remote sensing

MDPI

Article

Boost Correlation Features with 3D-MiIoU-Based Camera-LiDAR Fusion for MODT in Autonomous Driving

Kunpeng Zhang [1], Yanheng Liu [1,2], Fang Mei [1,2,*], Jingyi Jin [1] and Yiming Wang [1]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2] Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun 130012, China
* Correspondence: meifang@jlu.edu.cn

Abstract: Three-dimensional (3D) object tracking is critical in 3D computer vision. It has applications in autonomous driving, robotics, and human–computer interaction. However, methods for using multimodal information among objects to increase multi-object detection and tracking (MOT) accuracy remain a critical focus of research. Therefore, we present a multimodal MOT framework for autonomous driving boost correlation multi-object detection and tracking (BcMODT) in this research study to provide more trustworthy features and correlation scores for real-time detection tracking using both camera and LiDAR measurement data. Specifically, we propose an end-to-end deep neural network using 2D and 3D data for joint object detection and association. A new 3D mixed IoU (3D-MiIoU) computational module is also developed to acquire more precise geometric affinity by increasing the aspect ratio and length-to-height ratio between linked frames. Meanwhile, a boost correlation feature (BcF) module is proposed for the affinity calculation of the appearance of similar objects, which comprises an appearance affinity calculation module for similar objects in adjacent frames that are calculated directly using the feature distance and feature direction's similarity. The KITTI tracking benchmark shows that our method outperforms other methods with respect to tracking accuracy.

Keywords: autonomous driving; 3D-MOT; sensor fusion; deep neural network; feature correlation; affinity metric

Citation: Zhang , K.; Liu, Y.; Mei, F.; Jin, J.; Wang, Y. Boost Correlation Features with 3D-MiIoU-Based Camera-LiDAR Fusion for MODT in Autonomous Driving. *Remote Sens.* **2023**, *15*, 874. https://doi.org/10.3390/rs15040874

Academic Editors: Anup Basu, Chengcai Leng and Hemanth Venkateswara

Received: 21 November 2022
Revised: 19 January 2023
Accepted: 1 February 2023
Published: 4 February 2023

1. Introduction

The role of three-dimensional (3D) object tracking [1–4] has received increased attention across several disciplines in recent years, such as automatic driving, robotics, and human–computer interaction. There is a trend of equipping more sensors such as cameras, LiDAR, and radar on vehicles. Self-driving vehicles can obtain more detailed perceptual information with multiple sensors, and this in turn can result in safer and more reliable driving behaviors. Kim et al. [5] proposed a simple and effective multi-order data association method that can handle the results of different object detection algorithms and can handle data with different modalities named EagerMOT. Shenoi et al. proposed JRMOT [6] to integrate information from RGB images and 3D point clouds for real-time, state-of-the-art tracking performance. Zhang et al. proposed mmMOT [7], which is the first attempt to apply the deep features of point clouds for tracking operations. Their studies have shown that comparing a single sensor and multiple sensors resulted in the multi-sensor fusion method, which significantly improved tracking accuracy. Therefore, one of the greatest challenges in tracking objects in 3D space is to provide more accurate detection information for tracking when using multi-modal information provided by multiple sensors. A typical multi-object tracking system usually consists of several components, such as an object detector, object correlator, data association, and track management. Meanwhile, affinity metrics with robustness should combine appearance features and geometric features to

address both minor appearance differences and complex motion differences between objects. In contrast to fusion methods, substantially less information about the effects of using multimodal features obtained from multiple sensors for multi-object detection is available and has not been closely examined. Previous studies of 3D-MOT have an overemphasis on detecting the correlation of the distance between features while ignoring the correlation of the direction between features.

To summarize, our contributions are as follows:

- This paper presents an end-to-end network named boost correlation multi-object detection tracking (BcMODT). BcMODT can simultaneously generate 3D bounding boxes and more accurate association scores from camera and LiDAR measurement data for real-time detection by using the boost correlation feature (BcF);
- This paper proposes a new 3D-CIoU computing module, enhancing the fault tolerance of intersection-over-union (IoU) computing. This 3D-CIoU can handle more scenarios by using the length-to-width and length-to-height ratios of the detected bounding box and tracked bounding box;
- We combine 3D-GIoU and 3D-CIoU, named 3D mixed IoU (3D-MiIoU), instead of 3D mean IoU (3D-Mean-IoU) in [8], as the calculation method for geometric affinity, which can express the geometric affinity between objects more carefully;
- The approach is evaluated on the large autonomous driving benchmark KITTI [9], and the results show that compared with existing methods, the proposed method effectively improves the tracking accuracy, IDSW, and other evaluation metrics.

The rest of this paper is organized as follows. The related work on the MOT method is described in Section 2. Our method is presented in Section 3. Section 4 shows the experimental evaluation, analysis, and limitations of our method. The findings, conclusions, and future research work are summarized in Section 5.

2. Related Works

2.1. Multi-Object Tracking Framework

There are two basic paradigms for solving multi-object tracking (MOT) problems. One is tracking by detection (TBD), which treats detection and tracking as separate tasks. According to the methods proposed in [10–12], most current MOT methods follow the TBD paradigm. However, MOTs that follow the TBD paradigm have many problems, such as low performance and error accumulation, since object detection, object association, and data association are all in a cascading track. Therefore, to solve these problems, joint-detection tracking (JDT) [13] is trained for end-to-end learning. Wu et al. [2] proposed a new online tracking model, track to detect and segment (TraDeS), which improves multi-object tracking by feeding information from the tracking stage to the detection stage and by designing a Re-ID loss that is more compatible with the detection loss. In addition, there are already many tracking methods based on the JDT paradigm, such as CenterTrack [14], ChainedTracker [15], JDE [16], Retinatrack [13], and JMODT [17]. ChainedTracker [15] constructs tracklets by chaining paired boxes in every two contiguous frames. Zhang et al. [7] showed that using the correlation of each detection pair can improve the model's performance. Although JDT is more beneficial for the overall performance, designing its model is more difficult. Therefore, it is particularly important for the JDT paradigm to design a more reasonable model with multi-sensor information.

Thus far, it has been confirmed in [2,6,7,17] that compared with single-sensor fusion, multi-sensor fusion significantly improves tracking accuracy. A greater portion of studies on 3D-MOT-based multi-sensor information fusion emphasized the importance of the impact of sensor calibration accuracies on 3D-MOT. However, the attribute information between objects is ignored.

2.2. Affinity Metrics for Object Detection

The affinity between objects can be estimated by appearance, motion prediction, and geometric intersections. Unlike fluoroscopy-based object detection data fusion [18],

current object-level bulk feature fusion schemes use image features and LiDAR features in tandem or based on attention maps [6] to represent multi-modal features. Among them, the appearance affinity calculation mainly involves the following methods, which are only based on camera features: ODESA [19], SMAT [8], CenterTrack [14], ChainedTrack [15], JDE [16], and Retina Track [13]. Methods based on batch fusion features obtained from the camera and LiDAR include JRMOT [6], and those based on the point-wise fusion of cameras and LiDAR include JMODT [17]. However, these methods only integrate information from each modality separately and do not fully use the relationship between features.

2.2.1. Appearance Modality

To improve the accuracy of multi-object joint detection and tracking, the shared feature given by the region proposal network (RPN) [20,21] requires additional processing. JMODT [17] uses the traditional IoU to filter invalid candidate features and uses absolute subtraction [7] as the operation of candidate feature correlation to represent the objects' correlation between adjacent frames. Meanwhile, mmMOT uses point multiplication, subtraction, and absolute subtraction to represent similarities between candidate frames. Moreover, mmMOT concludes, via experiments, that absolute subtraction has the best performance in the similarity calculation of adjacent frames. In conclusion, these studies show that they only consider the distance similarity [22] of features in adjacent frames, and they do not cover the direction similarity of features between adjacent frames. Therefore, we propose a boost correlation feature module which considers the distance similarity of features in adjacent frames and the direction similarity between features. Combining the two enhances the association score measured by the camera and LiDAR, thus providing more accurate information for real-time joint detection and tracking. Table 1 shows the most advanced MOT mainstream methods in autonomous driving.

Table 1. A methodological comparison between state-of-the-art MOT and the proposed BcMOT method.

Type	Method	Object Detection and Correlation		Affinity Metrix			Data Asociation	Year
		Detection	Correlation	Appearance Modality	Motion	Geometry		
TBD [10,11]	ODESA [19]	2D	Re-ID	Camera	KF	2D IoU	HA	2020
	SMAT [8]	2D	Re-ID Optical Flow	Camera	×	2D IoU	HA	2020
	JRMOT [6]	3D	Re-ID	Camera + LiDAR (Batch Fusion)	KF	3D IoU	JPDA	2020
	mmMOT [7]	3D	Re-ID Start-End	Camera + LiDAR	×	×	MIP	2019
JDT [13–17]	CenterTrack [14]	2D/3D	Paired Detection	Camera	Offset	2D Distance	Greedy	2020
	ChainedTrack [15]	2D	Parallel Re-ID	Camera	×	2D IoU	HA	2020
	JDE	2D	Parallel Re-ID	Camera	KF	2D Distance	HA	2019
	Retina Track [13]	2D	Parallel Re-ID	Camera	KF	2D IoU	HA	2020
JDT [13–17]	JMODT [17]	3D	Parallel Re-ID Start-End	Camera + LiDAR (Point-Wise Fusion)	KF	3D DIoU	Improved MIP	2021
	BcMODT (Ours)	3D	Parallel Re-ID Start-End BcF	Camera + LiDAR (Point-Wise Fusion) BcF	KF	3D MiIoU	Improved MIP	2022

KF = Kalman filter; Offset = image-based deep offset prediction; IoU = intersection over union; DIoU = distance-IoU affinity; MiIoU = mixed-IoU affinity; HA = Hungarian algorithm; JPDA = joint probabilistic data association; MIP = mixed-integer programming.

2.2.2. Motion and Geometry

Geometrical affinity is calculated by using the intersection over union(IoU) [23,24] between two boxes. The motion relationship between objects can be represented using a variety of different metrics, such as the degree of overlap between the predicted bounding box for an object and the ground truth bounding box, the similarity of the motion patterns of the two objects, or the degree of temporal coherence between two objects. The specific metric used to represent the motion affinity will depend on the specific application and the characteristics of the objects being tracked. The Kalman filter [25,26] is the mainstream motion prediction algorithm, which has been applied in JRMOT [6], JMODT [17], and CenterTrack [14], among others. The network will predict a series of prediction boxes when testing with the trained model.

At this moment, most studies use the NMS [27] to remove some redundant boxes; that is, it is used to remove some boxes with an IoU that is greater than a certain threshold, and then the IoU with the ground truth in the remaining boxes is calculated. Generally, it is specified that the detection is correct when the IoU value of the candidate box and the ground truth is greater than 0.5. The IoU cannot accurately reflect the size of the coincidence degree. The detection effect of the same IoU is quite different. Rezatofighi et al. used the IoU to subtract the proportion of the empty area of the smallest external rectangle as the GIOU [28]. Unlike the IoU, which only focuses on overlapping areas, the GIoU not only focuses on overlapping areas but also on other non-overlapping areas, which can better reflect the coincidence degree of the two. Since the GIoU still relies heavily on the IoU, it is difficult to converge in the two vertical directions due to large errors, which is why the GIoU is unstable. Some scholars modified the penalty term of the maximized overlapping area by introducing the minimum bounding box into the GIoU to minimize the standardized distance between the two BBox center points in order to accelerate the convergence process of loss. The DIoU [29] is more consistent with the object box regression mechanism than the GIoU, taking into account the distance, overlap rate, and scale between the object and anchor so that the object box regression becomes more stable. There will be no divergence in the training process for the IoU and GIoU. Although the DIoU can directly minimize the distance between the center points of the prediction box and the actual box to accelerate convergence, another important factor in the bounding box regression, the aspect ratio, has not been considered yet.

3. Methodology

3.1. System Architecture

The network comprises several parts, which are indicated in a blue font, that work in tandem to achieve continuous object tracking, as shown in Figure 1.

The BcMODT uses a deep neural network composed of several subnetworks, including a backbone network, RPN, RCNN [30], and a PointRCNN [31]. This pipeline, in which each stage builds upon the output of the previous one, enables the system to perform highly efficient object tracking. The backbone network extracts features from the input images and input point cloud, the RPN generates object proposals, the RCNN classifies and refines these proposals, and finally, the PointRCNN performs 3D object detection and instance segmentation.

The detection network uses the RoI and proposal features to generate detection results. The correlation network uses the RoI and BcF to generate Re-ID affinities and start-end probabilities. The proposed 3D-MiIoU and BcF are shown in Figure 1 with green boxes. The data association module refines the affinities between similar objects. The affinity computation module is enabled by the mixed-integer programming [7] approach, which associates detections with objects based on their similarities. Finally, the track management module ensures continuous tracking despite potential object occlusions and reappearances.

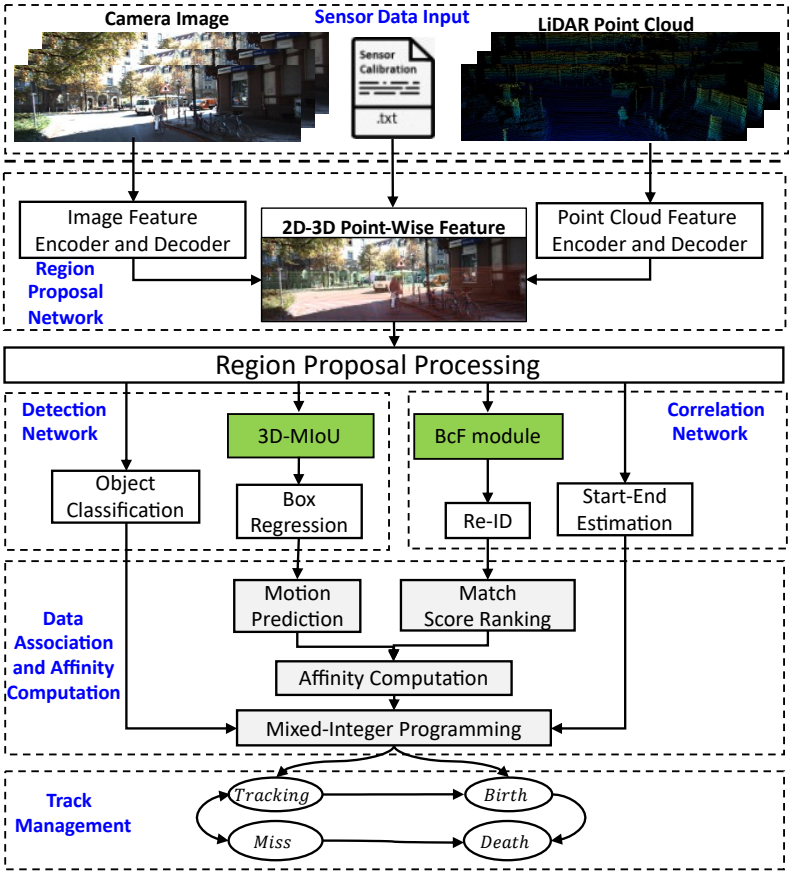

Figure 1. The system architecture of the proposed camera-LiDAR-based joint multi-object detection and tracking system.

3.2. Boost Correlation Feature

To generate 3D bounding boxes and more accurate association scores from the camera and LiDAR measurement data, the shared feature given by the RPN requires additional processing. Without changing the 2D or 3D encoding modules, the RPN features are filtered by the threshold, and the object features under the same ID are homogenized, as shown in Figure 2.

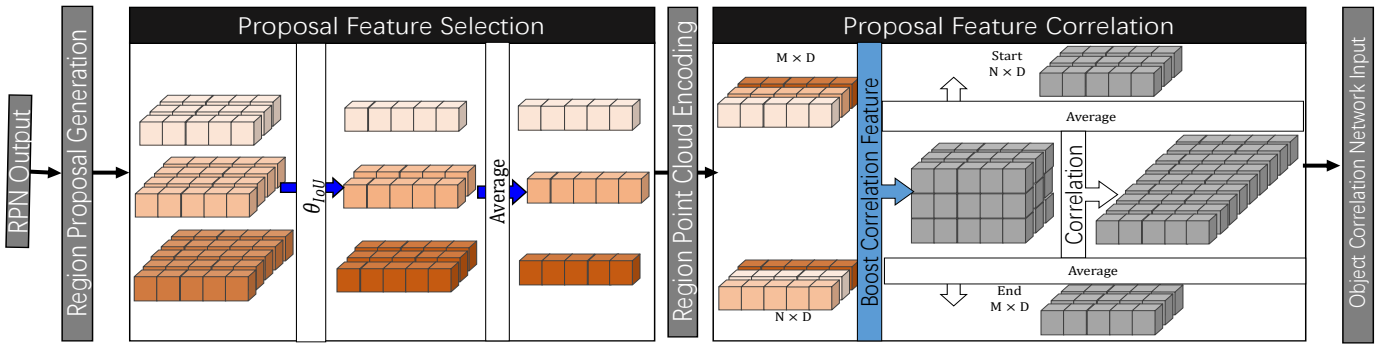

Figure 2. Region proposal processing for training the object correlation network. The input proposal features with the same ID label are shown with the same color.

We use a high-threshold θ_{IoU} to filter out proposal regions from the RPN, which reduces the input of invalid areas and helps ensure network convergence. In addition, to improve the feature obscurity caused by information loss, we improve the robustness of the proposal features by calculating the average value of proposal features with the same ID. Generally, the first operation eliminates unnecessary inputs to ensure the stability of the

training process. The second operation enhances the proposed features by utilizing shared knowledge and supplementing missing information.

The features selected from the proposal feature selection process are passed through the region point cloud-encoding module, where they are transformed and encoded. The encoded features are then used in the BcF module as a pair-wise correlation operation to represent the dependency between objects in adjacent frames.

The mmMOT method [7] proposed element-wise multiplication, subtraction, and absolute subtraction to calculate the candidate feature correlation [12]. To infer the adjacency, the correlation of each detection pair is needed. The correlation operation is batch-agnostic, and thus it can handle cross-modality, and the operation is applied channel by channel to take advantage of the neural network. In JMODT, ineffective candidate features are filtered based on the traditional IoU threshold [17], and absolute subtraction is used to calculate the candidate feature correlation to indicate the correlation between adjacent frames. However, none of these methods cover the directionality of the feature. Therefore, the correlation feature needs to be considered in a more diversified manner. Cosine similarity is a measure of the similarity between two vectors [32]. It is dimensionally independent and insensitive to the sizes of features. Therefore, it can be extended to feature computing at high latitudes. Moreover, it is more concerned with distinguishing the difference from the feature direction, and it is not sensitive to the absolute value. Therefore, we combine the features of $frame_{t-1}$ and $frame_t$ and their cosine similarity to represent the object dependency between adjacent frames, named the boost correlation feature. For M candidate features in the given t frame and N candidate features in the given $t-1$ frame, the size of the feature correlation matrix is $M \times N$. In order to obtain the relationship between global objects, the characteristic matrix is averaged from the rows and columns. Since the start-end estimation is symmetrical, the generated N start features and M end features are transferred to the start-end network together:

$$\begin{aligned} BcF_{d,k} &= \|F_d - F_k\| \frac{F_d \cdot F_k}{\|F_d\| \|F_k\|} \\ &= \|F_d - F_k\| \frac{\sum_{i=1}^{n} F_{d_i} \times F_{k_i}}{\sqrt{\sum_{i=1}^{n} (F_{d_i})^2} \times \sqrt{\sum_{i=1}^{n} (F_{k_i})^2}} \end{aligned} \tag{1}$$

The boost correlation feature is defined in Equation (1), where F_d and F_k denote the feature information of the detected bounding box B_d and tracked bounding box B_k, respectively. $F_d \cdot F_k$ is the inner product of features F_d and F_k. We combine the features of $frame_{t-1}$ and $frame_t$ and their cosine similarity to represent the object dependency between adjacent frames, named the BcF. The schematic diagram of the boost correlation feature module is shown in Figure 3. Specifically, $Feature_{t-1}$ and $Feature_t$ are the features of $frame_{t-1}$ and $frame_t$, respectively, and t represents the time. $||Feature_{t-1} - Feature_t||$ represents the absolute subtraction of $Feature_{t-1}$ and $Feature_t$, $Feature_{t-1} \cdot Feature_{t-1}$ represents the dot product of $||Feature_{t-1}||$ and $||Feature_t||$, and $||Feature_{t-1}||$ and $||Feature_t||$ represent the magnitudes of $Feature_{t-1}$ and $Feature_t$.

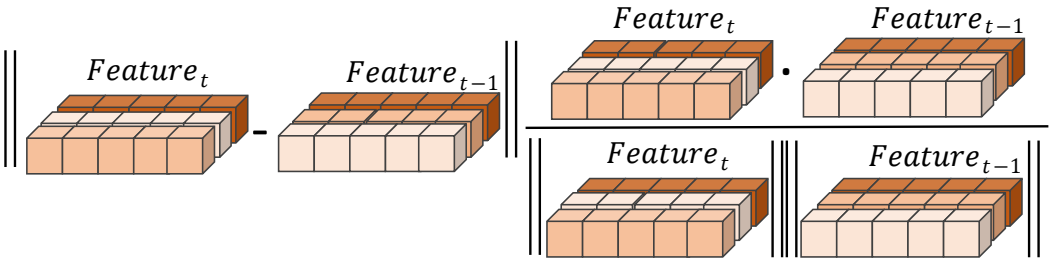

Figure 3. Schematic diagram of boost correlation feature module.

3.3. 3D-IoU

In this section, first, we introduce the 3D-GIoU and 3D-CIoU, which were proposed based on the 2D-IoU, 2D-GIoU, 2D-CIoU, 3D-IoU, 3D-GIoU, and 3D-DIoU in object detection in this paper [28,29,33,34]. Then, we introduce the 3D-MiIoU, which consists of several parts, and the 3D-MiIoU will be described in this subsection.

All 3D-IoUs require the overlapping volume $overlap_{3D_{d,k}}$ and union volume $Union_{d,k}$, which comprise the calculation methods of $overlap_{3D_{d,k}}$. The volume of union $Union_{d,k}$ is provided in advance. In Equation (2), $overlap_{bev_{d,k}}$ is the area of overlap in the top view between B_d and B_k, $overlap_{h_{d,k}}$ is their overlapping height values, and $overlap_{3D_{d,k}}$ is their overlapping volume:

$$overlap_{3D_{d,k}} = overlap_{bev_{d,k}} * overlap_{h_{d,k}} \tag{2}$$

$$Union_{d,k} = vol_d + vol_k - overlap_{3D_{d,k}} \tag{3}$$

In Equation (3), $Union_{d,k}$ is the union volume of the detected bounding box B_d and tracked bounding box B_k, while vol_d and vol_k are the volumes of the detected bounding box B_d and tracked bouding box B_k, respectively. With the defination of $overlap_{3D}$ and $Union_{d,k}$, IoU_{3D} is as defined in Equation (4):

$$IoU_{3D} = \frac{B_d \cap B_k}{B_d \cup B_k} = \frac{overlap_{3D_{d,k}}}{Union_{d,k}} \quad \in [0,1] \tag{4}$$

3.3.1. 3D-GIoU

Bounding boxes overlap differently in 3D and top view cases. For Figure 4a,b, magenta and green represent the tracked bounding box B_k and detected bounding box B_d, respectively. Gray-green represents their intersection. In addition, the blue bounding box lines represent the smallest enclosing box, and the blue dotted line represents its diagonal. The blue bounding box line represents the largest enclosing box, and the blue dotted line represents its diagonal, where $center_{dis}$ is the center point distance of B_d and B_k. Table 2 shows the details of the parameters in Figures 4 and 5.

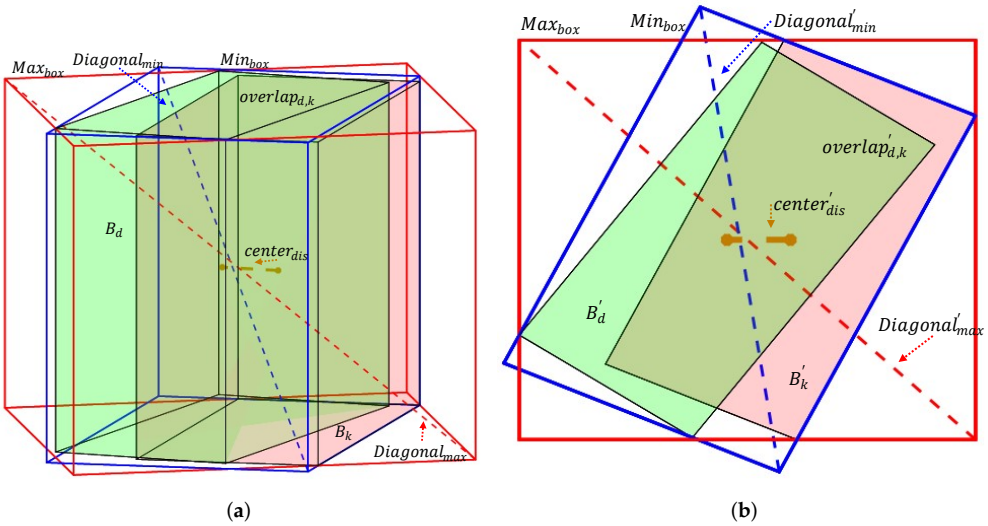

Figure 4. Schematic diagram of the 3D view and top view of 3D-MiIoU. (**a**) 3D-MiIoU. (**b**) Top view of 3D-MiIoU.

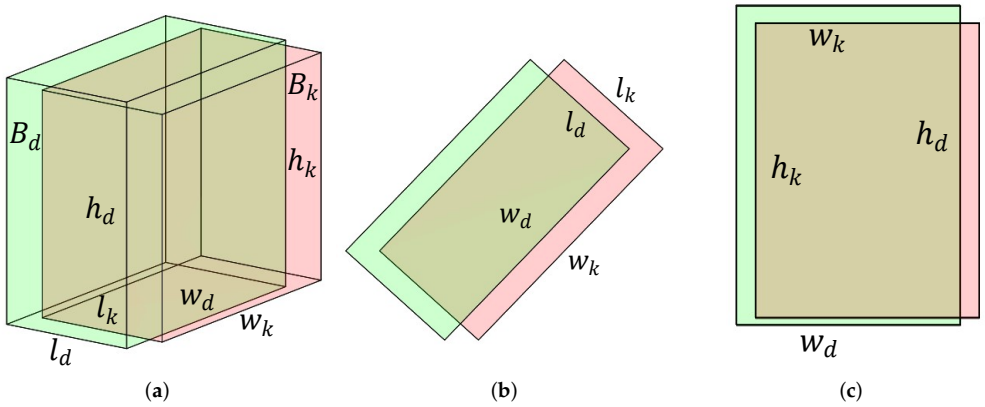

Figure 5. Views of detection boxes and tracking boxes: 3D, top, and right views. (**a**) 3D view. (**b**) Top view. (**c**) Right view.

Table 2. Description of the parameters in Figures 4 and 5.

Parameter	Description
B_d	Detected bounding box in 3D view.
B_k	Tracked bounding box in 3D view.
B'_d	Detected bounding box in top view.
B'_k	Tracked bounding box in top view.
$center_{dis}$	The center distance of B_d and B_k in 3D view.
$center'_{dis}$	The center distance of B_d and B_k in top view.
$Diagonal_{min}$	Diagonal distance of Min_{box}.
$Diagonal_{max}$	Diagonal distance of Max_{box}.
$Diagonal'_{min}$	$Diagonal_{min}$ in top view.
$Diagonal'_{max}$	$Diagonal_{max}$ in top view.
Min_{box}	Minimum bounding boxes of B_d and B_k.
Max_{box}	Maximum bounding boxes of B_d and B_k.
$overlap_{d,k}$	The overlapping volume of B_d and B_k.
$overlap'_{d,k}$	Overlapping area of B'_d and B'_k.

Here, 3D-mGIoU denotes the 3D-GIoU with Box_{min}, and 3D-MGIoU denotes the 3D-GIoU with Box_{max}. When Max_{box} and Min_{box} overlap, $GIoU_{m3D}$ is equal to $GIoU_{M3D}$. Both the 3D-mGIoU and 3D-MGIoU are defined in Equation (5):

$$\begin{aligned} GIoU_{m3D} &= IoU_{3D} - \left(\frac{vol_{min} - Union_{d,k}}{vol_{min}}\right) \\ GIoU_{M3D} &= IoU_{3D} - \left(\frac{vol_{max} - Union_{d,k}}{vol_{max}}\right) \end{aligned} \tag{5}$$

when Max_{box} and Max_{min} overlap and $GIoU_{m3D}$ is equal to $GIoU_{M3D}$.

3.3.2. 3D-CIoU

As for the 2D-CIoU, when the center points of the two boxes coincide, the values of $center_{dis}$ and $Diagonal_{min}$ do not change. Therefore, it is necessary to introduce the

length-to-width ratio and length-to-height ratio between B_d and B_k. Equation (6) defines the 3D-CIoU:

$$\begin{aligned} CIoU_{m3D} &= IoU_{3D} - \left(\frac{center_{dis}}{Diagonal_{min}}\right)^2 - \alpha v \\ CIoU_{M3D} &= IoU_{3D} - \left(\frac{center_{dis}}{Diagonal_{max}}\right)^2 - \alpha v \end{aligned} \tag{6}$$

In Equation (6), the 3D-CIoU is different from the 3D-DIoU. The authors of [17] used $\frac{center_{dis}}{Diagonal_{min}}$. Here, we use $\left(\frac{center_{dis}}{Diagonal_{min}}\right)^2$, and the 3D-CIoU has two additional parameters compared with the 3D-DIoU, which are α and v. Here, α is a parameter used to balance the scale, and it is defined in Equation (7), while v is used to measure the proportion's consistency between the detected bounding box B_d and tracked bounding box B_k, and it is defined in Equation (8):

$$\alpha = \frac{v}{(1 - IoU_{3D}) + v} \tag{7}$$

$$v = \frac{4}{\pi^2}\left(\left(\arctan\frac{l_d}{w_d} - \arctan\frac{l_k}{w_k}\right) + \left(\arctan\frac{l_d}{h_d} - \arctan\frac{l_k}{h_k}\right)\right)^2 \tag{8}$$

where l_d, w_d, and h_d as well as l_k, w_k, and h_k in Figure 5 represent the length, width, and height of B_d and B_k, respectively. In Equation (8), l_d/w_d, and l_k/w_k are the ratios of the length and width of B_d and B_k, respectively, while h_d/w_d and h_k/w_k are the ratios of the height and width of B_d and B_k, respectively. $(arctan(l_d/w_d) - (arctan(l_k/w_k))$ and $(arctan(h_d/w_d) - (arctan(h_k/w_k))$ calculate the difference in the length and width ratios and the difference in the height and width ratios of B_d and B_k, respectively, and v calculates the difference between B_d and B_k by the difference of the inverse tangent value of the aspect ratio of B_d and B_k, which can make full use of the geometric characteristics of B_d and B_k, rendering the affinity more accurate.

3.3.3. 3D-MiIoU

Thus, the 3D-MiIoU uses a combination of the 3D-GIoU and 3D-CIoU. Moreover, it makes calculations with the minimal bounding box Min_{box} and maximum bounding box Max_{box}. Because the IoU based on the minimum external rectangular box and the maximum external rectangular box is calculated several times for the intersection part, we use the average of the 3D-GIoU and 3D-CIoU as the 3D-MiIoU, and the formula for calculating the 3D-MiIoU is defined as follows:

$$MiIoU_{3D} = Average(GIoU_{m3D} + GIoU_{M3D} + CIoU_{m3D} + CIoU_{M3D}) \tag{9}$$

Here, the 3D-MiIoU combines the advantages of the 3D-GIoU and 3D-CIoU, and there are three ways to overlap B_d and B_k. For the first method, when B_d and B_k have no overlap at all, IoU_{3D} is equal to zero, and $MiIoU_{3D}$ is also equal to zero. For the second method, when B_d and B_k completely overlap, IoU_{3D} is equal to one. At this moment, $center_{dis} = 0$, $v = 0$, α does not exist, $vol_{min} = vol_{max} = Union_{d,k}$, and $MiIoU_{3D} = (GIoU_{m3D} + GIoU_{M3D}) = 2 * IoU_{3D}$. Therefore, $MiIoU_{3D}$ needs to be averaged. For the final method, when B_d and B_k do not completely overlap, $MiIoU_{3D}$ contains four IoU_{3D}s. If we want to use $MiIoU_{3D}$ as the IoU, we need to average $MiIoU_{3D}$ to reduce the influence of multiple IoU_{3D}s on $MiIoU_{3D}$. Considering three different overlapping situations, we use the average value of $(GIoU_{m3D} + GIoU_{M3D} + CIoU_{m3D} + CIoU_{M3D})$ as the 3D-MiIoU. Based on the experimental results of the KITTI dataset [9], the 3D-MiIoU improved its performance more than the others. The pseudo-code of the 3D-MiIoU is provided in Algorithm 1.

Algorithm 1 Three-dimensional mixed intersection over union.

Require: $B_d = (x_d, y_d, z_d, h_d, w_d, l_d, \theta_d)$, $B_k = (x_k, y_k, z_k, h_k, w_k, l_k, \theta_k)$;
Ensure: 3D-MiIoU

function 3D-MIIOU(B_d, B_k)
  Calculate B'_d and B'_k for B_d and B_k on Top View;
    $B'_d = (x'_d, z'_d, w'_d, l'_d, \theta'_d)$,
    $B'_k = (x'_k, z'_k, w'_k, l'_k, \theta'_k)$;
  Calculate the smallest 3D enclosing box Min_{Box} and the largest 3D enclosing box Max_{box};
    $Min_{Box} = (x_{min}, y_{min}, z_{min}, h_{min}, w_{min}, l_{min}, \theta_{min})$,
    $Max_{Box} = (x_{max}, y_{max}, z_{max}, h_{max}, w_{max}, l_{max}, \theta_{max})$;
  Calculate $Digonal_{min}$ and $Digonal_{max}$;
    $Digonal_{min} = \sqrt{{h_{min}}^2 + {w_{min}}^2 + {l_{min}}^2}$,
    $Digonal_{max} = \sqrt{{h_{max}}^2 + {w_{max}}^2 + {l_{max}}^2}$;
  Calculate vol_d, vol_k, vol_{min}, vol_{max}, $overlap_h$, $overlap_{bev}$, $overlap_{3D}$;
    $vol_d \leftarrow h_d * w_d * l_d$,
    $vol_k \leftarrow h_k * w_k * l_k$,
    $vol_{min} \leftarrow h_{min} * w_{min} * l_{min}$,
    $vol_{max} \leftarrow h_{max} * w_{max} * l_{max}$,
    $overlap_h \leftarrow min(\|y_d + \frac{h_d}{2}\|, \|y_k + \frac{h_k}{2}\|) - max(\|y_k - \frac{h_d}{2}\|, \|y_k - \frac{h_k}{2}\|)$,
    $overlap_{bev} \leftarrow B'_d \cap B'_k$,
    $overlap_{3D} \leftarrow$ calculate with Equation (2);
  Calculate 3D-MiIoU:
  if $overlap_{bev}$ or $overlap_h == 0$:
    $IoU_{3D} = 0$
  else:
    $IoU_{3D} \leftarrow$ calculate with Equation (4),
    $mGIoU_{3D} \leftarrow$ calculate with Equation (5),
    $MGIoU_{3D} \leftarrow$ calculate with Equation (5),
    $mCIoU_{3D} \leftarrow$ calculate with Equation (6),
    $MCIoU_{3D} \leftarrow$ calculate with Equation (6);
  3D-MiIoU$\leftarrow$ calculate with Equation (9).

3.4. Affinity Computation

This section introduces the affinity calculation module. Compared with calculating the appearance affinity based on the distance of the camera-LiDAR fusion features, this section adds the BcF to the appearance affinity, integrates the factors of fusion feature directivity between adjacent frames, improves the measurement of appearance affinity, and calculates the appearance affinity more accurately. Geometric affinity combines the characteristics of the proposed 3D-GIoU and 3D-CIoU by adding the aspect ratios of B_d and B_k as penalty items, and the geometric features of the box can be used efficiently. This combination renders the affinity calculation more accurate and can provide more accurate information for data association and tracking. Algorithm 2 provides the pseudo-code of the affinity calculation.

B_d and B_k are the detected bounding box and tracked bounding box, respectively, A^{app} denotes the appearance affinity, and $A^{3D-MiIoU}$ denotes the geometrical affinity, while X^{aff} is the weighted sum of the appearance affinity A^{app} and geometrical affinity $A^{3D-MiIoU}$.

Algorithm 2 Affinity metric with BcF and 3D-MiIoU.

Require: Detection measurements D, tracks K and their proposal features $\boldsymbol{F} = \{F_i, i \in \boldsymbol{D} \cap \boldsymbol{K}\}$.

Ensure: refined affinities $X^{aff} = \left\{x_{d,k}^{aff}, d \in D, k \in K\right\}$

function LS($Features_{pred}, Features_{next}$)
  for each $k \in K$ **do**
    $B_k \leftarrow$ 3D box prediction for track k using **Kalman Filter**.
    for each $d \in D$ **do**
      $BcF_{d,k} \leftarrow$ calculate with Equation (1);
      $a_{d,k}^{app} \leftarrow$ Appearance Re-ID for boost feature $BcF_{d,k}$;
      $B_d \leftarrow$ 3D box prediction for detection d;
      $a_{d,k}^{\text{3D-MiIoU}} \leftarrow$ calculate with Algorithm 1.
    end for
  end for
$A^{app} \leftarrow \left\{a_{d,k}^{app}, d \in D, k \in K\right\}$;
$A^{\text{3D-MiIoU}} \leftarrow \left\{a_{d,k}^{3d-MiIoU}, d \in D, k \in K\right\}$;
$\mathbf{P} \leftarrow$ Softmax A^{app} along columns;
$\mathbf{Q} \leftarrow$ Softmax A^{app} along rows;
$A^{app} \leftarrow \frac{1}{2}(P + Q)$;
$X^{aff} \leftarrow \alpha A^{app} + \beta A^{\text{3D-MiIoU}}$.

3.5. Time Complexity

BcMODT uses cosine similarity to calculate the feature distance and direction similarity between adjacent frames rather than using absolute subtraction, as observed in other methods such as JMODT and mmMOT. The time complexity of the absolute subtraction of F_d and F_k depends on their size, while the time complexity of computing the cosine similarity between two vectors, F_{t-1} and F_t, is $O(n)$, where n is the number of elements in the vectors.

Additionally, our method combines the features of $frame_{t-1}$ and $frame_t$ and their cosine similarity to represent the object dependency between adjacent frames. The time complexity of performing the element-wise multiplication of vector $F_{t-1} - F_t$ by a scalar value, which is the cosine similarity of $F_{t-1} - F_t$, is also $O(n)$. This is because computing the cosine similarity requires calculating the dot product of the two vectors and the magnitudes of the two vectors and then dividing the dot product by the product of the magnitudes. In summary, the time complexity of the operation of absolute subtraction and the BcF is $O(n)$.

4. Experiments

This section provides the experimental results for BcMODT, including the experiment's settings, baseline and evaluation metrics, quantitative results, ablation experiments, qualitative results, and limitations.

4.1. Experimental Settings

This work ran on a computer with an Intel(R) Core (TM) i7-10700K CPU, 32 GB of RAM, and RTX 3090 $\times$ 2 and programs with the languages of Python and Pytorch [35]. We used the pretrained detection model of EPNet [36]. The correlation network was trained for 60 epochs with a batch size of 4. We used the AdamW [37] optimizer with a cosine annealing learning rate [38] of 2×10^{-4}. The parameters of all compared methods were set according to their best performances. For data association, we used improved MIP [17] as the data association method in this study, and the parameters were set as in JMODT.

4.2. Baseline and Evaluation Metrics

We evaluated our proposed 3D-MOT method on the KITTI [9] tracking dataset. The new method consists of 21 training sequences and 29 test sequences of forward-looking

camera image information and LiDAR point cloud information. The training sequence is divided into approximately equal training sets and verification sets.

In addition, each ground truth in the frame contain a 3D bounding box with a unique ID. Only objects with a 2D-IoU [23] greater than 0.5 can be accepted as TP. According to KITTI standards [9], we used CLEARMOT, MT/ML/FP/FN, ID switch (IDSW), and fragmentation (Frag) to evaluate the MOT performance [39]. The details of the official evaluation metrics are shown in Table 3.

Table 3. Evaluation measures.

Measure	Better	Perfect	Description
MOTA [39]	Higher	100%	Multi-object tracking accuracy.
MOTP [39]	Higher	100%	Multi-object tracking precision.
HOTA [40]	Higher	100%	Higher-order tracking accuracy.
MT	Higher	100%	Mostly tracked targets.
ML	Lower	0%	Partly tracked targets.
FP	Lower	0	The total number of false positives.
FN	Lower	0	The total number of false negatives (missed targets).
IDSW	Lower	0	Number of identity switches.
FRAG	Lower	0	The total number of times a trajectory is fragmented.
Time	Lower	-	The total execution time.
FPS	Higher	-	Frames per second.

4.3. Quantitative Results

Compared with other published methods, such as AB3DMOT, mmMOT, JRMOT, and JMODT, in the vehicle-tracking benchmark tests using the KITTI dataset [9], our method improved the accuracy to a certain extent and outperformed the other methods with respect to some indicators, as shown in Tables 4 and 5. Tables 4 and 5 provide two evaluation standards. Table 4 is the evaluation data based on MOTA [39], and Table 5 is based on HOTA [40].

Table 4. KITTI car tracking results based on MOTA. "×" means not, while "✓" means yes.

Method	**AB3DMOT [11]**	**mmMOT [7]**	**JRMOT [6]**	**JMODT [17]**	**BcMOT Ours**
Benchmark	Car	Car	Car	Car	Car
JDT	×	×	×	✓	✓
AT	✓	✓	✓	×	×
MOTA ↑	83.92%	84.77%	85.70%	86.27%	**86.53**%
MOTP ↑	85.30%	85.21%	**85.48**%	85.41%	85.37%
MODA ↑	83.95%	85.60%	85.98%	86.40%	**86.66**%
MODP ↑	88.21%	88.28%	**88.42**%	88.32%	88.29%
TP ↑	33,864	33,695	34,556	35,857	**35,972**
FP ↓	978	**711**	772	772	1248
FN ↓	4542	4243	4049	3433	**3341**
MT ↑	66.77%	73.23%	71.85%	77.38%	**78.31**%
ML ↓	9.08%	2.77%	4.00%	2.92%	**2.62**%
IDSW ↓	**10**	284	98	45	45
Frag ↓	**199**	753	372	585	626
Runtime	0.005 s	0.002 s	0.007 s	0.001 s	0.001 s

The data are accessed on 21 November 2022 from https://www.cvlibs.net/datasets/kitti/old_eval_tracking.php.

In the evaluation results based on MOTA on the KITTI benchmark, compared with the baseline and others, we can see that our method progressed in MOTA, MODA, TP, FP, MT, and ML. Moreover, in the evaluation's results based on HOTA on the KITTI benchmark, our method was better than the baseline and the other methods in HOTA, MOTA, TP, FP, MT, and ML.

Table 5. KITTI car tracking results based on HOTA. "×" means not, while "✓" means yes.

Method	AB3DMOT [11]	mmMOT [7]	JRMOT [6]	JMODT [17]	BcMOT Ours
Benchmark	Car	Car	Car	Car	Car
JDT	×	×	×	✓	✓
AT	✓	✓	✓	×	×
HOTA↑	69.99%	62.05%	69.61%	70.73%	**71.00**%
MOTA↑	83.61%	83.23%	85.10%	85.35%	**85.48**%
MOTP↑	85.23%	85.03%	85.28%	**85.37**%	85.31%
TP↑	29,849	30,325	30,108	30,954	**31,039**
FP↓	4543	4067	4284	3438	**3353**
FN↓	979	787	**752**	1249	1260
MT↑	66.92%	72.92%	70.92%	77.39%	**78.15**%
ML↓	9.08%	2.92%	4.62%	2.92%	**2.62**%
IDSW↓	**113**	733	271	350	381
Frag↓	**206**	570	273	693	732
Runtime	0.005 s	0.002 s	0.007 s	0.001 s	0.001 s

The data are accessed on 21 November 2022 from https://www.cvlibs.net/datasets/kitti/eval_tracking.php.

In Tables 4 and 5, our method had some improvement in multi-object tracking accuracy, ML, etc. compared with the baseline and other methods. In the MOTA-based evaluation criteria, our method improved by 0.26% over the baseline in MOTA, the true positive result improved by 115, the most tracked result improved by 0.93%, and the most lost result decreased by 0.3%, as shown in Table 4. In the HOTA-based evaluation criteria, our method improved by 0.27% over the baseline in HOTA, MOTA increased by 0.13%, the true positive result improved by 85, the most tracked result improved by 0.76%, and the most lost result decreased by 0.3% in Table 5. Although most of our evaluation metrics were better than those for the other methods, our methods had certain defects. The total number of times the trajectories fragmented was greater than the numbers in other methods. This may be related to the fact that 3D-MiIoU-based geometric affinity calculations use the aspect ratio. When the box's length, width, and height slightly change, the geometric affinity will fluctuate, resulting in slight fluctuations.

4.4. Ablation Experiments

In this subsection, we alternately removed the 3D-DIoU, 3D-IoU, 3D-GIoU, 3D-CIoU, 3D-MiIoU, and BcF to perform the ablation study, as shown in Table 6. It can be seen that our method could improve tracking performances.

Table 6 shows that multiple IoU ablation experiments with the BcF module were performed. Seven types of IoU combinations are provided in the first column of the table, and each IoU was ablated twice for the BcF. Moreover, two comparisons were performed for each IoU combination in the table for the BcF. Each IoU combination with × indicates the original IoU calculation method, and those with ✓ indicate that the IoU was added to the BcF module. When comparing all IoU methods, the accuracy of all IoU calculation methods significantly improved after adding the BcF module. Compared with the IoU method without adding the BcF module, our proposed 3D-MiIoU method had limited improvement in terms tracking accuracy, with only 0.20% improvement in comparison

with the baseline. Compared with the 3D-MiIoU without the BcF, our method increased by 0.37% in MOTA. Compared with the 3D-DIoU with the BcF, our method increased by 0.30% in MOTA. Compared with the 3D-DIoU, our method increased 0.57% in MOTA. Meanwhile, the FP, FN, and IDSW of all IoU methods decreased to different degrees after adding the BcF module.

Table 6. Evaluation of different metrics for affinity computation. "×" means without BcF, while "✓" means with BcF.

Measure	BcF	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDSW↓	Frag↓	Time↓	FPS↑
3D-DIoU	×	86.09%	87.13%	86.11%	1.39%	545	997	4	144	40.11	92.24
	✓	86.36%	87.16%	85.65%	1.39%	531	984	1	137	38.901	94.83
3D-GIoU	×	86.15%	87.15%	86.11%	1.39%	531	1004	4	144	38.84	95.27
	✓	86.39%	87.17%	85.19%	1.39%	516	993	5	139	39.629	95.11
3D-CIoU	×	85.82%	87.25%	84.72%	1.85%	500	1072	4	144	**37.63**	**98.32**
	✓	86.11%	87.26%	85.19%	1.85%	486	1055	3	137	38.69	95.36
3D-DIoU 3D-GIoU	×	86.12 %	87.10%	86.11%	1.39%	564	977	2	141	38.87	95.20
	✓	86.42%	87.12%	85.65%	1.39%	546	962	1	137	39.35	93.75
3D-DIoU 3D-CIoU	×	86.06 %	87.13%	86.11%	1.39%	546	999	4	144	39.14	94.53
	✓	86.39%	87.16%	85.65%	1.39%	526	987	**0**	137	38.57	95.71
3D-DIoU 3D-GIoU 3D-CIoU	×	86.10%	87.10%	86.11%	1.39%	565	978	2	141	39.59	93.45
	✓	86.46%	87.12%	85.65%	1.39%	541	964	**0**	137	39.26	93.96
3D-MiIoU (Ours)	×	**86.29**%	87.19%	85.65%	1.39%	527	993	4	**127**	38.50	96.69
	✓	**86.66**%	87.18%	**86.57**%	1.39%	490	990	3	142	38.34	97.29

Comparing the total execution time and FPS with BcF and without BcF of different metrics of the affinity computation in the ablation experiment, the results of the total execution time and FPS in Table 6 and Figure 6 show that our proposed method is almost the same as 3D-DIoU, because the time complexity of calculating the distance between adjacent frame features by using absolute subtraction and the time complexity of our proposed method are both $O(n)$. Therefore, our proposed method will not increase the total execution time of the algorithm. Both the total execution time and FPS in Figure 6 can explain that these additional calculations had no effect on the time complexity.

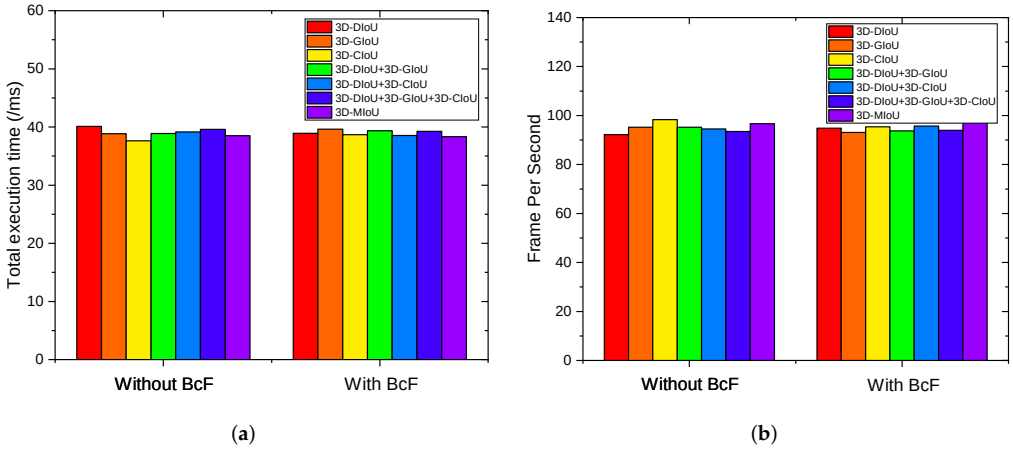

Figure 6. Total execution time and FPS of different metrics. (**a**) Total execution time. (**b**) Frames per second.

4.5. Qualitative Results

In multi-object detection and tracking, detection and tracking are very challenging because of occlusion and other problems. Whether in 2D images or 3D point clouds, the object may be partially or completely occluded for a while. We compared the 2D

visualization results for our method with the baseline on the KITTI [9] dataset. We selected 40 frames from 70 to 110 in sequence 0002 and displayed every five frames in Figure 7. Figure 7 shows the test results of the ground truth, AB3DMOT, JMODT, and our method.

Compared with AB3DMOT, the 80th, 85th, 90th, 100th, and 105th frames show that our method had good performance compared with AB3DMOT in terms of missed detections. The 70th and 90th frames show that our method was better than AB3DMOT in terms of false detection. Compared with the baseline (JMODT), the frames from 70 to 110 indicate that our method outperformed the baseline in terms of missed detection. The yellow circle of the lines from frame 70 to 105 show that our method was superior to the baseline (JMODT) in IDSW. By conducting a comprehensive comparison, our method was observed to be closer to the ground truth.

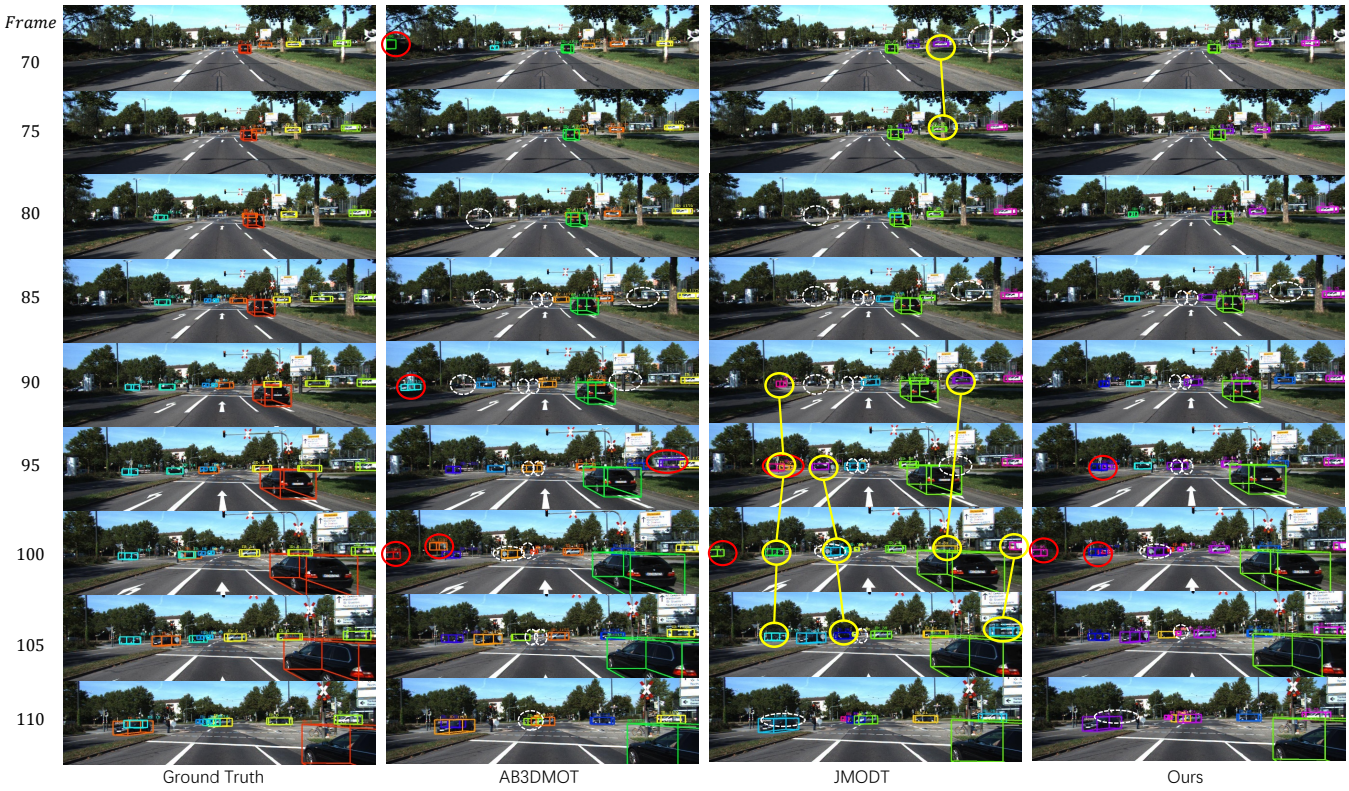

Figure 7. Visualization of tracking comparisons between the ground truth, baseline, and our improved work on trajectory 2D images of the KITTI [9] dataset. The squares indicate the detected objects. The red circle indicates false detection, the white dotted circle indicates missing detection, and the yellow circle indicates IDSW. All datasets and benchmarks on KITTI [9] are copyright by KITTI and published under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.

4.6. Limitations

However, our proposed BcMODT also has certain limitations. One limitation of the proposed method is that it may not perform as well when applied to other types of 3D object-tracking scenarios beyond autonomous driving. For example, the method may not be as effective when tracking objects in environments with complex backgrounds or in scenarios with a more significant number of objects. Another limitation is that the proposed method relies on the availability of both camera and LiDAR data, which may not always be feasible in specific applications. A dataset cannot prove the superiority of the proposed method, and experimental data of more representative datasets are required.

5. Conclusions

In this paper, an online 3D multi-object detection and tracking method was proposed. In summary, the 3D-MiIoU can improve geometrical affinities. The boost correlation feature

enhancement module is designed to enhance the correlation between similar objects and can also provide the network with an association score of the real-time detection camera and LiDAR measurement data. Extensive experiments were carried out on the public KITTI benchmark. Our method was superior to other methods in terms of tracking accuracy and speed. Without using additional training datasets, our method obtained the MOTA (86.53%) in the MOTA-based evaluation criteria and HOTA (71.00%) in the HOTA-based evaluation criteria. Compared with the baseline, BcMODT improved the MOTA in the MOTA-based evaluation criteria by (0.26%) and improved HOTA in the HOTA-based evaluation criteria by (0.27%). Due to the fusion of camera and LiDAR data, as well as the fusion of object detection and tracking, our method is very suitable for autopilot applications that require high tracking robustness and real-time performance.

In the future, we will focus on adapting the proposed method in order to better handle these types of scenarios and further improve the tracking accuracies. Additionally, it would be interesting to explore the use of additional modalities beyond camera and LiDAR data to observe if this leads to further improvements in terms of tracking performance. It would also be valuable to investigate methods for improving the real-time processing of large amounts of data in 3D MOT to enable more efficient tracking in complex scenarios.

Author Contributions: Methodology, K.Z.; software, K.Z. and J.J.; validation, K.Z. and Y.W.; writing—original draft preparation, K.Z.; writing—review and editing, Y.W., J.J. and F.M.; visualization, K.Z. and J.J.; supervision, Y.L. and F.M.; funding acquisition, Y.L. and F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported in part by the National Natural Science Foundation of China (62172186, 62002133, 61872158, and 61806083), in part by the Science and Technology Development Plan Project of Jilin Province (20190701019GH, 20190701002GH, 20210101183JC, 20210201072GX, and 20220101101JC), and in part by the Young Science and Technology Talent Lift Project of Jilin Province (QT202013).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020 ; pp. 6499–6508.
2. Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; Yuan, J. Track to detect and segment: An online multi-object tracker. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12352–12361.
3. Leibe, B.; Schindler, K.; Van Gool, L. Coupled detection and trajectory estimation for multi-object tracking. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
4. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1341–1360. [CrossRef]
5. Kim, A.; Ošep, A.; Leal-Taixé, L. Eagermot: 3d multi-object tracking via sensor fusion. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11315–11321.
6. Shenoi, A.; Patel, M.; Gwak, J.; Goebel, P.; Sadeghian, A.; Rezatofighi, H.; Martin-Martin, R.; Savarese, S. Jrmot: A real-time 3d multi-object tracker and a new large-scale dataset. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 10335–10342.
7. Zhang, W.; Zhou, H.; Sun, S.; Wang, Z.; Shi, J.; Loy, C.C. Robust multi-modality multi-object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 2365–2374.
8. Gonzalez, N.F.; Ospina, A.; Calvez, P. Smat: Smart multiple affinity metrics for multiple object tracking. In Proceedings of the International Conference on Image Analysis and Recognition, Povoa de Varzim, Portugal, 24–26 June 2020; pp. 48–62.
9. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
10. Li, Y.; Huang, C.; Nevatia, R. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2953–2960.

11. Weng, X.; Wang, J.; Held, D.; Kitani, K. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *arXiv* **2020**, arXiv:2008.08063.
12. An, J.; Zhang, D.; Xu, K.; Wang, D. An OpenCL-Based FPGA Accelerator for Faster R-CNN. *Entropy* **2022**, *24*, 1346. [CrossRef]
13. Lu, Z.; Rathod, V.; Votel, R.; Huang, J. Retinatrack: Online single stage joint detection and tracking. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14668–14678.
14. Zhou, X.; Koltun, V.; Krähenbühl, P. Tracking objects as points. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 474–490.
15. Peng, J.; Wang, C.; Wan, F.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Fu, Y. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 145–161.
16. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards real-time multi-object tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 107–122.
17. Huang, K.; Hao, Q. Joint Multi-Object Detection and Tracking with Camera-LiDAR Fusion for Autonomous Driving. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 6983–6989.
18. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3d object detection from rgb-d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, UT, USA, 18–23 June 2018; pp. 918–927.
19. Mykheievskyi, D.; Borysenko, D.; Porokhonskyy, V. Learning local feature descriptors for multiple object tracking. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–4 December 2020.
20. Wu, Y.; Liu, Z.; Chen, Y.; Zheng, X.; Zhang, Q.; Yang, M.; Tang, G. FCNet: Stereo 3D Object Detection with Feature Correlation Networks. *Entropy* **2022**, *24*, 1121. [CrossRef] [PubMed]
21. Zhao, M.; Jha, A.; Liu, Q.; Millis, B.A.; Mahadevan-Jansen, A.; Lu, L.; Landman, B.A.; Tyska, M.J.; Huo, Y. Faster Mean-shift: GPU-accelerated clustering for cosine embedding-based cell segmentation and tracking. *Med. Image Anal.* **2021**, *71*, 102048. [CrossRef] [PubMed]
22. You, L.; Jiang, H.; Hu, J.; Chang, C.H.; Chen, L.; Cui, X.; Zhao, M. GPU-accelerated Faster Mean Shift with euclidean distance metrics. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA, 27 June–1 July 2022; pp. 211–216.
23. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of localization confidence for accurate object detection. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–799.
24. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM international conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
25. Elhoseny, M. Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. *Circuits Syst. Signal Process.* **2020**, *39*, 611–630. [CrossRef]
26. Farag, W. Kalman-filter-based sensor fusion applied to road-objects detection and tracking for autonomous vehicles. *Proc. Inst. Mech. Eng. Part. J. Syst. Control Eng.* **2021**, *235*, 1125–1138. [CrossRef]
27. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS–improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
28. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
29. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *arXiv* **2019**, arXiv:1911.08287.
30. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
31. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
32. Nguyen, H.V.; Bai, L. Cosine similarity metric learning for face verification. In Proceedings of the Asian Conference on Computer Vision, Queenstown, New Zealand, 8–12 November 2010; pp. 709–720.
33. Xu, J.; Ma, Y.; He, S.; Zhu, J. 3D-GIoU: 3D generalized intersection over union for object detection in point cloud. *Sensors* **2019**, *19*, 4093. [CrossRef] [PubMed]
34. Chen, Y.; Li, H.; Gao, R.; Zhao, D. Boost 3-D object detection via point clouds segmentation and fused 3-D GIoU-L1 loss. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 762–773. [CrossRef] [PubMed]
35. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in Pytorch. 2017. Available online: https://openreview.net/forum?id=BJJsrmfCZ (accessed on 21 November 2022).
36. Huang, T.; Liu, Z.; Chen, X.; Bai, X. Epnet: Enhancing point features with image semantics for 3d object detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 35–52.
37. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
38. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.

39. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The clear mot metrics. *Eurasip. J. Image Video Process.* **2008**, *2008*, 246309. [CrossRef]
40. Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; Leibe, B. Hota: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 548–578. [CrossRef] [PubMed]