3D Multi-object Detection and Tracking with Sparse Stationary LiDAR

Meng Zhang, Zhiyu Pan, Jianjiang Feng(✉), and Jie Zhou

Department of Automation, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
{zhangm20,pzy20}@mails.tsinghua.edu.cn
{jfeng,jzhou}@tsinghua.edu.cn

Abstract. The advent of low-cost LiDAR in recent years makes it feasible for LiDAR to be used in visual surveillance applications such as detection and tracking of players in a football game. However, the extreme sparsity of point cloud acquired by such LiDAR is a challenge for object detection and tracking in large-scale scenes. To alleviate this problem, we propose a method of multi-object detection and tracking from sparse point clouds comprising a short-term tracklet regression stage and a 3D D-IoU data association stage. In the former stage, temporal information is aggregated by the proposed temporal fusion module to predict short-term tracklets formed by three bounding boxes. In the latter stage, the Distance-IoU scores of current tracklets and historical trajectories are computed to associate the data using Hungarian matching algorithm. To reduce the cost of manual annotations, we build a simulated point cloud dataset using Google Research Football for training. A real test dataset of football game is acquired by Livox Mid-100 LiDAR. Our experimental results on both datasets show that fusing multi-frames conduces to improving detection and tracking performance from sparse point clouds. Our 3D D-IoU tracking method also gets a promising performance on the nuScenes autonomous driving dataset.

Keywords: 3D detection · Multi-object tracking · Sparse Stationary LiDAR

1 Introduction

3D multi-object detection and tracking play an important role in the visual surveillance applications, such as football player tracking [11]. These visual surveillance systems mainly utilize RGB cameras for acquisition until now, which leads to some challenging problems: poor performance under non-ideal light and weather conditions, insufficient ability to distinguish foreground and background, and inaccurate object location estimations in 3D space. These challenges can be eliminated by using high-resolution LiDAR to capture point cloud of the scene, yet which is only used in the autonomous driving scene due to the high price.

H. Ma et al. (Eds.): PRCV 2021, LNCS 13019, pp. 16–28, 2021.
https://doi.org/10.1007/978-3-030-88004-0_2

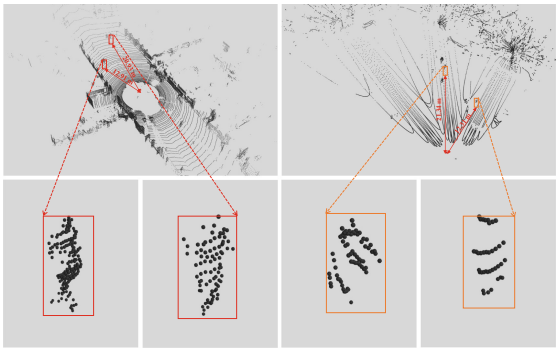


Fig. 1. Point clouds from Velodyne HDL-64E (left) and Livox Mid-100 (right). Point clouds of four persons with labeled distance from LiDAR are present in the second row.

In recent years, with the launch of low-cost LiDAR, such as Livox Mid-100, it is becoming feasible to apply LiDAR to visual surveillance applications. As shown in Fig. 1, the point cloud density of low-cost LiDAR is much lower than the normal LiDAR such as Velodyne HDL-64E laser scanner, making 3D object detection and tracking a very challenging problem.

Despite rapid progress, 3D MOT systems within tracking-by-detection paradigm still suffer a lot from the extreme sparsity of point clouds, which impairs both object detection stage and tracking stage. Most previous object detection works focus on single point cloud frame and leave out the temporal information between multi-frames. Subsequently, tracking-by-detection tracking methods struggle to find the association between current and past trajectories only using the detection bounding boxes of the current frame. To improve the suboptimal solutions based on single frame detection and tracking-by-detection methods, a novel architecture of detection and tracking for point clouds is introduced in this paper by considering the feature of stationary LiDAR. It is rational to aggregate point clouds over different timestamps and leverage temporal information more effectively when the LiDAR is fixed to the ground.

In this paper we propose a 3D multi-object detection and tracking method, with the application in football player tracking. Our method is composed of two stages: a short-term tracklet regression stage and a 3D Distance-IoU [25] (DIoU) data association stage. The first stage takes as input three successive point cloud frames and extracts features separately by the backbone of PointPillars [6]. The three feature maps are aggregated together utilizing 3D convolutional operations to predict short-term tracklets over three past frames. Since adjacent short-term tracklets are overlapping, the following data association stage calculates the DIoU scores of the overlapped pairs from the same object. Finally, a combination of split, birth and death module is applied to get the final tracking result.

As most of public point cloud datasets for 3D object detection and tracking are captured in autonomous driving scene such as KITTI [4], nuScenes [2] and Waymo [18], we build a simulated point cloud dataset for 3D detection and track-

ing with stationary LiDAR using Google Research Football [5]. We also acquire real point cloud data employing Livox Mid100 LiDAR on a football playground[1]. We train and evaluate our approach on our simulated dataset and test the generalization ability through the real data. We extend our method to nuScenes dataset to show its generalization ability despite that it was designed specifically for stationary LiDAR. Our experimental results show that aggregating temporal information to predict multi-frames detections improves the detection accuracy in sparse point cloud and D-IoU score used for matching conduces to a robust tracking result.

2 Related Work

2.1 3D Object Detection

Traditional football player detection and tracking algorithms are mostly based on RGB videos [11]. Despite rapid progress in CNN based object detection [1, 22], image-based methods face several challenges, such as processing of complex background and accurate estimation of 3D location.

3D object detection systems taking point clouds as input can output highly accurate 3D location, which can be divided into three categories [15]: view-based methods, voxel-grid based methods and raw point cloud-based methods. View-based [17] approaches always convert the point clouds into 2D image views such as the forward views, cylindrical views or bird's eye views. Voxel-grid based methods [7] firstly discretize the 3D point cloud into voxels, each of which is the binary occupancy of the voxel or the amount of points in the voxel. Point RCNN [16] takes as input unstructured point clouds without discretization, which is made up of two stages. Yang *et al.* [23] extended the raw point cloud-based method votenet [12] into the autonomous scenes with a novel fusion sampling strategy.

All of the detectors above process single frame without exploring the temporal information. However, Livox LiDAR can only capture one point on a human-sized object 100 m away in a scanning frame and five points of 50 m, which calls for the multi-frame fusion urgently. Though [24] leverages temporal information utilizing AST-GRU from consecutive frames, the detection results among each frames are lack of connection, which is not for tracking. Our temporal feature aggregation component instead outputs a short-term tracklet capturing temporal information, which contributes to robust tracking results.

2.2 3D Multi-Object Tracking

Most of the 3D multi-object tracking systems within the tracking-by-detection paradigm share the same structure as 2D MOT systems with changing the detection of 2D image plane to 3D space. Online Tracking methods [3,20,21] extract motion features or appearance features to compute the affinity matrix

[1] We plan to make both datasets publicly available.

and solve the bipartite graph matching problem by greedy or Hungarian algorithm. Kalman filters are commonly used to estimate the current state of motion features. Weng *et al.* [20] used 3D IoU to match detections with the previous frame and Chiu *et al.* [3] employed the Mahalanobis distance for data association between predicted and actual object detections instead. In [21], a novel joint feature extractor was proposed to learn appearance and motion features from 2D and 3D space simultaneously and a GNN-based architecture is used for data association.

Several methods [10,19] focus on the detection and tracking jointly from point cloud videos. Wang *et al.* [19] proposed an end-to-end 3D object detection and tracking network to generate point-wise tracking displacements for each object. Luo *et al.* [10] proposed a single network to do 3D detection, motion forecasting and tracking together with the spatio-temporal information. Qi *et al.* [14] proposed a end-to-end trained network only for 3D single object tracking in point clouds. Our network directly utilizes PointPillars [6] as backbone to predict 3D multi-object short-term tracklets over historical frames, which is beneficial to the tracking stage.

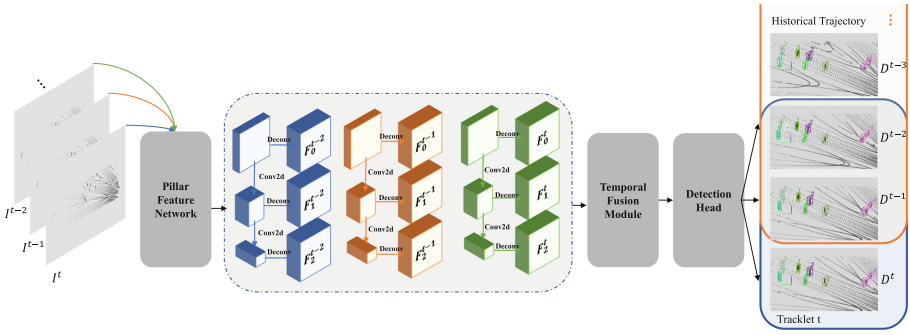


Fig. 2. System overview of our tracklet regression stage. We encode point cloud by a PFN layer and extract high-level features utilizing a 2D convolutional backbone. A temporal fusion module is applied to aggregate the temporal information of three frames. Finally, the detection head predicts a short-term tracklet (blue) as the result, which has two frames overlap with historical trajectory (orange) for tracking. (Color figure online)

3 Proposed Method

In this section, we elaborate on the framework of our 3D MOT method which consists of two main stages: tracklet regression stage and tracking stage. In the following, we first describe the tracklet regression stage in Sect. 3.1, which takes as input three point cloud frames and outputs the short-term tracklets. The data association stage is presented in Sect. 3.2 which links the tracklet regression results to the historical trajectories. Finally, the details of our simulated dataset and real data are provided in Sect. 3.3.

3.1 Tracklet Regression

Our data acquisition experiment showed that there is at most one point on a human-sized object 100 m away from the Livox Mid100 LiDAR in a scanning frame, which leads to a serious decline in the detection performance. A study on nuScenes dataset [2] has reported that accumulation of ten LiDAR sweeps by merging ten sweeps together in the coordinate system of key frame leads to a significant performance increase. However, the raw data accumulation makes it inefficient to leverage the temporal information. Our detector instead introduces a temporal fusion module of three frames in the stationary scenes to avoid the impairment of sparsity. The aggregated temporal information makes the network capable of capturing the motion features and giving a multi-frame tracklet as the result. The structure of our detector is shown in Fig. 2.

Feature Extracting Backbone. The tracklet detector first takes three consecutive point cloud frames $\{I^{t-2}, I^{t-1}, I^t\}$ as input, each of which is discretized uniformly into a set of pillars in the x-y plane after removing the surroundings. The non-empty pillars are fed into a Pillar Feature Network (PFN) [6] which is a simplified version of PointNet [13] here to extract the features and then scattered back to form a pseudo-image of size (C, H, W), where the H and W are decided by the grid size. The pseudo-images of three inputs are denoted by $\{F^{t-2}, F^{t-1}, F^t\}$. Then a top-down network composed of three 2D convolutional blocks is applied to each pseudo-image to extract high-level features. The structure of 2D backbone shown in Figs. 2 shares the same settings with the pedestrian backbone in PointPillars [6]. The features of each top-down block are upsampled to the input size by the following upsampling operation so that 3×3 final features in total are generated from three input frames.

Temporal Fusion Module. Towards the goal of predicting the tracklet, we take all of the output features from the 2D backbone to aggregate the temporal information. In contrast to the accumulation of nuScenes dataset [2] which is denoted by *early fusion*, our temporal fusion module applied after feature extracting is called *late fusion*. The structure of our temporal fusion module is illustrated in Fig. 3. With 3×3 final features of size (C, H, W), we utilize the features of the same level to form a group, which can be taken as a 4D tensor of size $(C, 3, H, W)$. Then a 3D convolution layer with $3 \times 3 \times 3$ kernel followed by BatchNorm and a ReLU is performed on each 4D tensor with no padding in the temporal dimension to aggregate the information of the same level. The aggregated features of three levels are squeezed and combined together through concatenation to form a tensor of size $(3C, H, W)$, which is fed into the detection head to obtain the final detection results.

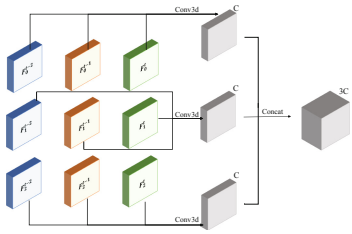


Fig. 3. The structure of the temporal fusion module. The extracted features of three frames at the same level are formed to create a 4D tensor and three levels are fused through performing a 3D convolution and concatenation.

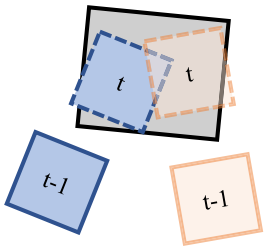


Fig. 4. The collision of the blue object and the red object leads to a undetected error. The black bounding box is the fusion of two objects actually so we obtain dotted bounding boxes M^t to split the black box D^t to regain the real bounding boxes. (Color figure online)

Detection Head. The fusion of temporal information allows the detection head to capture the motion features such as velocity or acceleration and thus to predict the tracklet of the object. Our detection head following SSD [9] uses two predefined anchor boxes at each location for short-term tracklet regression, which is denoted by three 3D bounding boxes $\{D^{t-2}, D^{t-1}, D^t\}$ for 7×3 regression values. The ground truth tracklet composed of 3D bounding boxes $\{G^{t-2}, G^{t-1}, G^t\}$ is assigned through the 2D IoU between current bounding box G^t and anchor boxes, and then all three boxes are encoded by the corresponding anchor with height and heading angle of the object as additional regression targets. And thus, in addition to the classification scores s_i and box regression of current frame $(x_i^t, y_i^t, z_i^t, l_i^t, w_i^t, h_i^t, \theta_i^t)$ at location i, we add two branches to predict the other two bounding boxes $(x_i^m, y_i^m, z_i^m, l_i^m, w_i^m, h_i^m, \theta_i^m)_{m \in \{t-1, t-2\}}$, which means that each location in the feature map obtains bounding boxes of three frames $\{D^{t-2}, D^{t-1}, D^t\}$ to make up of a short-term tracklet. The short-term tracklet non maximum suppression (NMS) is actually processed on the bounding boxes of current frame, and then tracklets are filtered by classification scores to get the final regression result.

3.2 Data Association

The data association stage takes as input the predicted short-term tracklets $\{D^i\}_{i=t-2}^t$ and the historical trajectories $\{M^i\}_{i=1}^{t-1}$, and finally outputs the updated trajectories $\{M^i\}_{i=1}^t$. We utilize 3D Distance-IoU (DIoU) [25] as the distance measurement since sparse point clouds contain no appearance information of objects. Due to the tracklet regression results, we compute the DIoU scores of overlapping frames directly without Kalman filters to predict the object state. Then we apply a Hungarian matching algorithm to link the trajectories.

We propose a trajectory management module to handle the collision, birth and death of trajectories to improve the tracking robustness.

3D DIoU. As we have mentioned in Sect. 3.1, the extreme sparsity of points in the large-scale scenes impairs the appearance features for that a few 3D points per object make it hard to distinguish the object from others through for even manual annotations. Hence we utilize the motion features represented by the short-term tracklets $\{D^{t-2}, D^{t-1}, D^t\}$ to compute the affinity of objects. Rather than using normal IoU in previous works [20], we use the 3D DIoU [25] instead to avoid the collapse of the normal IoU for non-overlapping or other special cases. Normal IoU only works when two bounding boxes have overlap, and would keep zero while providing no affinity information for non-overlapping cases. And thus, DIoU add a penalty term to consider the normalized distance between two central points in addition to the overlapped area. We extend the 3D DIoU following the penalty term of 2D DIoU while considering the additional dimension and heading angle, which is calculated as follows:

$$DIoU_{3D} = IoU_{3D} - \frac{\rho^2(c_1, c_2)}{\rho^2(a, b)} \tag{1}$$

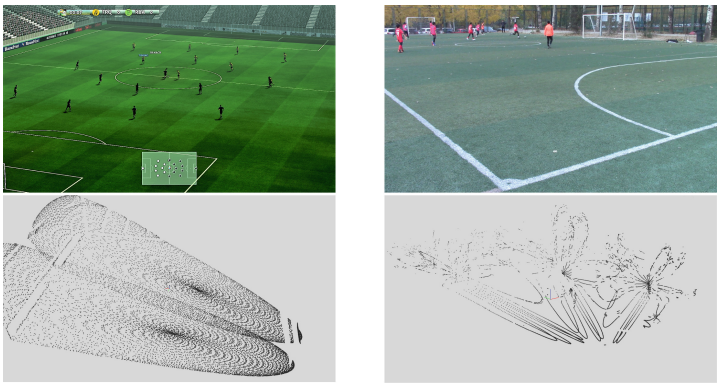
where c_1, c_2 are the center of two bounding boxes and $\rho(a, b)$ is the farthest point pair of the two boxes. With the penalty term, two bounding boxes far apart will get a lower 3D DIoU score despite no overlap. The distance measurement for non-overlapping case is necessary for the tolerance of inevitable detection errors due to the movement of object in the large-scale scene.

Association Module. Given the short-term tracklet $\{D^i\}_{i=t-2}^{t}$ and the historical trajectories $\{M^i\}_{i=1}^{t-1}$, the association module utilizes two overlapped frames to compute the affinity matrix. Considering the p objects of the short-term tracklet D and q objects of the trajectory M, we apply the two overlapped pairs of $\{D^{t-2}, M^{t-2}\}$ and $\{D^{t-1}, M^{t-1}\}$ to compute two affinity matrices of $S_{(p\times q)}^{t-1}$ and $S_{(p\times q)}^{t-2}$ using 3D DIoU. The final affinity matrix $S_{(p\times q)}$ is given through the weighted summing of two DIoU matrices considering the precision of bounding boxes regression over three frames. Given the affinity matrix between the tracklets and historical trajectories, we adopt the Hungarian algorithm to solve the bipartite graph matching problem. Finally, the matched pairs of $\{D^i\}_{i=t-2}^{t}$ and $\{M^i\}_{i=1}^{t-1}$ are combined together to update the trajectories as $\{M^i\}_{i=1}^{t}$.

Trajectory Management Module. Considering the frequent collision of football players, we propose a split module to deal with the players joined together. As shown in Fig. 4, the collision of the blue object and the red object leads to an undetected error that the black detection bounding box is the fusion of two objects. In this way, the data association module might mistakenly switch the track IDs of these two objects after the collision. Based on the observation that

the location and heading angle of the fusion bounding box are almost the average of two separate objects, we propose a split module to generate two bounding boxes instead of the false one to avoid the ID switch, which is fulfilled by taking the separate bounding boxes to split the fused bounding boxes.

Occlusion of objects or the sparsity of point clouds give rise to the birth and death of trajectories in the center of the scene. We retain all the unmatched historical trajectories as the dying trajectories for at most 6 frames. The dying trajectories are updated by the Kalman filter to keep in the scene, which matched in 6 frames are recovered to the normal ones to avoid the false negative, otherwise those unmatched for 6 continual frames are deleted from the scene. The association with a lower threshold of dying trajectories is performed after the matching between short-term tracklets and normal trajectories to reduce the trajectory breaks. At the same time, we accept the unlinked short-term tracklets as the initialized trajectories with new tracking identities. These initialized trajectories will be killed unless it has been consecutively detected for three frames to avoid false positives.



(a) (b)

Fig. 5. The simulated dataset (left) and the real data (right). We present the RGB data in the top and point cloud in the bottom.

3.3 Football Game Dataset

Several 3D point cloud datasets have been released in the past decade for autonomous driving while point cloud data of surveilance scenes captured by LiDAR fixed to the ground is especially scarce. Hence we build a simulated point cloud dataset and capture the real data for the football game scenes.

The cost of 3D point cloud dataset annotations is extremely high. For this reason, we use the Google Research Football [5] to generate the simulated data.

To be consistent with the real world data from the Livox Mid LiDAR, we sample on the depth map of google research football with a non-repetitive scanning pattern as shown in Fig. 5a, which is implemented by a polar rose curve $\rho = 350 \sin(3\pi\theta + \theta_0)$. The origin of coordinates is set to the center of the football playground. Google Research Football provides 12 keypoints of each person in the playground without bounding box. Accordingly, we consider the midperpendicular of the line between left and right shoulder as the heading direction and use the keypoints to generate the enclosing bounding box modeled as x, y, z, width, length, height and yaw angle. The simulated dataset is captured 11 Hz and consists of four point cloud videos for 821 frames in total.

Moreover, we record a football game using Livox Mid100 LiDAR with frequencies 20 Hz. Subsequently, we merge every two sweeps to the frequency 10 Hz following the ground plane calibration. The real data is composed of 1,121 consecutive frames for 116 s. The scene and the point cloud are illustrated in the Fig. 5b. We have manually labelled 200 frames to test the generalization ability of our method.

4 Experiments

4.1 Settings

Datasets and Metrics. We evaluated the proposed algorithm both on the point cloud simulated dataset and the real data. We also conducted experiments on nuScenes [2] dataset to prove the effectiveness of the 3D DIoU tracking method. We merged ten LiDAR sweeps to the key frame as the raw input data due to 2 Hz annotations.

The detection results of simulated dataset are all reported through employing the similar metrics of KITTI [4] in both bird's eye view (BEV) and 3D over IoU threshold of 0.5. The mAP (mean average precision) is used to evaluate the performance of three bounding boxes regression. We followed the AB3DMOT [20] and used sAMOTA, AMOTA and AMOTP as main tracking metrics. In addition, MOTA, MOTP, IDS, FRAG from CLEAR metrics are also applied as secondary metrics.

Implementation Details. As for simulated dataset, we only considered the point cloud within the range of $[-40, 40] \times [-40, 40] \times [-0.1, 2.5]$ meters to remove the surroundings, which was discretized to the grids of 0.25^2 m^2. The details of our backbone follow the implementation of PointPillars [6].

The lower bound threshold of matching algorithm is set to -0.125 for normal trajectories and -0.5 for dying trajectories. The maximum duration of dying trajectories is 6 frames and the minimum duration of new trajectories is 3 frames.

We modified the loss function to add the loss of past and future frames, which is shown as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{cls} + \sum_{i=t-2}^{t} \lambda_i \mathcal{L}_{loc}^{i} + \beta \mathcal{L}_{dir} \tag{2}$$

where $\mathcal{L}_{cls}$ is the focal loss [8], $\mathcal{L}_{loc}^{i}$ is smooth L1-norm and $\mathcal{L}_{dir}$ is the classification loss of heading angles as PointPillars [6]. We set $\alpha = 1, \lambda_{t-2} = \lambda_{t-1} = \lambda_t = 2, \beta = 0.2$ to make a balance.

4.2 Experimental Results

Results on Football Dataset. We report our method's short-term tracklet regression results on the simulated dataset. Firstly, to verify the effectiveness of our temporal fusion module, we present the D^t bounding box detection performance comparison of our method and other approaches in Table 1. PointPillars [6] with a single frame was used as the baseline in the experiment. Early fusion was conducted by merging 3 successive LiDAR sweeps as raw input of PointPillars. We can see that both early fusion and our late fusion method aggregate the temporal information and thus outperform the baseline of the single-frame method as expected. Additionally, late fusion achieved a higher mAP compared with the early fusion and outperformed by a large margin (6.18% in BEV and 14.86% in 3D), which revealed that the fusion of extracted features leverages temporal information more effectively. We evaluated on the point cloud of twice density and we also found that our multi-frame fusion module suffers less from the sparsity of point cloud. The 3D performance of our late fusion method was reduced by 11.83% mAP with the half density while single frame method got 17.91% drop.

Table 1. The comparative results of methods with different temporal information on the simulated dataset

Method	BEV	3D	Δ of 3D
Single Frame (×2)	56.46	51.65	–
Late Fusion (×2)	86.26	83.30	–
Single Frame (×1)	44.51	33.74	–17.91
Early Fusion (×1)	69.03	56.61	–
Late Fusion (×1)	75.21	71.47	–11.83

Table 2. Tracking performance on the simulated dataset

Method	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	MOTP↑	IDS↓	FRAG↓
AB3DMOT [20]	76.43	34.68	41.59	82.11	51.03	**2**	167
3D DIoU	84.49	39.58	**46.96**	85.82	51.72	42	196
3D DIoU + Birth/Death	85.14	39.43	45.22	86.14	**51.73**	5	**163**
3D DIoU + Birth/Death/Split	**85.56**	**39.96**	45.01	**86.24**	51.69	4	**163**

The main results of 3D multi-object tracking are summarized in Table 2. We report the AB3DMOT [20] result in Table 2 as baseline. We can see that our

method achieves higher AMOTA and AMOTP compared with the AB3DMOT baseline applying the public detection of the tracklet regression stage. We also conducted an experiment to evaluate the effect of the management modules in our method. As shown in the Table 2, adding the birth and death module reduces IDS and FRAG which enhances the robustness of trajectories. The birth module reduces the false positives caused by noise point cloud, and the death module alleviates the effect of occlusion which may lead to tracjectory breaks. Including the split module improves the tracking performance further.

We tested our model on 200 annotated frames of the real data. Our tracking results with a higher sAMOTA and AMOTA compared with AB3DMOT [20] are showed in Table 3. The qualitative 3D tracking results of the real data are shown in Fig. 6 with different color trajectories for different players.

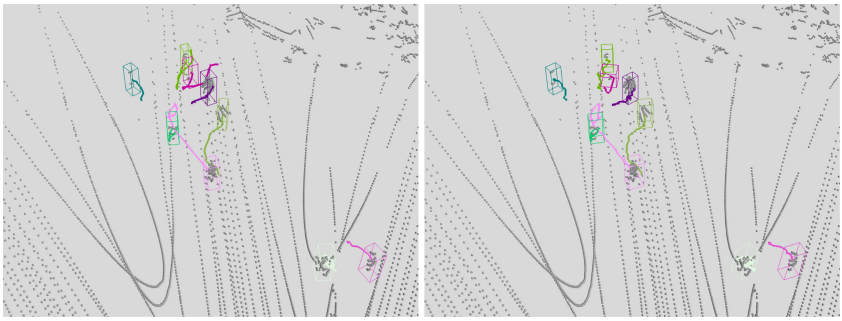


Fig. 6. 3D tracking visualization of real data. The annotated trajectories (left) and our tracking results (right) are drawn in different colors for different players.

Quantitative Results on nuScenes Benchmark. In addition to evaluation on football game dataset, we also report our preliminary results without fine adjustment on the car and pedestrian subset of nuScenes [2] dataset, which was captured using LiDAR mounted on cars. We should emphasize that the input point cloud of nuScenes is very dense since 10 sweeps are merged as input to be consistent with 2 Hz annotations and evaluation benchmark, which weakens the strength of our method. In addition, point cloud from moving LiDAR makes it hard to fuse the temporal information compared to the stationary LiDAR. In order to evaluate 3D trajectory more effectively, we proposed a metric of the 3D trajectory estimation error. The 3D trajectory estimation error of our method tested on nuScenes is 3.620 m for cars and 3.826 m for pedestrians, which is a promising result. Table 4 presents the AMOTA tracking results of our method on nuScenes validation set with the PointPillars detection results. We also report the official results of AB3DMOT [20] with a better detection results than PointPillars. We can see that our method outperforms by 4.3% compared with AB3DMOT in spite of the poorer detection due to the effectiveness of our 3D DIoU tracklet association method.

Table 3. Tracking performance on real data

Method	sAMOTA↑	AMOTA↑
AB3DMOT [20]	44.04	19.97
Ours	49.92	21.53

Table 4. Tracking results on nuScenes val set

Method	Car	Pedestrian
AB3DMOT [20]	69.4	58.7
Ours	73.7	63.0

5 Conclusion

We propose an online 3D multi-object detection and tracking method for sparse stationary LiDAR, which consists of two stages: tracklet regression stage and data association stage. We propose a temporal fusion module to give the short-term tracklet as output and a data association stage that utilizes 3D DIoU to link the tracklets and historical trajectories. Moreover, we build a simulated point cloud dataset for the football game scenes and capture the point cloud data of a real football game. The experimental results show that our tracklet regression and 3D DIoU data association method get a promising performance. We hope to reduce the computational cost to meet real time requirements later. We will also push on the collection and annotation of more real data and generate more realistic simulated data in the future.

Acknowledgments. The work was supported by the National Key Research and Development Program of China under Grant 2018AAA0102803.

References

1. Buric, M., Ivasic-Kos, M., Pobar, M.: Player tracking in sports videos. In: Cloud-Com (2019)
2. Caesar, H., Bankiti, V., Lang, A.H.: nuScenes: a multimodal dataset for autonomous driving. CoRR (2019)
3. Chiu, H., Prioletti, A., Li, J., Bohg, J.: Probabilistic 3D multi-object tracking for autonomous driving. CoRR (2020)
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR (2012)
5. Kurach, K., et al.: Google research football: a novel reinforcement learning environment. In: AAAI (2020)
6. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: fast encoders for object detection from point clouds. In: CVPR (2019)
7. Li, B.: 3D fully convolutional network for vehicle detection in point cloud. In: IROS (2017)
8. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. **42**, 318–327 (2020)
9. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2

10. Luo, W., Yang, B., Urtasun, R.: Fast and furious: real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In: CVPR (2018)
11. Manafifard, M., Ebadi, H., Moghaddam, H.A.: A survey on player tracking in soccer videos. Comput. Vis. Image Underst. **159**, 19–46 (2017)
12. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3D object detection in point clouds. In: ICCV (2019)
13. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. In: CVPR (2017)
14. Qi, H., Feng, C., Cao, Z., Zhao, F., Xiao, Y.: P2B: point-to-box network for 3D object tracking in point clouds. In: CVPR (2020)
15. Rahman, M.M., Tan, Y., Xue, J., Lu, K.: Recent advances in 3D object detection in the era of deep neural networks: a survey. IEEE Trans. Image Process. **29**, 2947–2962 (2020)
16. Shi, S., Wang, X., Li, H.: Pointrcnn: 3D object proposal generation and detection from point cloud. In: CVPR (2019)
17. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G.: Multi-view convolutional neural networks for 3D shape recognition. In: ICCV. IEEE Computer Society (2015)
18. Sun, P., et al.: Scalability in perception for autonomous driving: waymo open dataset. CoRR (2019)
19. Wang, S., Sun, Y., Liu, C., Liu, M.: Pointtracknet: an end-to-end network for 3-D object detection and tracking from point clouds. IEEE Rob. Autom. Lett. **5**, 3206–3212 (2020)
20. Weng, X., Wang, J., Held, D., Kitani, K.: 3D multi-object tracking: a baseline and new evaluation metrics. In: IROS (2020)
21. Weng, X., Wang, Y., Man, Y., Kitani, K.M.: GNN3DMOT: graph neural network for 3D multi-object tracking with 2D–3D multi-feature learning. In: CVPR (2020)
22. Yang, Y., Xu, M., Wu, W., Zhang, R., Peng, Y.: 3D multiview basketball players detection and localization based on probabilistic occupancy. In: DICTA (2018)
23. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3DSSD: point-based 3D single stage object detector. In: CVPR (2020)
24. Yin, J., Shen, J., Guan, C., Zhou, D., Yang, R.: LiDAR-based online 3D video object detection with graph-based message passing and spatiotemporal transformer attention. In: CVPR (2020)
25. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-IoU loss: faster and better learning for bounding box regression. In: AAAI (2020)