2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

FusionTrack: An Online 3D Multi-object Tracking Framework Based on Camera-LiDAR Fusion

Weizhen Zeng, Jiaqi Fan, Xuelin Tian, Hongqing Chu*, Bingzhao Gao

***Abstract*— 3D multi-object tracking is an important component of the perception module in autonomous driving systems. Due to the limitations of a single sensor, tracking methods based on either LiDAR or cameras always have certain deficiencies. Fusion-based tracking methods have received increasing attention. However, existing fusion-based tracking methods often underutilize image information, ignore the respective effects of appearance information and 2D detection results, and lack further analysis on the simultaneous use of both. This paper proposes a novel camera-LiDAR fusion tracking framework that primarily relies on the motion model using 3D objects. It fully leverages the appearance information and 2D detection results simultaneously from images and introduces three modules to reduce the number of false positive samples, false negative samples and ID switches, respectively. Besides, the entire tracking process does not require global processing and achieves online tracking. The proposed method achieves competitive results on the KITTI tracking dataset with 78.50% HOTA. Compared with EagerMOT using the same 3D and 2D detectors, the HOTA metric improved by 4.11%. Code is available on https://github.com/zengwz/FusionTrack.**

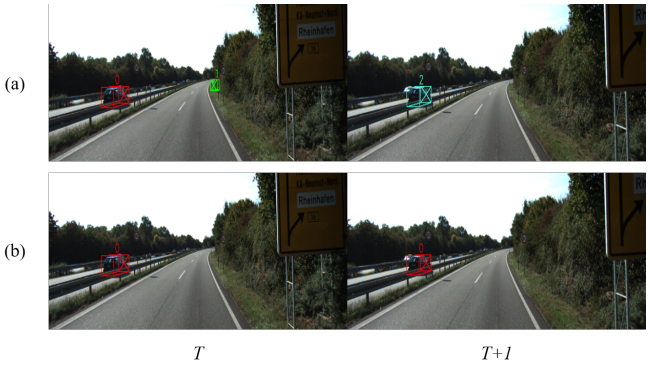

Fig. 1. **Difference between LiDAR-only method using the motion model and the proposed method.** (a): without image information, the green box with ID 1 is a false positive sample, and in frame T+1, the red object with ID 0 fails to match, resulting in a new trajectory with ID 2. (b): with the proposed method, the appearance information is utilized, successfully matching the trajectory with ID 0 between the two frames, while filtering out the false positive with ID 1 in (a) using 2D detection results.

I. INTRODUCTION

Multi-object tracking (MOT) plays a crucial role in autonomous vehicles by associating objects across consecutive frames and outputting their trajectories as inputs to downstream modules. Early research in multi-object tracking focused on the 2D domain using images as data. But the field of autonomous driving demands 3D MOT. relying solely on cameras often makes it challenging to achieve precise object detection and tracking in 3D space. Images captured by monocular cameras lack precise geometric information, such as distance and shape, which are crucial for achieving accurate 3D perception. Compared with cameras, LiDAR provide precise 3D positioning of objects, making it the mainstream sensor in autonomous driving. In 3D MOT, even with only LiDAR as the sensor, a motion model based on Kalman filter, such as [4], can effectively perform tracking tasks. However, multiple studies [18], [13], [14] indicate that image information can also play an important auxiliary role in 3D MOT. In [18], images can provide appearance information. Appearance features of objects can be extracted using appearance extraction networks, and appearance similarity can be computed as input for matching. Besides, images can provide 2D detection results. In [13], 3D detected objects are projected onto the 2D image plane, and different modalities of detection results are classified and merged based on IoU, enabling the utilization of 2D detection results to improve 3D tracking performance.

However, these methods in their respective frameworks fail to simultaneously utilize both appearance information and 2D detection results from images. In fact, these two types of information from images play different roles in 3D tracking, and they can be utilized simultaneously to improve different scenes. Therefore, this paper proposes a camera-LiDAR fusion tracking method that simultaneously exploits appearance information and 2D detection results. Compared with other fusion-based methods using 2D detection results, our method is more targeted and interpretable. It introduces appearance similarity matching module (ASMM), detection filtering module (DFM), and trajectory recovery module (TRM) to address different scenarios. ASMM is designed for pairs that can not be matched by geometric distance. DFM and TRM are designed for false and missed detections of the 3D detector. Moreover, the proposed method does not require global processing, taking the detection results of the current frame as input and directly outputting the final tracking results.

In summary, the contributions of our work are as follows:

- We construct an online 3D multi-object tracking framework that does not require global processing and fuses camera and LiDAR information on the paradigm of tracking-by-detection.
- We design a secondary matching strategy based on ap-

This work was supported in part by the National Nature Science Foundation of China (No. 62373289), and the Fundamental Research Funds for the Central Universities.

*Corresponding author: Hongqing Chu (chuhongqing@tongji.edu.cn).
All authors are with the School of Automotive Studies, Tongji University, Shanghai 201804, China.

pearance similarity using appearance information from images to address scenarios where geometric distance similarity alone is insufficient.
- We utilize 2D detection results obtained from images and propose detection filter module and trajectory recovery module to improve the detection results during the tracking process, indirectly enhancing the final tracking performance.

II. RELATED WORK

Camera-based 3D MOT. Since implicit depth estimation is required for camera-based 3D MOT, those methods are mainly implemented using deep learning. In camera-based 3D MOT, QD-3DT [6] builds upon QDTrack [2] and incorporates an LSTM network to predict the motion of objects. Deft [7] uses a common backbone network for the detection branch and tracking branch, and adds a matching head to output the final tracking results. Mutr3d [8], PFTrack [9] leverage the track query proposed by MOTR [3] and achieve the end-to-end 3D MOT in the context of omnidirectional cameras. Overall, due to the challenges in depth estimation with cameras, the perception performance achieved by cameras is noticeably inferior to methods based on LiDAR perception in 3D space.

LiDAR-based 3D MOT. With precise 3D localization of objects and the lack of semantic information in point clouds, most LiDAR-based methods rely on the motion model. AB3DMOT [4] proposes a fundamental 3D tracking framework based on LiDAR using Kalman filter for trajectory prediction and update, and the Hungarian algorithm for matching. [10] amplifies the search range for unmatched trajectories multiple times and assigns trajectory confidence as weights to the geometry distance similarity matrix, reducing uncertainty introduced by multiple consecutive frames not being updated. CenterPoint [5] predicts the 3D detection results and the displacement of objects between two adjacent frames for tracking simultaneously. SimpleTrack [11] analyses every module of the overall 3D MOT framework and makes several simple modifications on every module. PolyMOT [12] applies different motion models and hyperparameter thresholds based on specific object categories, allowing for the selection of the most suitable motion model for each category. Due to the lack of semantic information, those methods are difficult to finish tracking for specific scenes with the motion model.

Camera-LiDAR Fusion-based 3D MOT. In fusion-based 3D MOT, the image mainly provides appearance information and 2D detection results. EagerMOT [13], DeepFusionMOT [14] utilize a 2D detector to obtain 2D detection results from images and project 3D detection results onto the image plane. They divide all detection objects into 2D bounding boxes, 3D bounding boxes and common detection boxes based on IoU, and then match them with trajectories respectively. StrongFusionMOT [15] incorporates depth information to enhance the fusion of 2D and 3D detections, improving the robustness of the fusion process. MOTSFusion [17] fuses optical flow, detection results, depth maps and camera ego-motion to finish 2D segment tracking and 3D reconstruction tracking. mmMOT [16] achieves end-to-end fusion tracking framework by deep learning and encodes deep representation of point clouds to match. CAMO-MOT [18] extracts appearance information of 3D detections and incorporates an occlusion detection head to assist in selecting appearance features. Additionally, GNN3DMOT [19] achieves end-to-end fusion tracking using LSTM and GNN. It employs an LSTM to capture the features of consecutive frames in point clouds and images and utilizes GNN to compute the final matching loss matrix.

III. METHOD

The overall proposed tracking algorithm framework is shown in Fig. 2 and can be divided into three parts: data input, data association and trajectory management. 3D and 2D detection results and appearance features are obtained from point clouds and images for the data association. The data association is a two-stage matching process inserting DFM and TRM for tracking improvement. Finally, a common trajectory management method is utilized to manage pairs, unmatched detections and trajectories resulting from the data association.

A. Data Input

The input data consists of the point clouds and images. Common 3D detectors are used to obtain 3D object detection results. Detection results are filtered by the confidence score θ_{SF} and serve as input of data association. The 3D detection results are projected onto the image plane to extract appearance information for ASMM. Additionally, the images are processed by a 2D detection network to obtain detection results for DFM and TRM.

B. Data Association

After the data input module, we can obtain the 3D and 2D detection results and appearance features of 3D objects. In the data association module, the first step is to calculate the geometric distance similarity matrix using the constant acceleration (CA) motion model. The motion model is built upon Kalman filter and the state of the trajectory can be represented by a 13-dimensional vector $T_{Tra} = [x, y, z, v_x, v_y, v_z, a_x, a_y, a_z, l, w, h, \theta]$. The state of detection results can be represented by a 7-dimensional vector $T_{Det} = [x, y, z, l, w, h, \theta]$, which is the output format of the 3D detector. The distance metric is calculated using 3D gIoU. The input of the 3D detector is presented in the LiDAR coordinate system and we use GPS/IMU data and relative transformation matrixes to convert the 3D detection results to the global coordinate system.

Detection Filter Module (DFM). After the first matching, there are two possibilities for unmatched detections D_{3d}^{1u}: false positive detections and true newly appearing trajectories. To avoid generating many false positive samples, different methods can be utilized for this situation. One approach is to set a sensitive state for the trajectory, where newly

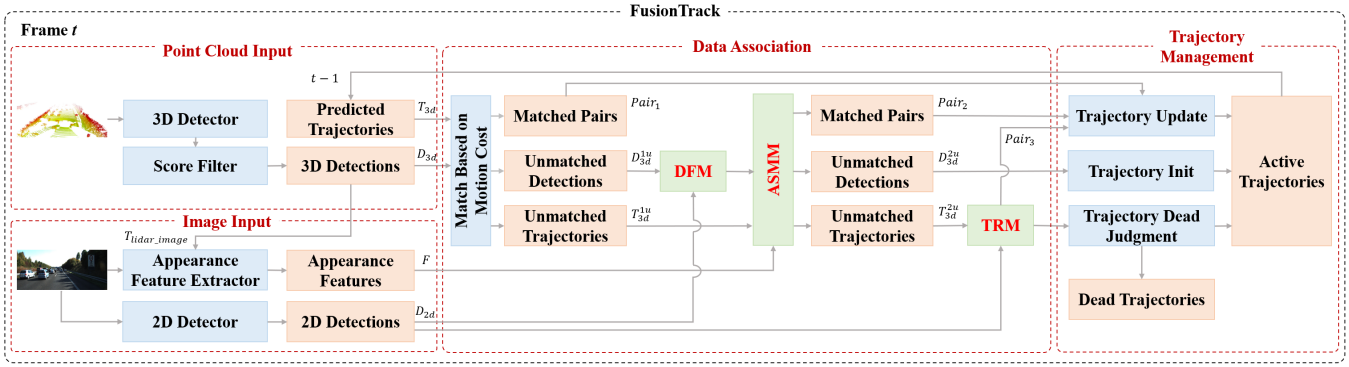

Fig. 2. **The pipeline of the proposed method.** 1) Input the 3D detections D_{3d} and the predicted states T_{3d}, and geometric distance similarity is computed using gIoU. the first matching results are matched pairs $Pair_1$, unmatched 3D detections D_{3d}^{1u}, and unmatched trajectories T_{3d}^{1u}. 2) D_{3d}^{1u} are projected onto the image plane as $D_{3d\_2d}^{1u}$. DFM is used to reduce the number of FP by matching $D_{3d\_2d}^{1u}$ and 2D detections D_{2d} using IoU. 3) The appearance features F are used to compute appearance similarity using the dot product. The second matching results are matched pairs $Pair_2$, unmatched 3D detections D_{3d}^{2u}, and unmatched trajectories T_{3d}^{2u}. 4) T_{3d}^{2u} is projected onto the 2D image plane as T_{2d}^{1u}. TRM is used to reduce the number of FN by matchcing T_{2d}^{1u} and D_{2d} using IoU. The recovered trajectories are $Pair_3$. 5) $Pair$ is the union of $Pair_1$, $Pair_2$ and $Pair_3$. In the trajectory management module, $Pair$ is used to update the trajectory states through Kalman filter. D_{3d}^{2u} is used to generate new trajectories. If T_{3d}^{2u} fails to match an object for multiple consecutive frames Age_{max}, this trajectory will be deleted. 6) The output of the tracking results are updated $Pair$ and the new trajectories from D_{3d}^{2u}.

appearing trajectories need to be continuously matched for several frames before being converted to active state and output as final tracking results. Another approach is global processing, where trajectories with a life cycle of only 1 or 2 frames are considered false positive samples and directly removed. The first approach introduces latency, resulting in the true newly appearing trajectories can not be outputted immediately. The second approach requires global processing and cannot achieve online tracking. Therefore, DFM is proposed here to reduce the number of false positive samples for online tracking. Unmatched detections are projected onto the image plane and every projected detection's IoU is computed with all 2D detection results D_{2ds}. If the maximum value of IoU is below the filter threshold θ_{DFM}, this unmatched detection is considered a false positive sample and is directly removed instead of generating a new trajectory. Otherwise, this unmatched detection is used for the second matching. The process can be presented as follows:

$$D_{3d\_2d}^{1u} = f_{project}(D_{3d}^{1u}), \tag{1}$$

$$u_{3d\_2d} = max(IoU(D_{3d\_2d}^{1u}, D_{2d})), \tag{2}$$

$$D_{3d}^{1u} = \begin{cases} D_{3d}^{1u} & if \quad u_{3d\_2d} > \theta_{DFM} \\ delete & if \quad u_{3d\_2d} \leq \theta_{DFM} \end{cases}, \tag{3}$$

where $f_{project}(\cdot)$ is the function used for transformation between different coordinate systems.

Appearance Similarity Matching Module (ASMM). After DFM, the second matching based on appearance similarity is performed to re-associate D_{3d}^{1u} and T_{3d}^{1u} that couldn't be matched based on geometric distance similarity. The training pipeline of appearance feature extraction network is shown in Fig. 3. In training phase two images are inputted at the same time. We use ResNet [23] and FPN [24] to extract the feature map of images. RoIAlign [25] is used to extract the the appearance features corresponding to the 3D detection results in the feature map. Inspired by PFTrack [9], cross-object attention is utilized to enhance the feature embedding for more distinctive representation. Finally, we utilize the contrastive learning in [2] to train our model.

In the inference phase, The appearance similarity is computed using the dot product of appearance embeddings and normalized using Softmax. Additionally, we set the Euclidean distance threshold θ_{eucl} for filtering the impossible pairs, reducing the interference for this appearance-based matching.

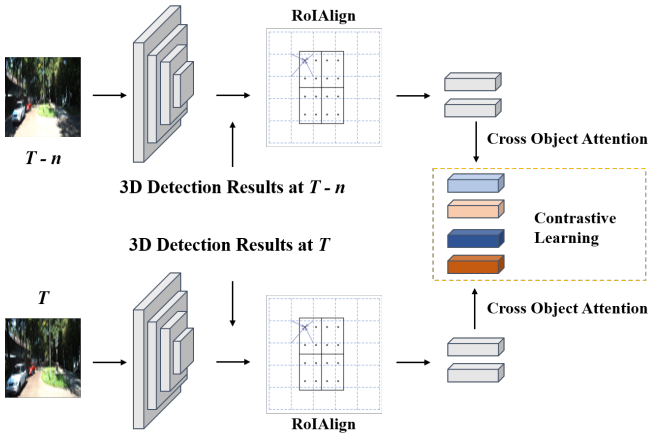

Fig. 3. **The training pipeline of the appearance feature extraction network.**

Trajectory Recovery Module (TRM). After ASMM, TRM is introduced to address the problem of missed detections caused by the 3D detector. For unmatched trajectories T_{3d}^{2u} after two matches, there are two possibilities: the trajectory has left the ego-vehicle's field of view, or the 3D detector misses the corresponding objects of the

trajectory. For the second case, A common global processing method is to check whether the trajectory is completely continuous after finishing tracking and use the predicted state to complete the fragment. For online tracking, TRM projects the predicted state of the trajectory onto the image plane and computes its IoU with D_{2ds}. If there exists a 2D detection result with an IoU above the recovery threshold θ_{TRM}, it is considered an object belonging to this trajectory missed by the 3D detector but detectable by the 2D detector. Since 2D detection results lack 3D information, they can not directly update 3D trajectory information. Therefore, TRM utilizes the 3D predicted state of the recovered trajectory to output the tracking result and update. The whole process can be presented as follows:

$$T_{3d\_2d}^{2u} = f_{project}(T_{3d}^{2u}), \tag{4}$$

$$u_{3d\_2d} = max(IoU(T_{3d\_2d}^{2u}, D_{2d})), \tag{5}$$

$$T_{3d}^{2u} = \begin{cases} recovered & if \quad u_{3d\_2d} > \theta_{TRM} \\ T_{3d}^{2u} & if \quad u_{3d\_2d} \leq \theta_{TRM} \end{cases}, \tag{6}$$

The final tracking results consist of the trajectories matched in both matches, newly generated trajectories and the trajectories recovered by TRM.

C. Trajectory Management

This module adopts a simple strategy to manage the lifecycle of trajectories. Trajectories that have not been matched with detections for multiple consecutive frames are deleted. For trajectories successfully matched, their state vectors are updated using Kalman filter, and their appearance feature embeddings are updated using the exponential moving average (EMA). For all trajectories existing in the lifecycle, Kalman filter is utilized for prediction to obtain their predicted states, which will be used to compute similarity with the 3D detection results in the next frame.

IV. EXPERIMENTS

A. Dataset

We use the KITTI tracking dataset to verify our method. The training set of this dataset consists of 21 sequences with a total of 8,008 frames, and the test set consists of 29 sequences with a total of 11,095 frames. According to [13], [18], sequences 1, 6, 8, 10, 12, 13, 14, 15, 16, 18, 19 from the training set are used as the validation set, and the remaining sequences are used for training our model. The primary reference metric is the Higher-Order Tracking Accuracy (HOTA). Other metrics will also be listed for further analysis.

B. Implementation Details

For the ablation studies and result visualization on the validation set, we used CASA [20] as the 3D object detector and RRC [22] as the 2D object detector. We also test other 3D detectors' performance on the validation set. For a fair comparison on the KITTI 2D tracking benchmark, our submitted version uses Point-GNN [21] as the 3D object detector, same as [13]. The maximum age of a trajectory Age_{max} is set to 15. θ_{giou}, θ_{SF}, θ_{ASMM}, θ_{DFM}, θ_{TRM}, θ_{eucl} is set to -0.2, 0.5, 0.5, 0.6, 0.6, 10.0m. For the appearance feature extraction network, we train it with RTX 3080 GPU. During training, the network is trained for 40 epochs using the SGD optimizer (momentum is 0.9, decay rate is 0.0001) with a batch size of 4. The learning rate is $2.5e^{-4}$ for the first 20 epochs, $2.5e^{-5}$ for the last 20 epochs.

C. Experimental Results

We report our results on the KITTI 2D tracking benchmark in Table I and compare it with other public methods. Our method achieves a HOTA of 78.50%, an IDS of 46 and a FP of 159. Compared with EagerMOT [13], which uses the same 3D and 2D detector results as input, our method achieves a +4.12% HOTA improvement, a -193 IDS reduction and a -3338 FP reduction. However, the significant reduction in false positive (FP) leads to a significant increase in false negative (FN). Our method has 3383 false negative samples, much higher than other public methods. This is because for 3D detection results that are not matched in the first time, if they are not detected by the 2D detector, they will be directly removed, resulting in an increase in the number of false negative. A better-performing 2D detector may help improve this issue. Overall, although the balance between FP and FN samples is not ideal, the great improvement in primary metric HOTA demonstrates the effectiveness of our proposed method.

D. Ablation Studies

The Effect of Detection Filter Module. As shown in Table II, "MO+DFM" demonstrates the significant contribution of using the DFM alone. Compared with "MO", DFM leads to a significant reduction in FP (-561), at the cost of a slight increase in FN (+31). Considering the significant reduction in FP samples achieved by DFM, the minor increase in FN is acceptable, and the final HOTA improves by +2.85%.

The Effect of Appearance Similarity Matching Module. "MO+ASMM" demonstrates the effect of using ASMM alone in Table II. Compared with "MO", ASMM leads to a significant reduction in the number of IDS (-28) and an improvement of +0.47% HOTA. This proves that incorporating appearance similarity for matching after the motion model-based matching can effectively reduce the number of IDS and enhance overall tracking performance.

The Effect of Trajectory Recovery Module. As shown in Table II, "MO+TRM" demonstrates the effect of using TRM alone. After incorporating TRM, the number of FN is significantly reduced (-108). This comparison demonstrates

TABLE I

A COMPARISON OF EXISTING METHODS APPLIED TO THE KITTI TRACKING BENCHMARK TEST SET (CAR CLASS). METHODS LABELED BY "*" USE THE SAME DETECTOR(POINT-GNN [21] AND RRC [22]).

Method	Modality	Type	HOTA↑	MOTA↑	MOTP↑	FN↓	FP↓	IDS↓	MT↑	ML↓
AB3DMOT [4]	LiDAR	3D	69.81	83.49	85.17	1060	4492	126	67.08	11.38
mono3DT [26]	Camera	3D	73.16	84.28	85.45	745	4282	379	73.08	2.92
DEFT [7]	Camera	2D	74.23	88.38	84.46	1006	2647	344	84.31	**2.15**
EagerMOT* [13]	LiDAR&Camera	3D	74.39	87.82	**85.69**	454	3497	239	76.15	2.46
DeepFusionMOT [14]	LiDAR&Camera	3D	75.46	84.63	85.02	601	4601	84	68.61	9.08
StrongFusionMOT [15]	LiDAR&Camera	3D	75.65	85.53	85.07	259	4658	58	66.15	6.00
OC-SORT [1]	Camera	2D	76.54	90.28	85.53	407	2685	250	80.00	3.08
PC3T [27]	LiDAR	3D	77.80	88.81	84.26	814	2810	225	80.00	8.46
PermaTrack [28]	Camera	2D	78.03	**91.33**	85.65	**402**	2320	258	**85.69**	2.62
Ours*	LiDAR&Camera	3D	**78.50**	89.57	85.51	3383	**159**	**46**	76.31	3.85

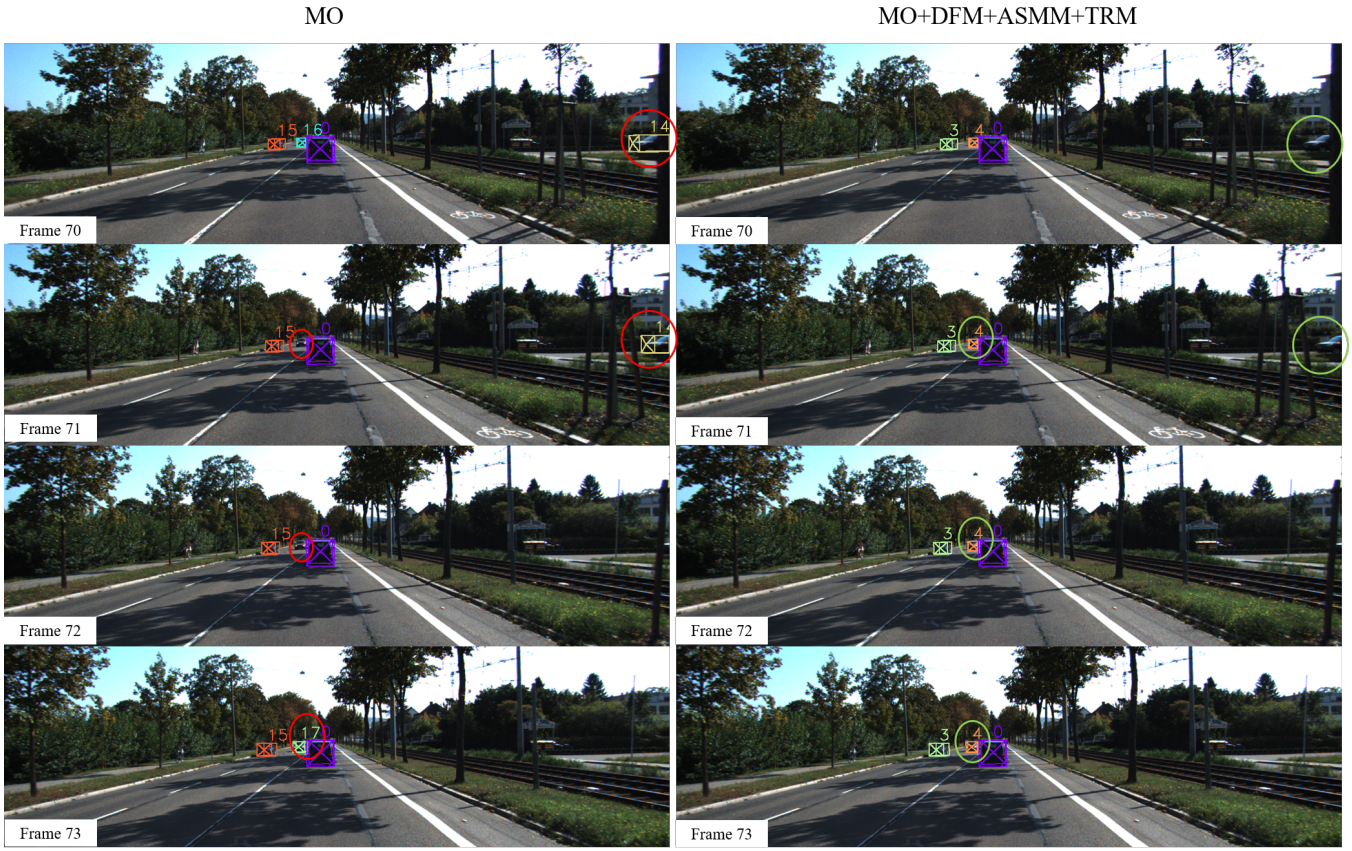

Fig. 4. **Visualization of comparison results between MO and Ours.** Those images are from frames 70 to 73 of sequence 10 in the KITTI tracking validation set. The red circles represent the areas where errors occurred by "MO", and the green circles represent the same areas improved by the proposed method.

that using TRM can complete the 3D detector's missed objects at the corresponding positions of trajectories, reducing the number of FN. However, due to the balance between FN and FP samples, reducing one will inevitably increase the other, increasing of 18 FP samples in this case. The final HOTA improves by +0.54% compared with "MO".

The results of ablation studies demonstrate that the DFM, TRM, and ASMM modules address the issues of false positive and false negative results of the 3D detector and the failure of geometric distance similarity-based matching. They significantly reduce the numbers of FP, FN, and ID switches respectively and have strong interpretability. When all three modules are used together ("MO+DFM+ASMM+TRM"), the FP, FN, and IDS are significantly reduced at the same time. In our carefully designed tracking framework, there is no interference among the three proposed modules. The final HOTA improves by +4.17% compared with "MO".

E. Visualization

We find a sequence of images for visualization, which can fully demonstrate the effects of the three proposed modules, as shown in Fig. 4. The first column of Fig. 4 represents the

TABLE II

THE RESULTS OF THE ABLATION STUDY OF EACH MODULE ON THE KITTI TRACKING VALIDATION SET. "MO" MEANS TRACKING WITH THE MOTION MODEL BUILT BASED ON [4]. 3D DETECTOR IS CASA [20].

Method	HOTA↑	FN↓	FP↓	IDS↓
MO	81.12	486	733	34
MO + DFM	83.97	517	**172**	32
MO + ASMM	81.59	486	736	**6**
MO + TRM	81.66	**378**	750	30
MO + ASMM + DFM	84.51	518	171	**6**
MO + ASMM + DFM + TRM	**85.19**	416	175	**6**

visualization of tracking results using only "MO", where the problematic tracking results are highlighted in red circles. In frames 70 and 71, the yellow bounding box with ID 14 is a false positive that does not exist in the ground truth. In frames 71 and 72, the 3D detector loses the trajectory with ID 16. the detection result corresponding to this trajectory is re-detected in frame 73 but fails to match using geometric distance similarity and generates a new trajectory with ID 17.

In the second column of Fig. 4, the visualization of tracking results using our proposed method is shown. The green circles indicate improvements compared with the results obtained by "MO". In frames 70 and 71, the rightmost FP object is filtered out by DFM, reducing the number of FP. Then, in frames 71 and 72, TRM is used to recover the trajectory with ID 4. The predicted state of this trajectory is used to complete the missed detection, reducing the number of FN. Finally, in frame 74, even though this trajectory fails to match the re-detected object using geometric distance similarity, it successfully matches the re-detected object after using ASMM, avoiding the generation of a new trajectory.

REFERENCES

[1] J. Cao, J. Pang, X. Weng, R. Khirodkar and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9686-9696.

[2] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 164–173.

[3] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking with transformer," in *European Conference on Computer Vision*, 2022, pp. 659-675.

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

[5] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 784-11 793.

[6] H.-N. Hu, Y.-H. Yang, T. Fischer, T. Darrell, F. Yu, and M. Sun, "Monocular quasi-dense 3d object tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1, 2022.

[7] M. Chaabane, P. Zhang, J. R. Beveridge, and S. O'Hara, "Deft: Detection embeddings for tracking," arXiv, 2021. [Online]. Available: https://arxiv.org/abs/2102.02267.

[8] T. Zhang, X. Chen, Y. Wang, Y. Wang and H. Zhao, "Mutr3d: A multi-camera tracking framework via 3d-to-2d queries," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4537–4546.

[9] Z. Pang, J. Li, P. Tokmakov, D. Chen, S. Zagoruyko and YX. Wang, "Standing Between Past and Future: Spatio-Temporal Modeling for Multi-Camera 3D Multi-Object Tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17 928-17 938.

[10] 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," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5668-5677, 2021.

[11] Z. Pang, Z. Li and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," in *European Conference on Computer Vision*, 2022, pp. 680-696.

[12] X. Li, T. Xie, D. Liu, J. Gao, K. Dai, Z. Jiang, L. Zhao and K. Wang, "Poly-mot: A polyhedral framework for 3d multi-object tracking," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 9391-9398.

[13] A. Kim, A. Ošep and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 315–11 321.

[14] X. Wang, C. Fu, Z. Li, Y. Lai and J. He, "DeepFusionMOT: A 3D multi-object tracking framework based on camera-LiDAR fusion with deep association," in *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8260-8267, 2022.

[15] X. Wang, C. Fu, J. He, S. Wang and J. Wang, "StrongFusionMOT: A Multi-Object Tracking Method Based on LiDAR-Camera Fusion," in *IEEE Sensors Journal*, vol. 23, no. 11, pp. 11 241 - 11 252, 2023.

[16] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2365–2374.

[17] J. Luiten, T. Fischer and B. Leibe, "Track to reconstruct and reconstruct to track," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp.1803-1810, 2020.

[18] L. Wang, X. Zhang, W. Qin, X. Li, J. Gao, L. Yang, Z. Li, J. Li, L. Zhu, H. Wang and H. Liu, "Camo-mot: Combined appearance-motion optimization for 3d multi-object tracking with camera-lidar fusion," in *IEEE Transactions on Intelligent Transportation Systems*, vol.24, no. 11, pp. 11 981 - 11 996, 2023.

[19] X. Weng, Y. Wang, Y. Man and KM. Kitani, "Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6499-6508.

[20] H. Wu, J. Deng, C. Wen, X. Li, C. Wang and J. Li, "CasA: A cascade attention network for 3-D object detection from LiDAR point clouds," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-11.

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

[22] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu, "Accurate single stage detector using recurrent rolling convolution," in *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5420-5428.

[23] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.

[24] T.Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117-2125.

[25] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask r-cnn," in *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2961-2969.

[26] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu, "Joint monocular 3d vehicle detection and tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5390-5399.

[27] 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," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5668–5677, 2022.

[28] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon, "Learning to track with object permanence," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 840–10 849.