Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8895696, 10 pages
https://doi.org/10.1155/2020/8895696

Research Article

A 3D Multiobject Tracking Algorithm of Point Cloud Based on Deep Learning

Dengjiang Wang, Chao Huang, Yajun Wang, Yongqiang Deng, and Hongqiang Li

VanJee Technology Co., Ltd., Beijing 100193, China

Correspondence should be addressed to Yajun Wang; wangyajun@wanji.net.cn

Received 25 September 2020; Revised 24 November 2020; Accepted 28 November 2020; Published 10 December 2020

Academic Editor: Bekir Sahin

3D multiobject tracking (MOT) is an important part of road condition detection and hazard warning algorithm in roadside systems and autonomous driving systems. There is a tricky problem in 3D MOT that the identity of occluded object switches after it reappears. Given the good performance of the 2D MOT, this paper proposes a 3D MOT algorithm with deep learning based on the multiobject tracking algorithm. Firstly, a 3D object detector was used to obtain oriented 3D bounding boxes from point clouds. Secondly, a 3D Kalman filter was used for state estimation, and reidentification algorithm was used to match feature similarity. Finally, data association was conducted by combining Hungarian algorithm. Experiments show that the proposed method can still match the original trajectory after the occluded object reappears and run at a rate of 59 FPS, which has achieved advanced results in the existing 3D MOT system.

1. Introduction

With the rapid development of computer vision, image processing, and other technologies as well as the emergence of deep learning, the field of object detection has achieved great development. From the high accuracy of two-step RCNN [1], fast RCNN [2], and faster RCNN [3] to the high speed of one-step YOLO [4], YOLOv2 [5], YOLOv3 [6], and SSD [7] and from anchor-based methods [8, 9] to anchor-free methods [10, 11], object detection has made great progress in both accuracy and speed. At the same time, the development of object detection also promoted the development of other fields, including object tracking. Multiobject tracking is a branch of object tracking, which is closely related to the development of object detection [12]. Object tracking algorithm is divided into single-object tracking algorithms [13] and multiobject tracking algorithms [14]. Single-object tracking algorithms are widely used in monitoring and navigation systems. Among the single-object tracking algorithms, SiamMask [15] only needs to initialize the frame; then it can generate the masks segmented with the object and the boundary frames in the video with the speed up to 35 FPS. SiamRPN++ [16] develops a Siamese tracker based on ResNet architecture. Chen et al. [17] proposed a multiscale fast correlation filtering tracking algorithm based on a feature fusion model. Zhang et al. [18] exploited spatial and semantic convolutional features extracted from convolutional neural networks in continuous object tracking. Multiobject tracking is widely used in autonomous driving systems because it can associate the results of object detection in time without switching the identities of multiple targets [19, 20]. The autonomous driving system can estimate the location of the object by using tracking algorithm and avoid accidents. In the MOT algorithm, simple online and realtime tracking (SORT) [21] adopts the Kalman filter and Hungarian matching algorithm to track the objects, which obtains fast and great tracking performance, but it may cause the ID switch of the occluded object after it reappears. In order to reduce the frequency of ID switch, simple online and realtime tracking with a deep association metric (DeepSORT) [22] was proposed. DeepSORT combines the advantages of SORT, and it makes up for the defects of the SORT by adding the reidentification network of pedestrians, extracting the pedestrian features, and

matching the feature similarity. Ristani and Tomasi [23] put forward DeepCC algorithm, and Tang et al. [24] put forward LMP algorithm; all of these algorithms use reidentification to improve the performance of tracking algorithm through matching the similarity of trajectories. In order to enhance the robustness of complicated changes of multiple objects and complex background scene, Chen et al. [25] proposed the visual object tracking algorithm based on adaptive combination kernel. In addition, tracking has many other applications, such as tracking in basketball games [26].

Since the related algorithms become more and more proven in image processing, the development of image object detection algorithm cannot escape from the limitations of two-dimensional data, and the drawbacks of data are more obvious, which lead to many problems in the algorithm. For example, object detection and tracking algorithm are greatly affected by light, rain, snow, and haze weather. Under such conditions, the object detection accuracy is low and the recognition results are two-dimensional without including distance and volume. However, the point clouds acquired by LiDAR are little affected by the light and have the information of distance and volume, which can overcome these problems above and make up for the shortages of image processing. In recent years, with the decrease of LiDAR cost, more and more researchers use LiDAR to replace the camera for object detection. At the meantime, different from image, point clouds are sparse and disorder space points. However, the proven algorithm used in the image processing cannot be directly used in point clouds. In order to solve this problem, many researchers adopted projection methods [27–31] to project 3D objects into multiple views and fuse the features of each view for detection and recognition. Using the projection method provides a transformation idea from point clouds to image processing. However, a large number of projections will cause the increase of computation, while reducing the number of projections will cause the lack of information. Wu et al. [32] and Le and Duan [33] applied the idea of voxelization to voxelate the point clouds and processed it directly, which improved the efficiency of object detection. The development of point cloud object detection also promoted the development of point cloud tracking algorithm. Weng and Kitani [34] extended the two-dimensional SORT to three-dimensional and proposed the AB3DMOT algorithm, which performed well on the KITTI dataset [35]. In order to improve the performance of point clouds multiobject tracking and retrieve the ID information of occluded objects, we combine reidentification algorithm of pedestrian and 3D Kalman filter and apply them to point clouds. Our contributions are as follows:

(i) The tracking algorithm based on deep learning of image processing is introduced into the tracking algorithm based on point cloud, and a tracking algorithm model based on deep learning is established.

(ii) The proposed tracking algorithm model uses the three-channel image composed of bird's eye view (BEV), density, and intensity maps of the point cloud to train the point cloud reidentification network. The two-dimensional features of the three-channel image are extracted by using the point cloud recognition network, and they are made cascade matching with the location features of the IOU.

(iii) The proposed tracking algorithm model performs well in the point cloud tracking algorithm. The original trajectory can be matched again after occlusion. The proposed model provides a new baseline for the point cloud tracking algorithm.

2. Related Works

2.1. 3D Object Detection. 3D object detection is an indispensable part of 3D object tracking, and the 3D bounding box of detection is also very important for the effect of tracking. 3D object detection can be divided into four categories: image processing methods, voxel-based methods, point-based methods, and some fusion methods. Li et al. [36] presented 3D point cloud to 2D image, and then used the 2D end-to-end full convolution neural network to predict target confidence and 3D bounding boxes through bounding boxes encoding. Simon et al. [37] transformed point clouds into BEV map, density map, and intensity map, and used the method of image processing for 3D detection. Zhou and Tuzel [38] proposed VoxelNet, which divided point clouds into different voxels. Then, they used the VFE (Voxel Feature Encoding) layer to encode features uniformly. Finally, RPN (region proposal network) was used for category classification and 3D bounding boxes regression. Based on the VoxelNet, Yan et al. [39] proposed sparsely embedded convolutional detection (SECOND) by using sparse convolution, which improved the accuracy of detection further. Qi et al. [40] put forward PointNet through using point clouds directly. PointNet adopted spatial transformation matrix to align point clouds and the combined convolutional neural network (CNN) to obtain good results in object segmentation and detection. This method is a better one than two-dimensional image processing. Later, in order to solve the shortcomings of PointNet, Qi et al. [41] put forward PointNet++ by modifying PointNet. Shi et al. [42] put forward PV-RCNN by combining the advantages of voxel-based and point-based methods and then achieved the highest score on KITTI data. In addition, there are some other multisensor fusion methods: MV3D [43] fused BEV and front view of point clouds with RGB image; AVOD [44] fused RGB images and six-channel BEV map consisting of five equal height slices and density map; and F-ConvNet [45] used 2D region to estimate end-to-end of bounding boxes in 3D space.

2.2. 3D MOT. The difference between 3D MOT and 2D MOT is that the tracking objects of 3D MOT are three-dimensional and have height information and distance information. Osep et al. [46] proposed a 2D-3D Kalman filter to jointly use images and the 3D world coordinate system. Baser et al. [47] proposed an online multiobject tracking method based on

CNN. Hu et al. [48] used long short-term memory network (LSTM) learning module to predict long-term motion more accurately. Frossard and Urtasun [49] described this problem as a linear programming problem and adopted CNN to detect and match end-to-end. Zhang et al. [50] put forward mmMOT to encode point clouds in the process of data association and realized the fusion of multimodal data. Shenoi et al. [51] developed JRMOT which used a two-dimensional RGB image and three-dimensional point cloud. Here, three-dimensional point cloud was used for detection, and a two-dimensional RGB image was used for reidentification based on CNN, and then multi-object tracking was achieved. The camera shooting angle results in the occlusion of the object in a RGB image, so we combine the aerial view of point cloud with the reidentification method based on CNN to match the similarity and use the three-dimensional Kalman filter to predict the three-dimensional information of the object's movements.

3. Materials and Methods

According to the characteristics of point clouds, 2D and 3D separation methods are used. We use the 3D Kalman filter to predict the 3D coordinate information of the point clouds and extract the features of the bird's-eye view by the reidentification network. Our system uses the three-dimensional object detection networks such as SECOND to obtain the three-dimensional coordinate information X, Y, Z, L, W, H, and θ. These seven parameters represent the coordinates of the center point, length, width, height, and heading angle of the frame. The object detection results are transformed into 2D bounding boxes in the three-channel image which is composed of BEV, density, and intensity map, and then, they are sent to the reidentification network to extract features. X, Y, Z, L, W, H, and θ are used for state prediction and trajectory matching of the 3D Kalman filter. After that, the results of feature matching and 3D Kalman filter matching are output to obtain the ID information of the current detection results. The flow chart is shown in Figure 1.

3.1. 3D Object Detection. With the rapid development of 3D object detection, many 3D object detections have obtained good results in the KITTI dataset. We use the advanced 3D detector on the KITTI dataset to conduct experiments and directly use their detection results for performance test of tracking. The detection result of D is obtained by high-precision 3D object detection. D includes $\{X, Y, W, L, \theta, Z, H, S\}$ (S represents the detection score). D_t is the detection result of frame t and $D_t = \{D_{t1}, D_{t2}, \ldots, D_{tn}\}$ (n represents the number of objects detected). In addition, considering the detection speed and effect, we choose SECOND as the three-dimensional object detection detector of our tracking system. SECOND uses sparse convolution to improve significantly the speed of training and reasoning. The structure chart of SECOND is shown in Figure 2, and the detection performance is shown in Figure 3.

3.2. 3D Kalman Filter. In order to describe the moving object, we use the Kalman filter to predict the next frame state of it. It predicts the position of the current frame by the position information of historical target and then establishes the following state equation as equation (1) for each goal:

$$\mu = [x, y, z, l, w, h, \theta, \hat{x}, \hat{y}, \hat{z}, \hat{l}, \hat{w}, \hat{h}, \hat{\theta}]^T, \tag{1}$$

where x, y, and z are the x, y, and $z$coordinates of the point clouds, θ denotes the course angle, and l, w, and h denotes length, width, and height of the object, respectively.

By observing the movement law of vehicles and target characteristics of the point cloud, we find that the height and z coordinate of vehicle and pedestrian hardly changed during the movement. In order to reduce the calculation amount and improve the performance, we ignore the height H and Z coordinates. In the experiment, we find that adding angle will cause the increase of the radian of the predicted target, and the target's angle will be flipped over. Therefore, the final state model we use is as follows:

$$\mu = [x, y, l, w, \hat{x}, \hat{y}, \hat{l}, \hat{w}]^T. \tag{2}$$

The status of detection results can be expressed as follows:

$$D_t = \left[x_t^d, y_t^d, l_t^d, w_t^d, \theta_t^d, z_t^d, h_t^d\right]^T. \tag{3}$$

The predicted state equation can be expressed as follows:

$$T_{\text{test}} = \left[x_t^d + \hat{x}, y_t^d + \hat{y}, l_t^d + \hat{l}, w_t^d + \hat{w}, \theta_t^d, z_t^d, h_t^d\right]^T. \tag{4}$$

3.3. Point Cloud Reidentification. The point cloud is different from the image in which point cloud has no fine-grained features, and the fine features are difficult to distinguish. Although RGB images can be used for reidentification to obtain a large number of fine-grained features, they have some problems: the image may encounter obscuring; the farther the distance, the smaller the target; the farther the distance, the less distinctive the features which are even difficult to distinguish. On the contrary, the aerial view of point cloud has a large field of view and no occlusion of the object, which is conducive to reidentification and solves the problems existing in the image.

Reidentification can realize the matching of feature similarity in the trajectory, so that when it appears again after the object is blocked, it can find the original trajectory by comparing with the features in the trajectory, while the traditional matching method will cause id jump. We use the three-channel image composed of BEV map, density map, and intensity map of point clouds to replace the RGB image to realize feature extraction. Due to the difference between the point cloud coordinate system and the image coordinate system, equation (5) is used to convert the point cloud coordinate system to the image coordinate system, and the transformation diagram is shown in Figure 4:

$$\begin{cases} x_t = -y + w, \\ y_t = -x + h, \end{cases} \tag{5}$$

where x and y represent coordinates in the point cloud coordinate system, h denotes the distance from the point

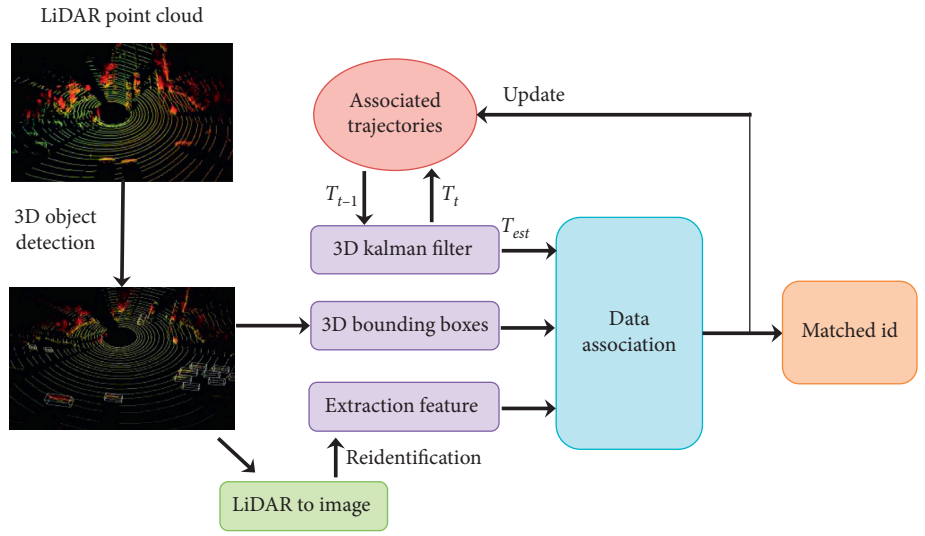

Figure 1: Our proposed 3D MOT system is composed by 3D object detection and tracking (data association and filtering) components. T_{t-1} and T_t refer to tracks at $t-1$ and tracks at t with the superscript indicating the space.

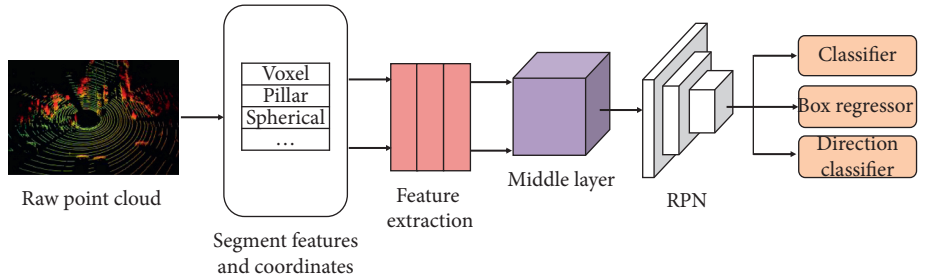

Figure 2: SECOND network structure diagram.

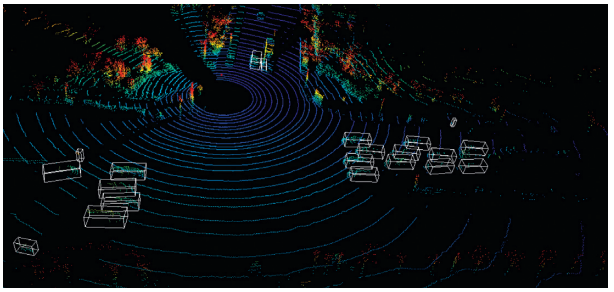

Figure 3: SECOND detects point clouds of roadside LiDAR.

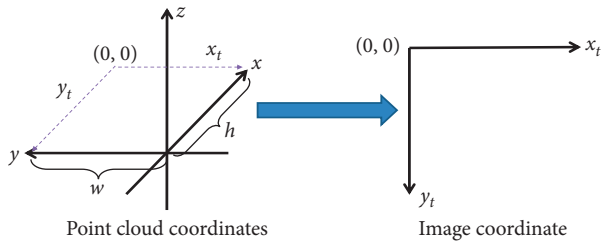

Figure 4: Converting point cloud coordinates to the image coordinate system.

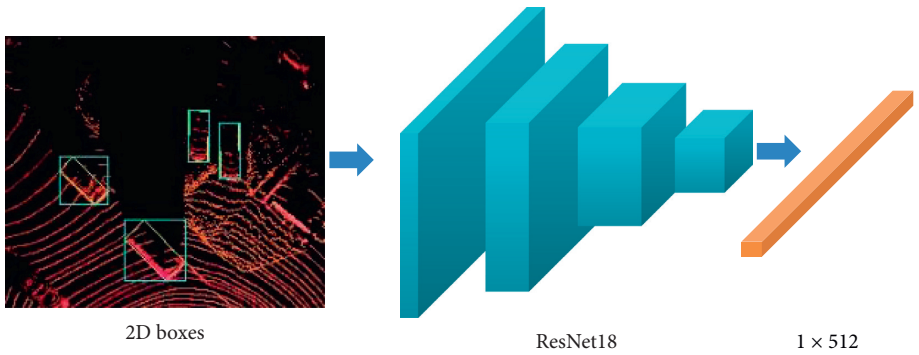

Figure 5: Reidentification network structure diagram.

Table 1: 3D experimental comparison.

Method	Samota	AMOTA	AMOTP	MOTA	MOTP	IDS	FRAG	FPS
FANTract [47]	0.8297	0.4003	0.7501	0.7430	0.7524	35	202	23.1
AB3DMOT [34]	0.9178	0.4426	0.7741	0.8335	0.7843	0	15	207.4
Ours KF	0.8842	0.4203	0.7777	0.7743	0.7856	229	275	882.5
Ours deep	0.9077	0.4361	0.7769	0.8084	0.7899	0	34	67

Table 2: 2D experimental comparison.

Method	MOTA	MOTP	IDS	FRAG	FPS
Complexer-YOLO [37]	0.7570	0.7846	1186	2092	100
DSM [49]	0.7615	0.8342	296	868	10
FANTrack [47]	0.7772	0.8283	150	812	25
AB3DMOT [34]	0.8384	0.8524	9	224	214.7
Ours	0.8047	0.8723	4	44	59

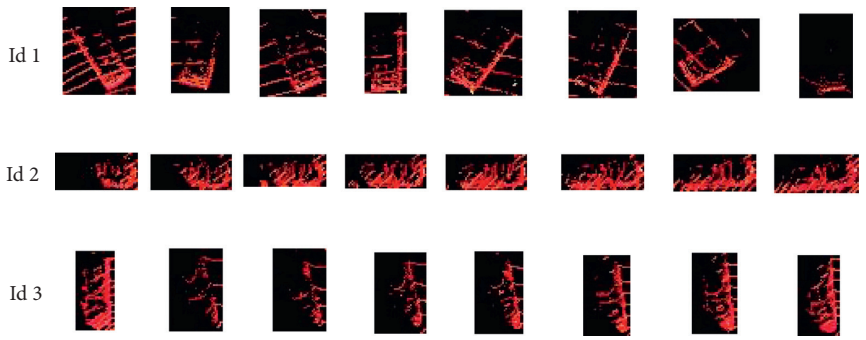

Figure 6: Reidentification of training samples.

cloud boundary to the y-axis, w is the distance from the point cloud boundary to the x-axis, and $x_t$and $y_t$represent the coordinates in the image coordinate system.

After coordinate transformation, the height of point clouds is mapped to the pixel value to obtain an aerial view, and then, the intensity value of corresponding points in the BEV map is mapped into the intensity map. Finally, we calculate the density value of corresponding point clouds in the image by using equation (6). The resultant three-channel picture is shown in Figure 5:

$$\rho_i = \frac{\min\left(1, \log\left(c_i + 1\right) - \log\left(c_{\min} + 1\right)\right)}{\log\left(c_{\max} + 1\right) - \log\left(c_{\min} + 1\right)}, \tag{6}$$

where$\rho_i$represents density of the ith location point, $c_i$represents the number of point clouds at the ith location

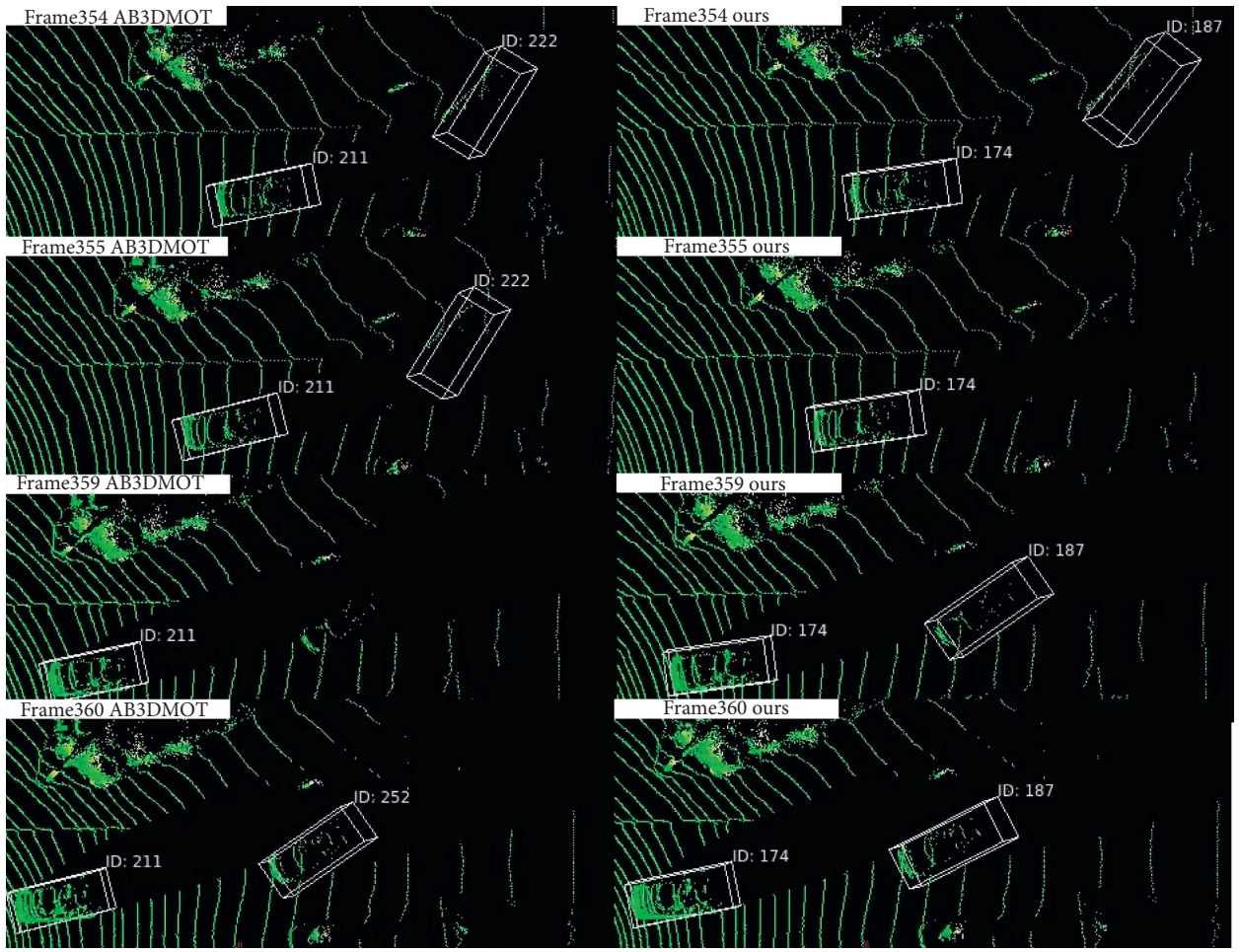

Figure 7: Effect comparison of frame 354 to frame 360 in Sequence 1.

point, $c_{\min}$is the minimum number of point clouds at the location point, and $c_{\max}$denotes the maximum number of point clouds at the location point.

After the image of the 2D bounding box in the converted three-channel image is cut and adjusted to $128 \times 64 \times 3$, it is sent to the reidentification network trained by ResNet18 for feature extraction. The reidentification network can match the similarity between the current detection box and the bounding box saved in the trajectory to find the trajectory of the target. The input size of the reidentification network is $128 \times 64 \times 3$, and the output feature vector is 1×512, which is shown in Figure 5.

4. Results and Discussion

The experiment in this paper is conducted on ubuntu16.04, GPU 1080ti. We use the KITTI MOT dataset and dataset of roadside 32-line LiDAR in our company to perform the evaluation. There are 21 sequences in the KITTI training set and 29 sequences in the test set which include point clouds,

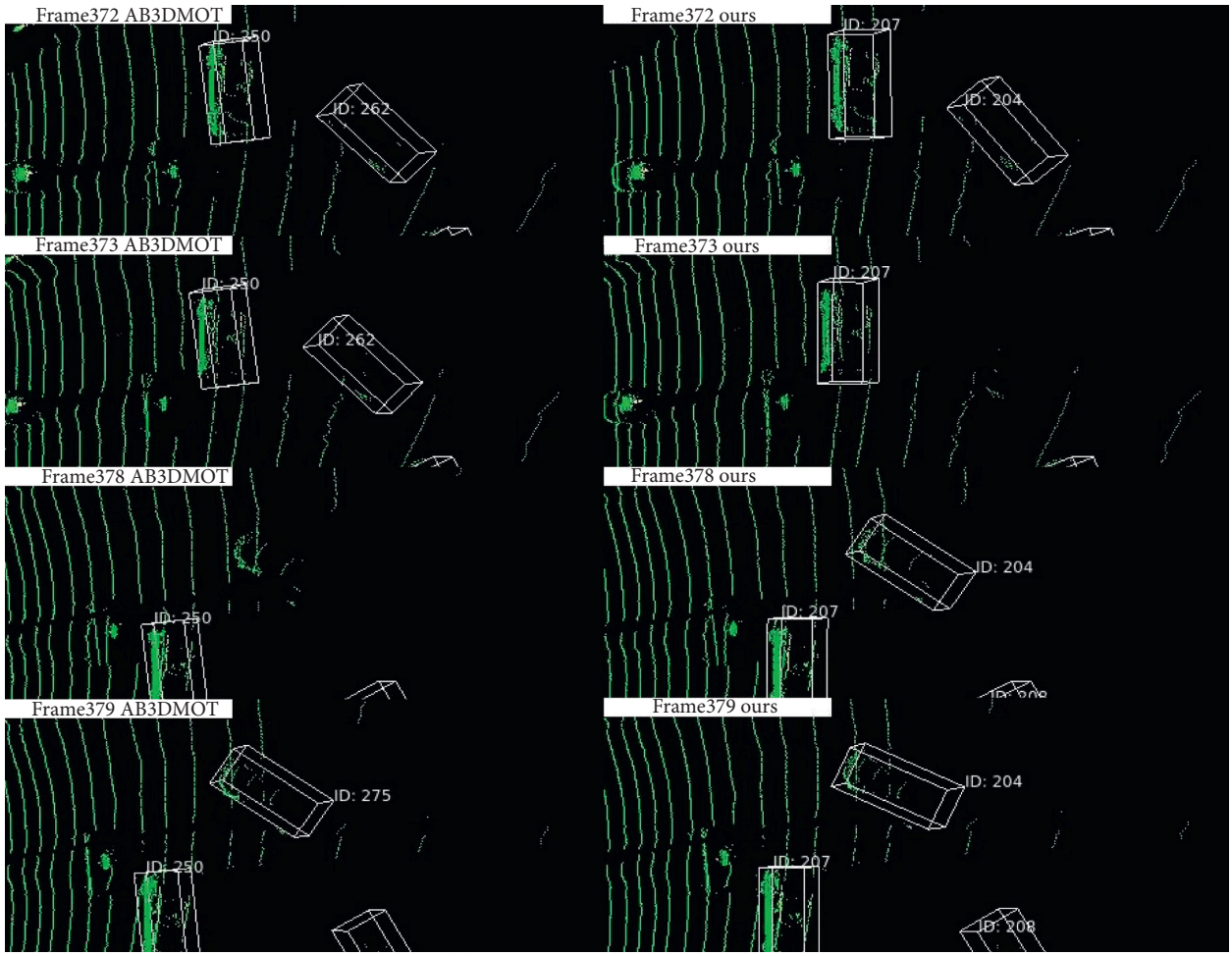

Figure 8: Effect comparison of frame 372 to frame 379 in Sequence 1.

images, and camera parameters. Since there is no label information in the KITTI train set, we use 8008 frames in the training set for the test and use sequences 1, 6, 8, 10, 12, 13, 14, 15, 16, 18, and 19 in reference [52] to validate it. In order to train the point cloud reidentification network, we use tags in sequence 0, 2, 3, 4, 5, 7, 9, 11, and 20 to extract 354 tracks and convert them into the three-channel image composed of BEV, intensity, and density maps. Partial data of the reidentification network in training are shown in Figure 6.

Tables 1 and 2 are comparison results of tracking experiments using object detection results provided by AB3DMOT. Due to the lacking labels of the roadside 32-line LiDAR dataset, this paper only shows the comparative recognition effect.

It can be seen from Tables 1 and 2 that our method has better performance than the FANTract method. Since our method is mainly used on the side of the road, there are a lot of occlusion and reappearance problems, which rarely occurs in the KITTI dataset. Therefore, the advantages of our method cannot be reflected in the KITTI dataset, which is slightly lower than those in AB3DMOT. In order to prove that our method can match the original trajectory and reflect the advantage of the reidentification network, we compare the occlusion in frame 354 to 360 and frame 372 to 379 in the first sequence of the KITTI dataset. In Figure 7, the vehicle id 222 in AB3DMOT jumps to 252 after blocking, while the id number of our method remains at 187 after blocking. In Figure 8, the vehicle id 262 of frame 372 in the AB3DMOT method reappears to be 275 after occlusion, while our method keeps the id 204 all the time.

Figure 9 is a segment of the roadside data. In our method, the id of the two objects with id numbers 4004 and 3985 remain unchanged after occlusion, while the corresponding vehicles id switching occur in the AB3DMOT method. No matter if it is KITTI data or roadside data, our

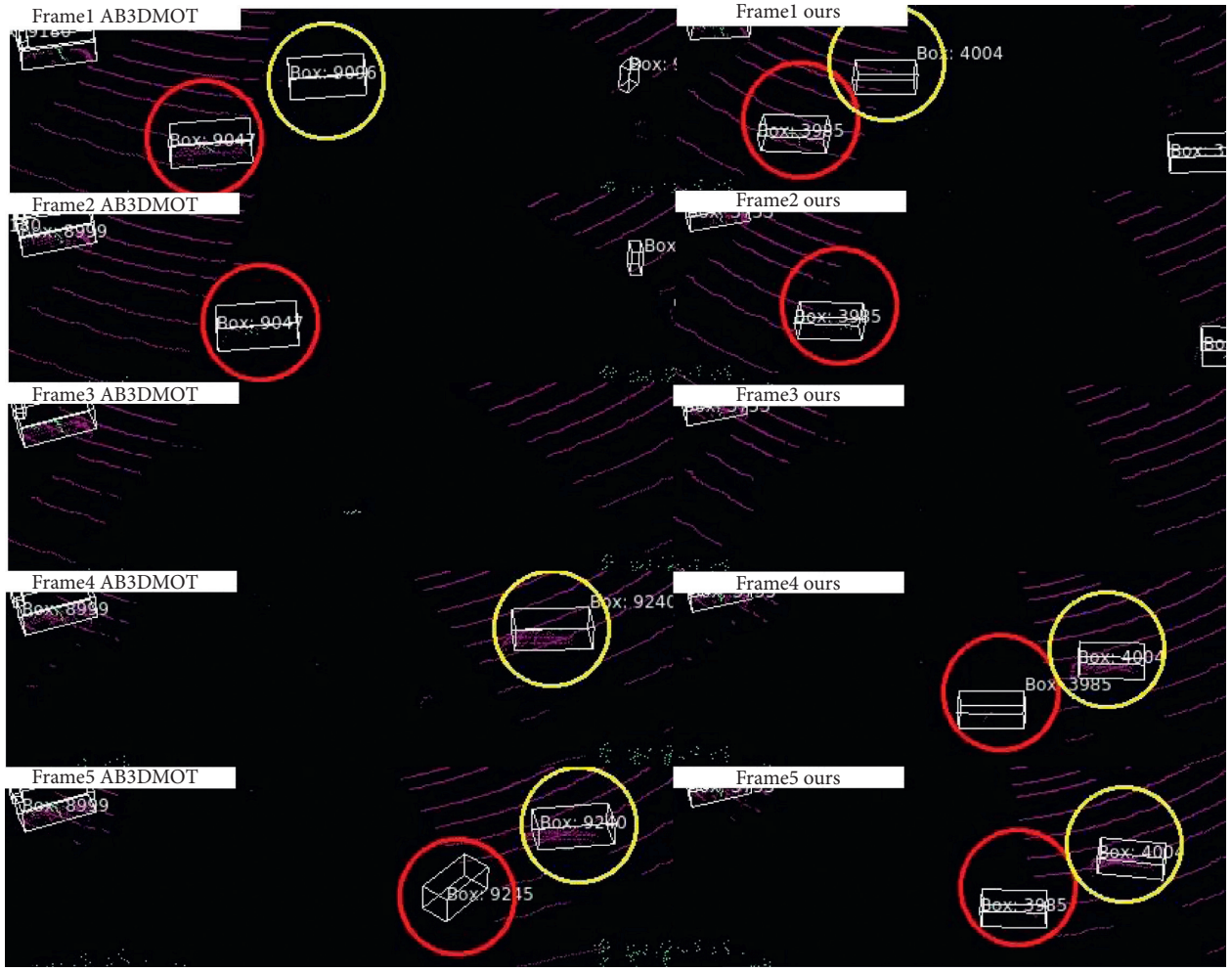

Figure 9: Effect comparison of roadside data.

method can keep the id number after occlusion, which reflects the advantage of the reidentification method in matching by features when lacking distance information.

5. Conclusions

This paper introduced re-identification algorithm into point cloud tracking algorithm based on 2D MOT, and proposed 3D MOT algorithm based on deep learning. We use the object detector to obtain the 3D boundary box of the target, and then, use the 3D Kalman filter to estimate state, combining with the re-identification algorithm to match feature similarity, and finally use the Hungarian algorithm for data association. On the KITTI dataset, our approach achieves competitive results, and on the roadside dataset, our approach is more prominent. It is believed that our method can be widely used in self-driving and roadside assisted driving.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research was supported by the National Key R&D Program for the 13th-Five-Year Plan of China (2018YFF0300305 in 2018YFF0300300).

References

[1] R. Girshick, J. Donahue, T. Darrell et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, Columbus, OH, USA, June 2014.

[2] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Washington, DC, USA, December 2015.

[3] S. Ren, K. He, R. Girshick et al., "Faster r-cnn: towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing*

Systems (NIPS), pp. 91–99, Montreal, Canada, December 2015.

[4] J. Redmon, S. Divvala, R. Girshick et al., "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[5] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[6] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[7] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 21–37, Amsterdam, The Netherlands, October 2016.

[8] S. Zhang, L. Wen, X. Bian et al., "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4203–4212, Salt Lake City, Utah, USA, June 2018.

[9] Z. Cai and N. Vasconcelos, "Cascade r-cnn: delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6154–6162, Salt Lake City, Utah, USA, June 2018.

[10] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750, Munich, Germany, September 2018.

[11] K. Duan, S. Bai, L. Xie et al., "Centernet: keypoint triplets for object detection," in *Proceedings of the IEEE International Conference On Computer Vision (ICCV)*, pp. 6569–6578, Seoul, South Korea, November 2019.

[12] Y. H. Lee, H. Ahn, H. B. Ahn et al., "Visual object detection and tracking using analytical learning approach of validity level," *Intelligent Automation and Soft Computing*, vol. 25, no. 1, pp. 205–215, 2019.

[13] B. Li, J. Yan, W. Wu et al., "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8971–8980, Salt Lake City, Utah, USA, June 2018.

[14] J. Xu, Y. Cao, Z. Zhang et al., "Spatial-temporal relation networks for multi-object tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3988–3998, Seoul, South Korea, November 2019.

[15] Q. Wang, L. Zhang, L. Bertinetto et al., "Fast online object tracking and segmentation: a unifying approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1328–1338, Long Beach, CA, USA, June 2019.

[16] B. Li, W. Wu, Q. Wang et al., "Siamrpn++: evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4282–4291, Long Beach, CA, USA, June 2019.

[17] Y. T. Chen, J. Wang, S. J. Liu et al., "The multi-scale fast correlation filtering tracking algorithm based on a features fusion model," *Concurrency and Computation: Practice and Experience*, 2019.

[18] J. Zhang, X. Jin, J. Sun, J. Wang, and A. K. Sangaiah, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 15095–15115, 2020.

[19] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3569–3577, Salt Lake City, Utah, USA, June 2018.

[20] S. Casas, W. Luo, and R. Urtasun, "Intentnet: learning to predict intention from raw sensor data," in *Proceedings of the Conference on Robot Learning (CoRL)*, pp. 947–956, Zürich, Switzerland, October 2018.

[21] A. Bewley, Z. Ge, L. Ott et al., "Simple online and realtime tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, Phoenix, AZ, USA, September 2016.

[22] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of the2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, Beijing, China, September 2017.

[23] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6036–6046, Salt Lake City, Utah, USA, June 2018.

[24] S. Tang, M. Andriluka, B. Andres et al., "Multiple people tracking by lifted multicut and person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3539–3548, Honolulu, HA, USA, July 2017.

[25] Y. Chen, J. Wang, R. Xia, Q. Zhang, Z. Cao, and K. Yang, "The visual object tracking algorithm research based on adaptive combination kernel," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4855–4867, 2019.

[26] P. K. Santhosh and B. Kaarthick, "An automated player detection and tracking in basketball game," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 625–639, 2019.

[27] H. Su, S. Maji, E. Kalogerakis et al., "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953, Washington, DC, USA, December 2015.

[28] Z. Yang and L. Wang, "Learning relationships for multi-view 3D object recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 7505–7514, Seoul, South Korea, November 2019.

[29] C. R. Qi, H. Su, M. Nießner et al., "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5648–5656, Las Vegas, NV, USA, June 2016.

[30] Y. Feng, Z. Zhang, X. Zhao et al., "Group-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 264–272, Salt Lake City, Utah, USA, June 2018.

[31] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," 2019, https://arxiv.org/pdf/1906.01592.pdf.

[32] Z. Wu, S. Song, A. Khosla et al., "3D shapenets: a deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, Boston, MA, USA, June 2015.

[33] T. Le and Y. Duan, "Pointgrid: a deep network for 3d shape understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9204–9214, Salt Lake City, Utah, USA, June 2018.

[34] X. Weng and K. M. Kitani, "A Baseline for 3D MOT," 2019, https://arxiv.org/abs/1907.03961.

[35] A. Geiger, P. Lenz, and R. Urtasun *"Are we ready for autonomous driving? The KITTI vision benchmark suite," in*

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354–3361, Providence, RI, USA, June 2012.

[36] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," 2016, https://arxiv.org/abs/1608.07916.

[37] M. Simon, K. Amende, A. Kraus et al., "Complexer-YOLO: Real-time 3D object detection and tracking on semantic point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, p. 0, Long Beach, CA, USA, June 2019.

[38] Y. Zhou and O. Tuzel, "Voxelnet: end-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4490–4499, Salt Lake City, Utah, USA, June 2018.

[39] Y. Yan, Y. Mao, and B. Li, "SECOND: sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, pp. 3337–3353, 2018.

[40] C. R. Qi, H. Su, K. Mo et al., "PointNet: deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, Honolulu, HI, USA, July 2017.

[41] C. R. Qi, L. Yi, H. Su et al., "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 5099–5108, Long Beach, CA, USA, December 2017.

[42] S. Shi, C. Guo, L. Jiang et al., "Pv-rcnn: point-voxel feature set abstraction for 3D object detection," in *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 10529–10538, Seattle, DC, USA, June 2020.

[43] X. Chen, H. Ma, J. Wan et al., "Multi-view 3D object detection network for autonomous driving," in *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 1907–1915, Honolulu, CA, USA, July 2017.

[44] J. Ku, M. Mozifian, and J. Lee, "Joint 3d proposal generation and object detection from view aggregation," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, Madrid, Spain, October 2018.

[45] Z. Wang and K. Jia, "Frustum convnet: sliding frustums to aggregate local point-wise features for amodal 3D object detection," 2019, https://arxiv.org/abs/1903.01864.

[46] A. Osep, W. Mehner, M. Mathias et al., "Combined image- and world-space tracking in traffic scenes," in *Proceedings of the 2017 IEEE International Conference on Robotics And Automation*, pp. 1988–1995, Marina Bay, Singapore, Singapore, June 2017.

[47] E. Baser, V. Balasubramanian, P. Bhattacharyya et al., "3D MOT with feature association network," in *Proceedings of the 2019 IEEE Intelligent Vehicles Symposium*, pp. 1426–1433, Paris, France, June 2019.

[48] H. N. Hu, Q. Z. Cai, D. Wang et al., "Joint monocular 3D vehicle detection and tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5390–5399, Seoul, South Korea, November 2019.

[49] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3D tracking by detection," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 635–642, Prague, Czech Republic, August 2018.

[50] W. Zhang, H. Zhou, S. Sun et al., "Robust multi-modality multi-object tracking," in *Proceedings of the IEEE International Conference On Computer Vision (ICCV)*, pp. 2365–2374, Seoul, South Korea, November 2019.

[51] A. Shenoi, M. Patel, J. Y. Gwak et al., "A real-time 3D multi-object tracker and a new large-scale dataset," 2020, https://arxiv.org/pdf/2002.08397.

[52] S. Scheidegger, J. Benjaminsson, E. Rosenberg et al., "Mono-camera 3D MOT using deep learning detections and pmbm filtering," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium*, pp. 433–440, Jiangsu, China, June 2018.