 + c \cdot N_{miss}, & if\ Y_{t-1}\ is\ not\ detected \\ D_0, & otherwise \end{cases} \tag{1}$$

Where D_0 represents the initial radius of the region of interest in $X-Y$ plane; c is a constant, and N_{miss} means the number of consecutive missed frames by 3D detector.

C. Pairwise evaluation system

Spatial 3D detector. Any advanced 3D detector can be added to our tracker. And we have chosen two common detectors(PointRCNN [18] and PV-RCNN [20]) to extract spatial detection data in RoI. The detection results include the boxes $\{Y_t^j\}_{j=1}^M$ and corresponding scores $\{Score_t^j\}_{j=0}^M$.

Spatial-temporal data association. To choose the most like object in $\{Y_j\}_{j=1}^M$, we exploit a new evaluation system that satisfy the requirement in accuracy and speed. For temporal motion data X_t^- and any of spatial detection data $\{Y_t^j\}_{j=1}^M = [x_t, y_t, z_t, l_t, w_t, h_t, \theta_t]^T$, we formulate the pairwise confidence as:

$$Conf_t^j = Score_t^j \cdot box_t^j, \quad j = 0, 1, \cdots, M \tag{2}$$

Where $Conf_t^j$ means the pairwise confidence, and box_t^j is the box confidence and defined by:

$$\begin{aligned} box_t^j = {} & \lambda_{dis} \cdot N\left(\left\| p_{Y_t^j} - p_{X_t^-} \right\|\right) \\ & + \lambda_\theta \cdot N\left(1 - \cos(\theta_{Y_t^j} - \theta_{X_t^-})\right) \\ & + \lambda_{IoU} \cdot N\left(1 - IoU_{X_t^-}^{Y_t^j}\right) \end{aligned} \tag{3}$$

Where λ_{dis}, λ_θ and λ_{IoU} are weights; $N(\cdot)$ denotes the Gaussian function; p is 3D position$\{x, y, z\}$; θ is the orientation, and $IoU_{X_t^{minu}}^{Y_t^j}$ means the 3D IoU between X_t^{minu} and Y_t^j. Note that we choose Y_t with the highest pairwise confidence as association result. For failed detection, we only predict the object's position based on the motion model.

State update. After spatial-temporal data association, we obtain the spatial detection data Y_t from data association and temporal motion data X_t^-. We fuse them by Kalman Filtering to get the update state X_t^+.

D. Tracking refinement

During tracking, we record the predicted trajectory and positions $\{p_j\}_{j=t-m}^t$ continuously. If $\{p_j\}_{j=t-m}^t$ are distributed in a certain range(*e.g.*, a ball with radius d), We consider that the object is static during this period period.

Once the predicted object is static, we adopt the Parzen-window Density Estimates [28] in the points set $\{p_j\}_{j=t-m}^t$ to refine the output, which can effectively reduce the random noise for position estimation.

$$f(p) = \frac{1}{mV} \sum_{j=t-m}^{m} \varphi\left(\frac{|p - p_j|}{h}\right) \tag{4}$$

TABLE I

EXTENSIVE COMPARISONS ON **KITTI** AND **NUSCENES** DATASET. NOTE THAT METHODS WITH * COME FROM MULTI-OBJECT TRACKING.

	Method	Car	Pedestrian	Cyclist	Mean	Car	Pedestrian	Bicycle	Overall
Precision	SC3D-EX	24.8	14.2	14.8	20.4	12.3	7.9	15.3	14.1
	SC3D-KF	57.9	37.8	70.4	48.5	21.9	12.7	34.7	20.2
	P2B	72.8	49.6	44.7	60.0	43.2	52.2	32.5	45.1
	PC3T*	73.5	57.1	65.8	68.1	-	-	-	-
	PC-TCNN*	74.1	56.3	67.4	68.4	-	-	-	-
	Ours	**75.1**	**61.3**	**72.2**	**71.5**	**46.9**	**55.3**	**36.7**	**49.0**
Success	SC3D-EX	21.2	8.1	11.1	15.7	14.5	7.4	13.6	16.4
	SC3D-KF	41.3	18.2	41.5	31.2	22.3	11.3	35.4	20,7
	P2B	56.2	28.7	32.1	42.4	38.8	28.4	32.8	36.5
	PC3T*	62.4	40.2	54.5	52.3	-	-	-	-
	PC-TCNN*	64.7	41.8	57.1	53.5	-	-	-	-
	Ours	**66.4**	**45.8**	**59.2**	**57.5**	**45.2**	**37.1**	**42.3**	**46.6**

TABLE II

EXTENSIVE EXPERIMENTS IN CAR ON KITTI WITH DIFFERENT POINTS NUMBER.

	Method	**Points $\geq$ 0**	**Points $\geq$ 10**	**Points $\geq$ 20**	**Points $\geq$ 50**	**Points $\geq$ 100**
	Frame Number	6424	4873	3206	1789	1511
Success	SC3D-EX	21.2	21.4	23.1	23.6	24.3
	SC3D-KF	41.3	42.2	45.8	46.5	47.1
	P2B	56.2	58.4	69.3	70.7	71.1
	Ours	**66.4**	**68.0**	**80.6**	**84.4**	**85.7**
Precision	SC3D-EX	24.8	25.2	28.4	29.1	29.9
	SC3D-KF	57.9	58.4	62.1	62.6	63.3
	P2B	72.8	73.5	85.6	86.4	87.2
	Ours	**75.1**	**76.8**	**90.4**	**92.5**	**93.9**

Where m is the number of points; h and V means the side length and volume of a small cube, respectively. $\varphi(\cdot)$ is the Gaussian window function and judges whether the position p locates in the cube. The position with the highest estimated value $f(p_t)$ as our final p_t. Note that we also update the X_t^+ based on p_t in static object tracking.

IV. EXPERIMENTS

In the experiments, we compare our tracker with current state-of-the-art methods, including SOT and MOT, on KITTI and nuScenes datasets. For fair comparisons, all settings in sample generation and evaluation metrics are the same.

A. Experimental setting

Datasets. The KITTI consists of 21 outdoor scenes and 8 types of objects. And the nuScenes contains 1000 scenes and the point clouds are captured by a 32-beam LiDAR. they are quite authoritative datasets for 3D object detection and tracking.

Parameters. The initial radius of search range R_0 is 2 meters, and the incremental constant c is 1.5 meter. In the pairwise confidence evaluation, the weights λ_{dis}, λ_{θ} and λ_{IoU} are 1.5, 1 and 2, respectively. To find static objects, the threshold d is 0.5 meter, and the side length h for Parzen-window Density Estimates is 0.1 meters. The mean μ and variance σ in the Gaussian function are 0 and 1.

Pretreatment. In our experiments, we transform all the point clouds and labels into a world coordinate system with the IMU/GPS data. We choose the PV-RCNN [20] as our detector.

Evaluation In 3D object tracking, One Pass Evaluation (OPE) is a widely-used metric that contains Success and Precision. Specifically, we define IoU between the output box and the ground truth as Success. Then we estimate Precision by AUC about the 3D distance between centers of the above boxes from 0 to 2 meters.

B. Comprehensive comparisons

Compare with SOT methods. Similar to SC3D [4] and P2B [3], our tracker only uses raw point clouds for 3D object tracking. We compare them on 19-20 scenes in KITTI and 150 scenes in nuScenes. As shown in Table I, our tracker significantly outperforms state-of-the-art methods by a large margin(above 10% increase on Success and Precision) in all categories. In SC3D, the approximate exhaustive search and poor feature extraction limit the speed and accuracy, and P2B addresses these weaknesses using VoteNet and improves Success and Precision. But they ignore the temporal motion information and lack proper data association during tracking. Our tracker effectively associates the spatial-temporal data and refines the results for robust tracking.

To compare deeply, we divide test data by the number of object points at the first frame, including Points $\geq$ 10, Points $\geq$ 20, Points $\geq$ 50, Points $\geq$ 100. For 3D object tracking, we select the object at the first frame. In real tracking applications, we always choose objects with richer information(*e.g.*, more points in point clouds). We comprehensively compare our tracker with SC3D and P2B in these divided data. Results (4sh - 7sh column in Table II) demonstrate the superiority

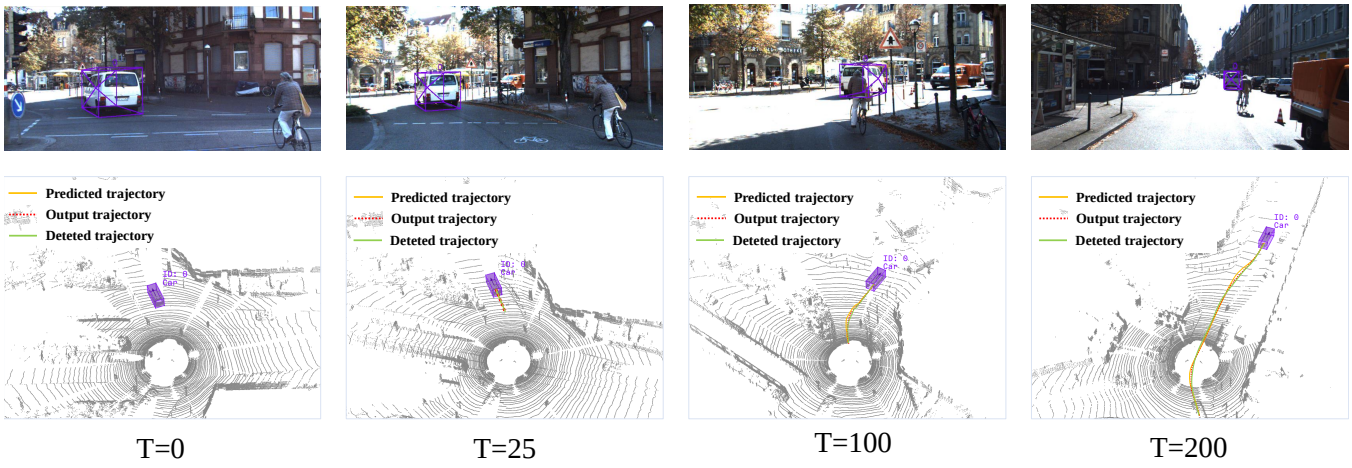

Fig. 4. Examples of 3D object tracking results. the first and second rows show the tracked object in the image and point clouds, respectively. The output result in the current frame is marked with a purple bounding box in the LiDAR coordinate system. Its updated trajectory is denoted with a yellow curve, and the output trajectory is marked with red points. Images are only used for better visualization in our work.

TABLE III
SPEED OF OUR TRACKER, P2B, SC3D AND PC3T.

Method	SC3D	P2B	PC3T	PC-TCNN	**Ours**
Speed(fps)	1.4	40	240	190	280

of our tracker for different objects.

Compare with MOT methods. The PC3T [25] and PC-TCNN [26] are well-known trackers in MOT, which have state-of-the-art performance on KITTI and nuScenes datasets. We test them on SOT and compare them with our tracker. The results in Table I show that our tracker also has superiority(2% improvement on Precision and 4% on Success). Two parts make our tracker more advantageous. One involves the pairwise evaluation, which more comprehensively considers the factors((*e.g.*, orientation offsets, and detection scores) in data association. And another is about the refinement scheme. The updated result is refined by the previous trajectory based on the density estimates, which improves our tracker a lot.

Speed. To measure the speed of the involved methods, we also average the running times on the test dataset. Our tracker can achieve 280 FPS on a single NVIDIA 1080Ti GPU, while P2B only runs with 40 FPS and SC3D in default setting runs with 1.8 FPS on the same platform(see Table III). Our tracker only focuses on the RoI rather than the whole scene, which significant speeds up our tracker. We run the PC3T and PC-TCNN with the same RoI strategy and our tracker is faster.

C. Ablation Study

Motion model analysis. To better estimate the object's state in every frame, we analyze three motion predictive methods, including traditional constant acceleration(CA), net-based LSTM and GRU, on Car category in KITTI datasets.

TABLE IV
ANALYSIS FOR DIFFERENT TEMPORAL MOTION MODELS

Method	**Success**	**Precision**	**FPS**
With CA	**66.4**	**75.1**	**280**
With LSTM	64.9	73.9	147
With GRU	65.2	74.3	196

TABLE V
ABLATION STUDIES FOR OUR TRACKER ON KITI

Method	**Success**	**Precision**
Our default setting	**66.4**	**75.1**
Without CA	57.5	71.2
Without PCE	63.0	72.0
Without refinement	64.4	72.9

The results show that our tracker with the traditional CA model has a better performance in testing(see Table IV). Using net-based methods needs a lot of training and parameter setting, and it also reduces the efficiency of the tracker. So our default settings are based on the CA model.

Effectiveness of temporal motion model. To investigate the effectiveness of the temporal motion model, we delete the constant acceleration model and only use the spatial detection data for 3D object tracking, while other components are unchanged. As shown in Table V (Without CA), the CA model contributes about 8.9% improvement on Success, and 3.9% improvement on Precision. The advantage of using the temporal motion model is to track the undetected object by state estimation. Results demonstrate the reasoning module with CAM plays an essential role, especially for Success, in our tracker.

Effectiveness of pairwise evaluation system. To study the efficiency of the well-designed pairwise confidence evaluation system, we only use the distance information to judge output results. As shown in Table V (Without PCE), the confidence-guided evaluation system achieve a performance

improvement of 3.4% and 2.1% on Success and Precision, respectively, which is a key part for our spatial-temporal data association. Results demonstrate that the well-designed confidence evaluation system can help better locate objects at the current frame.

Effectiveness of tracking refinement scheme. To verify the effectiveness of the refinement module, we remove it from the default setting. As shown in Table V (Without refinement), with the refinement, our tracker increases 2% and 2.2% on Success and Precision in nuScenes datasets. The refinement scheme can help our tracker better optimize the trajectory, and the results demonstrate the effectiveness of our refinement scheme.

V. Conclusions

Dealing with failed detection and incorrect matching in 3D object tracking remains challenging. This paper proposes a novel 3D single-object tracker to more stably, accurately, and faster track objects, even if they are missed. Our main idea is to predict the position by spatial-temporal data association and refine it with the previous trajectory. Specifically, we build a new pairwise evaluation system that comprehensively analyzes the difference between temporal motion data and spatial detected data. Then we utilize historical tracking information to update the region of interest to speed up our tracking. Finally, we present an effective refinement scheme to enhance the tracking accuracy with the previous trajectory. Experiments on the KITTI and nuScenes tracking datasets demonstrate that our tracker outperforms other state-of-the-art methods by a large margin and further proves the effectiveness for spatial-temporal data association in 3D object tracking.

References

[1] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7652–7660.

[2] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 359–10 366.

[3] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, "P2b: Point-to-box network for 3d object tracking in point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6328–6337.

[4] S. Giancola, J. Zarzar, and B. Ghanem, "Leveraging shape completion for 3d siamese tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1359–1368.

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

[6] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

[7] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[8] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1708–1716.

[9] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 037–11 045.

[10] N. Zhao, T.-S. Chua, and G. H. Lee, "Sess: Self-ensembling semi-supervised 3d object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 076–11 084.

[11] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.

[12] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "Tanet: Robust 3d object detection from point clouds with triple attention," *AAAI Conference on Artificial Intelligence*, 2020. [Online]. Available: https://arxiv.org/pdf/1912.05163.pdf

[13] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 689–12 697.

[14] 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. 43, no. 8, pp. 2647–2664, 2021.

[15] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Std: Sparse-to-dense 3d object detector for point cloud," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1951–1960.

[16] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6526–6534.

[17] S. Pang, D. Morris, and H. Radha, "Clocs: Camera-lidar object candidates fusion for 3d object detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 386–10 393.

[18] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 770–779.

[19] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[20] 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," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 526–10 535.

[21] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3569–3577.

[22] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7652–7660.

[23] U. Kart, A. Lukežič, M. Kristan, J.-K. Kämäräinen, and J. Matas, "Object tracking by reconstruction with view-specific discriminative correlation filters," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1339–1348.

[24] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 433–440.

[25] H. Wu, W. Han, C. Wen, X. Li, and C. Wang, "3d multi-object tracking in point clouds based on prediction confidence-guided data association," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.

[26] H. Wu, Q. Li, C. Wen, X. Li, X. Fan, and C. Wang, "Tracklet proposal network for multi-object tracking on point clouds," in *IJCAI*, 2021.

[27] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Kraehenbuehl, T. Darrell, and F. Yu, "Joint monocular 3d vehicle detection and tracking," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5389–5398.

[28] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning, second edition*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2018.