applied sciences

MDPI

Article

IOF-Tracker: A Two-Stage Multiple Targets Tracking Method Using Spatial-Temporal Fusion Algorithm

Hongbin Liu *, Yongze Zhao, Peng Dong, Xiuyi Guo and Yilin Wang

Shandong Key Laboratory of Smart Buildings and Energy Efficiency, School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan 250101, China; 2023085129@stu.sdjzu.edu.cn (Y.Z.); 2024080105@stu.sdjzu.edu.cn (P.D.); 2024080120@stu.sdjzu.edu.cn (X.G.); 2023085137@stu.sdjzu.edu.cn (Y.W.)
* Correspondence: liuhongbin19@sdjzu.edu.cn; Tel.: +86-187-5411-7558

Abstract: Multi-object tracking aims to track multiple objects across consecutive frames in a video, assigning a unique classifier to each object. However, issues such as occlusions, directional changes, or shape alterations can cause appearance variations, leading to detection and matching problems that in turn result in frequent ID switches. To solve these issues, this paper proposes a two-stage multi-object tracking framework based on a spatial and temporal fusion algorithm. First, the video frames are processed by a detector to identify objects and form rectangular detection areas. Meanwhile, an estimator predicts the target rectangular areas in the next frame. Then, we extract the optical flow of the target pixels within the detection and prediction areas, and then a temporal information model is established by calculating the average of the target pixels' optical flow. Afterward, we present a spatial information model using the R-IoU (Reverse of Intersection over Union) between the detection and prediction areas. This spatial and temporal information is combined with weighted matrix fusion, which achieves the feature matching and association task. Finally, we implement a two-stage association multi-object tracking model using the mentioned fusion algorithm. Experiments on the MOTChallenge dataset using the official detector show that our two-stage multi-object tracking method based on the spatial and temporal fusion algorithm is robust in handling occlusions and ID switch issues. As of the submission of this paper, the proposed method has achieved the top ranking in the MOT17 benchmark when evaluated with the official detector.

Keywords: multi-object tracking; ID switches; fusion algorithm; spatial and temporal information

check for updates

Academic Editor: Pedro Couto

Received: 17 November 2024
Revised: 21 December 2024
Accepted: 24 December 2024
Published: 26 December 2024

Citation: Liu, H.; Zhao, Y.; Dong, P.; Guo, X.; Wang, Y. IOF-Tracker: A Two-Stage Multiple Targets Tracking Method Using Spatial-Temporal Fusion Algorithm. *Appl. Sci.* **2025**, *15*, 107. https://doi.org/10.3390/app15010107

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Multi-Object Tracking (MOT) is an important task in the fields of computer vision and artificial intelligence, aiming to continuously track the trajectories of multiple moving objects in video sequences. This task not only requires accurately detecting targets in each frame but also maintaining the consistency of target identities, handling complex situations such as occlusion between targets, interaction, and environmental changes. However, today's multi-object tracking tasks also face several challenges: (1) Identity switch problem. Due to the interference of complex factors (such as obstacles, pedestrian interactions and fast movement), it is difficult to accurately match targets in the process of tracking. These mismatches easily lead to the same target being recognized as another target, resulting in confusion and tracking errors. (2) Missed detections. This occurs when a target is detected in the current frame but not in subsequent frames. This issue prevents the prediction region from finding a corresponding detection region, leading to tracking failure. (3) False

positives. The system incorrectly detects false targets. In such cases, the system may erroneously associate real targets with false ones, or it may associate multiple real targets with the same false target, resulting in tracking failure. (4) Target association problem. Due to insufficient utilization of various features of the target by the tracker, the system has difficulty maintaining continuous association and matching of the same target during the association phase, leading to identity confusion or tracking loss. These issues make it difficult for most tracking algorithms to achieve high tracking performance.

Many researchers currently address the challenges in multi-object tracking tasks within the framework of detection-based tracking. Larsen employed the BASE [1] minimalist probabilistic visual tracking method, which achieved significant results in enhancing tracking performance. However, in complex scenarios, such as handling highly irregular movements in the DanceTrack dataset, BASE's motion model assumption of continuous and slow-changing motion may lead to a decline in tracking performance. Liu proposed the SparseTrack [2] multi-object tracking method, which addresses occlusion issues through pseudo-depth-based scene decomposition and has achieved certain successes in improving tracking performance. However, in scenes with fast-moving and deformable targets, the pseudo-depth method struggles to accurately capture the relative depth relationships of targets, affecting tracking performance. Zhang introduced the ByteTrack [3] multi-object tracking algorithm framework, which solves tracking issues caused by target occlusion and the discarding of detection boxes with low confidence by using a high-performance detector. However, since it only utilizes one feature, either IoU or ReID, the method's insufficient use of target features negatively impacts tracking performance. Aharon proposed the BoT-SORT [4] multi-object tracking method, which significantly improves tracking performance by enhancing the Kalman filter in ByteTrack, introducing camera motion compensation, and optimizing the IoU-ReID fusion strategy. Although this method leverages appearance information to some extent, changes in pedestrian posture under different viewpoints affect the shape of detection boxes, leading to inaccurate IoU calculations and increased difficulty in extracting target appearance features. This can result in inaccurate matching during the target association process. With the advancement and optimization of object detection technologies, many researchers have opted to integrate high-performance detectors [5] with trackers to obtain high-precision detection results for optimizing tracking performance. However, this approach not only increases the cost of tracking tasks but also provides only marginal improvements. This method still generates a significant number of IDs. In recent years, some researchers have adopted neural network-based feature learning models to enhance the robustness of multi-object tracking under occlusion. However, since these methods rely solely on spatial features such as target appearance [6], the reliability of these features is low in complex environments with mutual occlusion, leading to inaccurate multi-object tracking. Song et al. [7] argue that temporal features (such as motion direction, speed, and acceleration) are also crucial tracking information that can effectively enhance multi-object tracking performance.

To address the aforementioned issues, we propose the IOF-Tracker multi-object tracking method, which is a two-stage model that combines spatial (R-IoU, Reverse of Intersection over Union) and temporal (Optical Flow) fusion algorithms. This method enables the tracker to fully utilize the optical flow motion features of the targets and combine the reverse intersection over union (R-IoU) between the detection and prediction regions for template matching. This reduces the frequent ID switching during tracking and allows the tracker to achieve long-term continuous tracking. The method enhances the accuracy and robustness of the tracker in handling multi-object tracking tasks, leading to more stable and precise multi-object tracking. Both the optical flow method [8,9] and the IoU method [10] are classical techniques. In this paper, we have chosen these two classic features as the base-

lines, and the R-IoU and mean optical flow are designed to make the features more suitable for multi-object tracking. Although some existing multi-object tracking methods [11] in the literature also involve the IoU and optical flow features, their performance still needs to be improved. In contrast, our method leverages the mean optical flow of targets to handle more complex scenes and combines R-IoU features to achieve multi-feature association, resulting in better tracking performance.

During the tracking process, we first use an existing detector to identify target regions in each frame. We then apply a Kalman filter to predict the positions of each target in future frames. The Kalman filter predicts the current position based on the target's motion model and the previous state. It corrects this prediction using the detected position information. By calculating the Kalman gain, which balances the weight between prediction and observation, the target state is updated to more accurately estimate the possible positions of the target in future frames, i.e., the predicted regions. Next, we extract optical flow information from both the detection regions and predicted regions of each target. During extraction, we average the optical flow information of all pixels within the region to obtain an average optical flow vector, reducing the noise interference in the optical flow. Subsequently, we calculate the R-IoU between the detection regions and predicted regions to obtain spatial information, which measures their spatial overlap. The feature fusion is achieved by weighting the addition of these two pieces of information in matrix dimensions. Finally, we integrate this fusion model into each stage of the two-stage cascaded matching. This allows the tracker to fully utilize both the temporal and spatial features of targets during the association and matching phase, thereby our approach can enhance the tracking performance of the tracker. The main contributions of this paper are as follows:

- To obtain the overall motion trend of the target and to suppress noise interference, we calculate the average optical flow vectors of each pixel within the detection and prediction regions. The average optical flow can improve the robustness of motion estimation.
- To address the issue that most trackers underutilize target motion features during multi-object tracking, we introduce the temporal feature represented by optical flow and the spatial feature indicated by R-IoU (Reverse of Intersection over Union). By fusing these temporal and spatial features, we develop a model that enhances the tracking performance of the tracker.
- We integrate the R-IoU and optical flow feature fusion model into a two-stage association tracking framework. Experimental results show that our IOF-Tracker multi-object tracking method significantly improves the tracking performance. On the MOT17 dataset, our method's HOTA score reaches 64.9. At the time of submission, our method ranks first on the MOT17 dataset.

2. Methods

In this section, we introduce the main implementation principles of the IOF-Tracker multi-object tracking method and demonstrate a two-stage multi-object tracking framework based on the fusion of temporal and spatial features (R-IoU and optical flow feature fusion model). The implementation principles of this tracking method are illustrated in Figure 1.

The input video frames are first processed by a detector to divide the detected multiple object bounding boxes into high-scoring detections and low-scoring detections. The information within the high-scoring detections is input into a deep appearance extractor (SBS-S50) [12] to extract appearance features, obtaining the deep features of the target images (ReID) [13]. We then use a Kalman filter with camera motion compensation to estimate the predicted bounding boxes and extract the average optical flow vectors of the pixels within the detection and prediction boxes. By matching the optical flow vectors in

these two boxes, we generate an optical flow feature similarity matrix. Next, we calculate the non-overlapping degree of the detection and prediction boxes to obtain the R-IoU feature similarity matrix. We combine these two features with weighted fusion to create a fused R-IoU and optical flow feature similarity matrix, which is then used in the two-stage association tracking process.

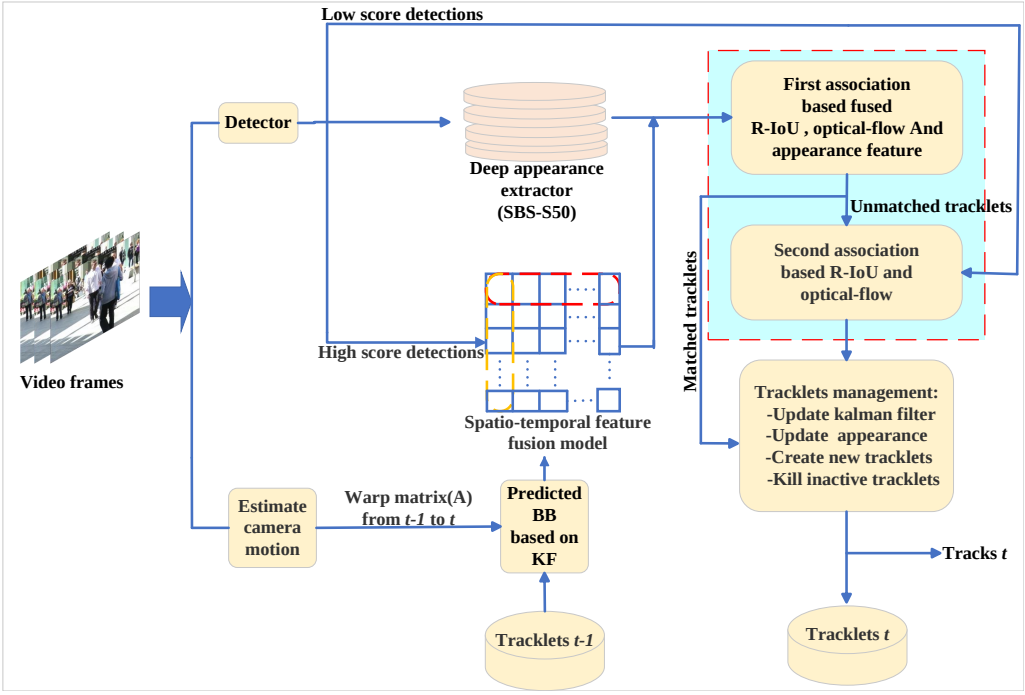

Figure 1. IOF-Tracker multi-object tracking method.

When the tracking enters the association phase, during the first association, we match the target information between the high-scoring detections and predicted boxes using the fused R-IoU and optical flow features, combined with ReID. The first association produces three results: matched tracklets, unmatched detections and remaining tracklets. The matched tracklets proceed to the tracklets updating, unmatched detections build new tracklets, and the remaining tracklets are associated with low-scoring detections for a second association based on the fused R-IoU and optical flow features. The second association generates two results: matched tracklets and re-remaining tracklets. The matched tracklets proceed to the tracklets updating, while the remaining tracklets undergo a similar process. If they remain unmatched for 30 consecutive frames, the re-remaining tracklets are placed in the lost tracklets. We will provide detailed introductions to the two-stage tracking framework and the fused R-IoU and optical flow feature model in subsequent sections. Specifically, our approach is shown step by step in the pseudocodes (Algorithm 1).

Algorithm 1: IOF-Tracker: A Two-Stage Multiple Targets Tracking Method Using Spatial-Temporal Fusion Algorithm.

Data: Video frame set F, initial tracking state set T_{t-1}
Result: Updated tracking state set T_t^*, lost target set L
Input : D_t, P_t
Output: T_t^*, L

F_t // Current frame
$D_t \leftarrow$ `Detect Objects`(F_t) // detection boxes
$D_h, D_l \leftarrow$ `Detect Objects`(F_t) // High scoring detection boxes D_h, low scoring detection boxes D_l
$P_t \leftarrow$ `Predict Tracks`(T_{t-1}) // Predict the target bounding boxes in the next frame
$OF \leftarrow$ `Calculate OF Vectors`(D_t, P_t) // Calculate the average optical flow vectors to represent the temporal information of the target
$OF_d \leftarrow \|((D_t, D_t + 1)$ and $(P_t, P_t + 1))\|$ // Calculate the Euclidean distance of the optical flow vector for consecutive frames
$IoU \leftarrow$ `Calculate IoU`(D_t, P_t) // Calculate the Intersection over Union (IoU)
$IoU_d \leftarrow 1 - IoU$ // IoU_d is an element in the IoU distance matrix
$IoU_dm, OF_dm \leftarrow$ `Calculate Feature Matrix`(IoU_d, OF_d) // Calculate the feature similarity matrices of IoU_d and OF_d separately
$S_f \leftarrow \alpha \times IoU_dm + (1 - \alpha) \times OF_dm$ // S_f is the similarity matrix obtained by fusing IoU and OF features
S // S is the feature similarity matrix obtained by fusing S_f and ReID
$MT_h, UD_h, RT_h \leftarrow$ `Match`$(D_h, P_t, S) \otimes \theta_1$ // The relationship between first stage matching judgment and θ_1
$T_t^* \leftarrow$ `Update Tracks`(MT_h, T_{t-1}) // Update tracking states
$MT_l, UD_l, RRT \leftarrow$ `Match`$(D_l \cup RT_h, P_t, S_f) \otimes \theta_2$ // The relationship between second stage matching judgment and θ_2
$T_t^* \leftarrow$ `Update Tracks`(MT_l, T_t^*) // Update tracking states
foreach *track* $\in RRT$ **do**
  $i \leftarrow 0$
  while $i < 30$ ***and*** *track* $\notin L$ **do**
    $D_{t+i} \leftarrow$ `Detect Objects`(F_{t+i})
    $MT, UD, remainingRRT \leftarrow$ `Match`(D_{t+i}, P_{t+i}, S_f)
    if *track* $\in remainingRRT$ **then**
      $T_t^* \leftarrow$ `Update Tracks`(MT, T_t^*)
      $i \leftarrow 30$
      // If the track is matched within 30 frames, update i and break the loop
    end
    else
      $i \leftarrow i + 1$
      // If the track is not matched within the next 30 frames, add it to the lost track set L
    end
    if $i = 30$ **then**
      $L \leftarrow L \cup \{track\}$
    end
  end
end

2.1. Fusion Model of R-IoU and Optical Flow Features

2.1.1. Optical Flow Features

Optical flow [14] is caused by the movement of foreground objects, camera motion, or both. The optical flow feature is represented by assigning a velocity vector to each pixel in the image. These velocity vectors form the optical flow field. In the absence of moving objects, the optical flow field usually appears continuous and uniform. However, when moving objects are present in the image, their optical flow features will differ from the stationary background, resulting in discontinuous and uneven characteristics in the optical flow field.

In this section, we optimize the optical flow field calculation based on Gunnar Farneback's optical flow algorithm [15]. Farneback's algorithm estimates optical flow by polynomial fitting for each pixel and its neighboring pixels. However, due to the large number and complexity of pixels in the target area, the optical flow vectors obtained for each pixel are significantly affected by noise, making optical flow feature matching difficult. Therefore, we optimize this by averaging the optical flow vectors obtained for each pixel, generating a mean optical flow vector with less noise.

Since the Farneback optical flow method is already integrated into the current OpenCV, we only provide the basic steps for calculating the optical flow vector of a moving target as follows:

Assume that the image sequence is denoted as $I(x, y, t)$, where $X = [x, y]$. An image sequence is represented by each consecutive frame extracted from the video. Assuming constant image brightness, i.e., there is no change in image brightness, the derivative is 0. The formula is as follows:

$$\frac{dI(X,t)}{dt} = \frac{\partial I}{\partial X}\frac{\partial X}{\partial t} + \frac{\partial I}{\partial t} = 0 \tag{1}$$

where, $\frac{\partial X}{\partial t}$ represents velocity in a small time interval, which can be denoted as $\frac{\partial X}{\partial t} = \left[\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}\right] = [\vec{u}, \vec{v}]$. Then, the following equations can be derived:

$$I_x u + I_y v + I_t = 0 \tag{2}$$

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \end{bmatrix} = -I_t \tag{3}$$

By following the basic steps for solving optical flow as described above and combining them with the Farneback optical flow method, we can obtain the horizontal component and vertical component of the optical flow for each pixel.

For a certain pixel (k, l), its optical flow vector can be expressed as $(\vec{u}_{k,l}, \vec{v}_{k,l})$, where $\vec{u}$ and $\vec{v}$ represent the motion components of the pixel in the horizontal and vertical directions, respectively.

Since the magnitude of the resultant vector can be directly used to detect the intensity of motion, such as determining whether there is noticeable motion in a certain area. Direction information helps analyze the movement trend of objects: For example, whether the object is moving left, right, up, or down. The direction information better describes and matches different motion patterns. Therefore, we process the $(\vec{u}_{k,l}, \vec{v}_{k,l})$ components of each pixel point as follows.

First, we perform the average optical flow vector as shown in Equation (4).

$$\vec{u}_{\text{avg}} = \frac{1}{P \times Q}\sum_{k=1}^{P}\sum_{l=1}^{Q}\vec{u}_{k,l} \quad , \quad \vec{v}_{\text{avg}} = \frac{1}{P \times Q}\sum_{k=1}^{P}\sum_{l=1}^{Q}\vec{v}_{k,l} \tag{4}$$

where $P \times Q$ is the number of pixel points in the target area.

Then, the average sum vector $\vec{M}_{avg}$ is expressed as depicted in Equation (5).

$$\vec{M}_{avg} = (\vec{u}_{avg}, \vec{v}_{avg}) \tag{5}$$

The final achieved effect is shown in Figure 2.

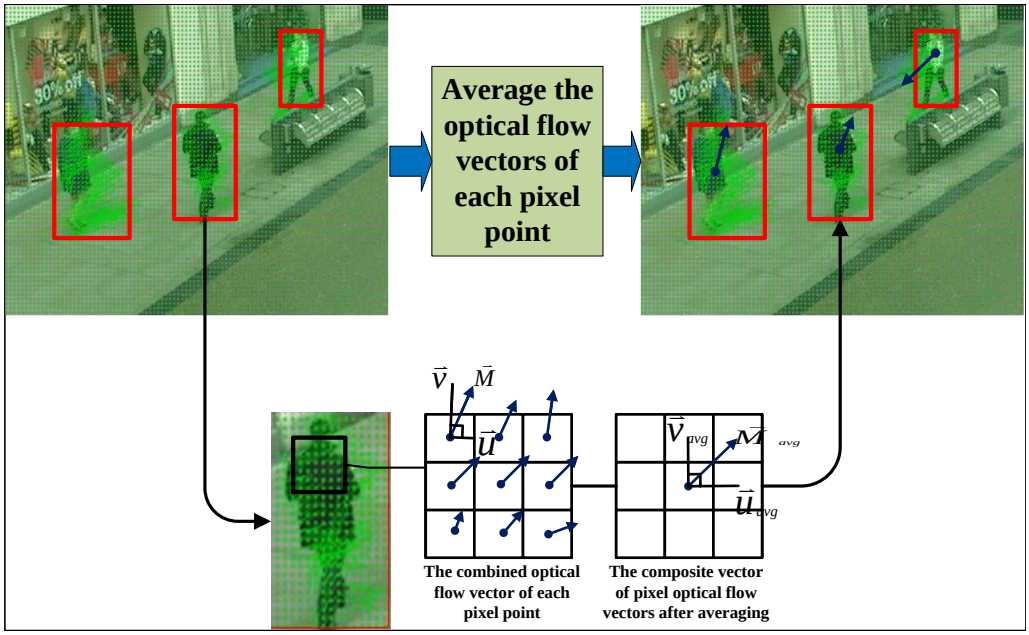

Figure 2. Principle of average optical flow of pixel points.

2.1.2. Optical Flow Feature Matching

In optical flow vector similarity matching, we use the Euclidean distance between optical flow vectors to compare the similarity of the average optical flow vectors of targets in consecutive frames. If the Euclidean distance between two average optical flow vectors falls within a certain set threshold, the targets within the detection and prediction regions are considered likely to be the same target. This is shown in Equation (6).

$$d = \left\| \vec{M}_1 - \vec{M}_2 \right\| \tag{6}$$

where $\vec{M}_1$ and $\vec{M}_2$ are the average optical flow vectors of two target boxes in consecutive frames. By setting a threshold τ, if $d > \tau$, it indicates that the optical flow information of the targets does not match. Conversely, if $d < \tau$, it indicates that the optical flow information of the targets matches.

2.2. Spatial Position Matching

In target tracking and detection tasks, IoU (Intersection over Union) [16] is a commonly used metric to evaluate the positional matching degree between two rectangular boxes (usually the target detection box and the tracking box). IoU value measures the ratio of the overlapping area of the two rectangular boxes to their union area and is typically used to measure the matching degree between the predicted bounding box and the ground truth bounding box.

Spatial position matching calculation steps:

For two consecutive frames of images, suppose that in frame t there is a set of detection boxes $A = \{A_1, A_2, \ldots, A_m\}$, and based on frame $t-1$, the predicted box set for frame t is $B = \{B_1, B_2, \ldots, B_n\}$. All elements within set A are the (x, y, w, h) of each detection box, while all elements within set B are the (x, y, w, h) of each predicted box.

Calculate the IoU values for each pair of detection boxes and predicted boxes, as shown in Equation (7).

$$\mathrm{IoU}(A,B) = \frac{A_p \cap B_q}{A_p \cup B_q}, \quad p = (1,2,3,..,m.), q = (1,2,3,\ldots,n) \tag{7}$$

Distance representation of IoU values:

IoU values range between 0 and 1, where values closer to 1 indicate greater similarity between the two bounding boxes. To convert the IoU value into a "distance" form, we can use the following transformation Formula (8).

$$\text{R-IoU} = 1 - \mathrm{IoU} \tag{8}$$

This transformation method leverages the complement set portion of the IoU. If two boxes are perfectly aligned, IoU = 1, the complement set portion is 0. If two boxes do not overlap at all, IoU = 0, and the complement set portion is 1. Similarly to the optical flow method, a threshold is set. When the complementary set exceeds this threshold, the two boxes are considered non-overlapping. Smaller values of the complementary set indicate higher overlap. This process generates an R-IoU similarity matrix, similar to the optical flow method.

2.3. Feature Fusion

This section includes two fusion modules. The R-IoU, optical flow, and appearance feature fusion model is used in the first stage of the tracking framework, while the R-IoU and optical flow feature fusion model is used in the second stage.

2.3.1. R-IoU and Optical Flow Feature Fusion Model

Assume that $d_{i,j}^{\text{R-I o U}}$ is the value based on R-IoU, and $d_{i,j}^{\text{OptFlow}}$ is the combined distance value based on optical flow. The final feature similarity distance can then be represented as $d_{i,j}^{\text{SFusion}}$. The formula is shown in (9).

$$d_{i,j}^{SFusion} = \alpha \cdot d_{i,j}^{\text{R-I o U}} + (1-\alpha) \cdot d_{i,j}^{\text{OptFlow}} \tag{9}$$

where α is a weighting factor used to balance the contributions of R-IoU and optical flow in the tracking task. The final feature similarity matrix is composed of $d_{i,j}^{\text{SFusion}}$. Through this process, the R-IoU and optical flow information fusion model is obtained, which will be applied to the second association stage of the tracking framework.

The R-IoU and optical flow feature fusion model is shown in Figure 3. In branch 1, the video frame image is input, and the detector detects the target boxes in the current frame. Meanwhile, the optical flow algorithm obtains the overall optical flow of the image and extracts the average optical flow vector of the target area. In branch 2, the estimator uses information from the previous frame to predict the bounding boxes for the current frame and then extracts the average optical flow vector within the predicted boxes. Subsequently, the R-IoU method is used to compute the complementary set between detection boxes and predicted boxes, and this value is compared with threshold δ_1. Values greater than δ_1 indicate that the detection box and prediction box do not match using the R-IoU method, while values less than δ_1 indicate a match. Similarly, the average optical flow vectors within the detection and prediction boxes are combined and measured using a certain method, and this value is compared with threshold δ_2. Values greater than δ_2 indicate that the boxes do not match using the optical flow feature method, while values less than δ_2 indicate a match. Then, the R-IoU feature similarity matrix and the optical flow feature similarity matrix are fused using a matrix weighting method to obtain the fused feature

similarity matrix. The Hungarian algorithm (HA) is applied to this fused matrix to perform one-to-one matching between detection boxes and prediction boxes, ultimately obtaining the optimal tracking boxes.

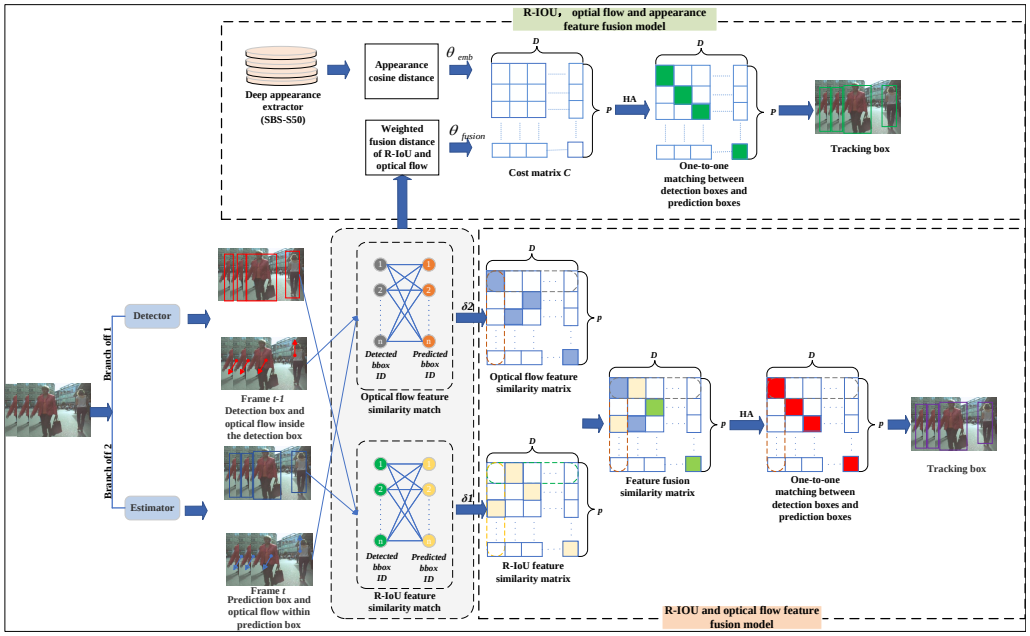

Figure 3. Feature fusion model.

2.3.2. R-IoU, Optical Flow and Appearance Feature Fusion Model

The model fuses three features: the R-IoU feature, the optical flow feature and the appearance feature obtained from ReID. As shown in Formula (10), When both the appearance cosine distance $d_{i,j}^{cos}$ and the fusion model distance $d_{i,j}^{\text{SFusion}}$ are less than the given thresholds, we use the appearance cosine distance as the new fused distance $\widehat{d}_{i,j}^{cos}$; Otherwise, we set the new fused distance $\widehat{d}_{i,j}^{cos}$ to 1. Then, we use the minimum value between the spatiotemporal fusion model $\widehat{d}_{i,j}^{cos}$ and the new distance $\widehat{d}_{i,j}^{cos}$ as the final value for the elements in the cost matrix C. Our formula for the R-IoU and optical flow feature fusion combined with ReID can be expressed as follows:

$$\widehat{d}_{i,j}^{\cos} = \begin{cases} d_{i,j}^{\cos}, \left(d_{i,j}^{\cos} < \theta_{\text{emb}}\right) \cap \left(d_{i,j}^{\text{SFusion}} < \theta_{\text{fusion}}\right) \\ 1, \text{ otherwise} \end{cases}, \quad C_{i,j} = \min\left\{d_{i,j}^{\text{SFusion}}, \widehat{d}_{i,j}^{\cos}\right\} \tag{10}$$

where $C_{i,j}$ represents the (i, j) element of the cost matrix C, $d_{i,j}^{cos}$ represents the appearance cosine distance of the target within the prediction box and the detection box, $\widehat{d}_{i,j}^{cos}$ is the new distance score after fusion, $d_{i,j}^{\text{SFusion}}$ is the distance score after fusion of R-IoU and optical flow motion features. θ_{emb} and θ_{fusion} are both set thresholds, where we set both θ_{emb} and θ_{fusion} to 0.5. Through this process, we can obtain a fusion model that simultaneously integrates R-IoU, optical flow features, and appearance features. For the cost matrix C, the Hungarian algorithm can be used to find the optimal one-to-one matching results between detection boxes and prediction boxes. This model will be applied to the first association stage of the tracking framework. The R-IoU, optical flow and appearance feature fusion model is shown in Figure 3.

Based on the aforementioned content, we can obtain our spatiotemporal feature information model (i.e., the R-IoU, optical flow and appearance feature fusion model for the first association stage, and the R-IoU and optical flow feature fusion model for the second association stage). This model will be used in the matching and association module

of the tracking framework in Section 2.4 to associate spatiotemporal features of targets across consecutive frames.

2.4. Two-Stage Tracking Framework

The primary function of the two-stage tracking framework is to process high and low-scoring detection boxes through different association methods in separate stages, followed by a comparative loop to filter out high-quality target detection boxes. First, for high-scoring detection boxes, they participate in the first-stage association, which results in three outcomes: matched tracklets (MT), unmatched detections (UD), and remaining tracklets (RT). Matched tracklets enter tracklets update (MTU), unmatched detections enter new tracklets (UDN), and remaining tracklets (RT) are those that are neither new nor matching with previous trajectories and require further correction. Then, UDN continues with the first association in subsequent frames, while RT associates with low-scoring detection boxes in the second round. This results in matched tracklets (MT) and re-remaining tracklets (RRT). Matched tracklets enter tracklets updation (MTU), and re-remaining tracklets (RRT) undergo similar operations as before. If the loop repeats for 30 frames without a match, the RRT is discarded. The principle is illustrated in Figure 4.

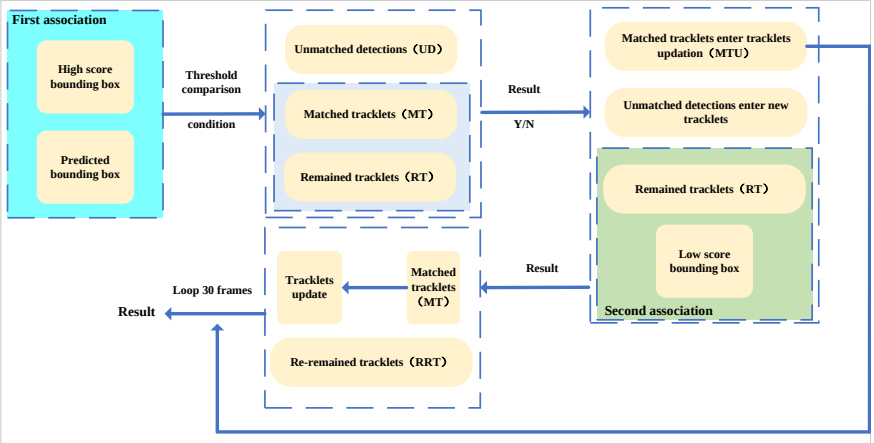

Figure 4. The two-stage tracking framework

3. Experimental Results and Analysis

3.1. Dataset Section

This experiment was conducted on the two most popular datasets in the field of multi-target tracking under unconstrained environments for pedestrian detection and tracking: MOT17 [17] and MOT20 [18]. MOT17 contains video sequences captured by both static and moving cameras. In contrast, MOT20 includes crowded scenes and has added new video sequences and more complex scenarios, significantly enhancing the test for target tracking algorithms. Both datasets contain training and test sets but no validation set.

3.2. Experimental Environment

In this paper, we conducted all experiments and training using the PyTorch framework on a desktop computer equipped with the operating system Ubuntu 20.04, a processor of 12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz, and a GeForce RTX 3080ti GPU. For a fairer comparison, we directly used a publicly available detector, which was trained on MOT17 and MOT20 through ablation studies. For optical flow feature extraction, we used the trained detector to detect targets and extract optical flow within the target boxes. In all experiments, we used a two-stage tracking algorithm with a default detection score threshold of 0.6. In the linear assignment step, if the similarity distance between the

detection box and the tracking box exceeds 0.5, the match is rejected. For lost tracks, we keep them for 30 frames in case they reappear.

3.3. Benchmark Evaluation Metrics for Multi-Object Tracking

This paper uses six evaluation metrics [19] to assess the performance of the tracking algorithm.

(1) FP (False Positive): False alarms indicating that a tracking trajectory was generated when there was no real object.
(2) FN (False Negative): Missed detections indicating that a real object was present but no tracking trajectory was generated.
(3) IDF1 (IDF1 Score): This combines the precision and recall of identifying objects across frames and measures the tracker's ability to maintain consistent object identities throughout the video sequence.
(4) IDs (ID Switches): The number of times the target ID changes, primarily measuring the consistency of the target tracking trajectory.
(5) MOTA (Multiple Object Tracking Accuracy): Tracking accuracy, measuring the performance of the tracking algorithm in detecting objects and maintaining tracking trajectories.
(6) HOTA (Higher-Order Tracking Accuracy): This is a metric used to evaluate the performance of multi-object tracking. Unlike traditional metrics such as MOTA and IDF1, HOTA aims to comprehensively reflect the tracking algorithm's capabilities in localization and identification. It considers several higher-order factors, such as the accuracy of objects, the consistency of relative positions, and the complexity of maintaining target identities.

3.4. Weight Selection Experiment

To determine the weight coefficients in our R-IoU and optical flow fusion model, we conducted tests to select these weight coefficients. Due to the limited number of submissions allowed by MOTChallenge for official user results, we used the latter half of the MOT17 training set as the standard for weight selection tests, based on the SDP detector provided by the official.

We selected weights with high MOTA and IDF1 scores and a low number of ID switches as the optimal weight coefficients. Here, α represents the weight coefficient for the IoU feature similarity matrix, and in the table, we use β to denote $(1-\alpha)$, which represents the weight coefficient for the optical flow feature similarity matrix. From the experiments in Table 1, which tested each training set of MOT17-SDP, it was found that to achieve both the highest MOTA and IDF1 scores and the fewest ID switches, the weight α should be selected as 0.8 and β as 0.2. This combination of weights proved to be relatively optimal. Through the weight selection experiment, when α is set to 0.8 and β to 0.2, the three evaluation metrics of the MOT17-SDP training set all reached the ideal values we desired.

Table 1. Weight selection test based on the latter half of the MOT17 training set (↑: The higher the parameter, the better. ↓: The lower the parameter, the better. ✓: indicates selection. Bold font: indicates the best value in that column).

α β	0	0.1	0.2	0.3	0.4	0.5	MOTA ↑	IDF1 ↑	IDs ↓
1.0	✓						91.374	84.823	339
0.9		✓					91.364	84.838	342
0.8			✓				**91.404**	**85.744**	**320**
0.7				✓			90.822	83.809	399
0.6					✓		86.489	74.332	1050
0.5						✓	80.950	69.320	1550

3.5. Ablation Study

Our ablation study primarily aims to verify whether our R-IoU and optical flow feature fusion model can improve the benchmark metrics for multi-object tracking. Due to the official restrictions imposed by MOTChallenge on the number of attempts, researchers can submit results to the test server, we continued to use part of the MOT17-SDP training set for testing.

As shown in Table 2, adding optical flow features alone to the baseline significantly reduces the benchmark tracking metrics. This issue arises because the MOT17 dataset is a multi-object tracking dataset with a high pedestrian density. The interactions between pedestrians during overlapping have a large impact on the optical flow field. Therefore, the tracking methods that rely solely on optical flow perform poorly. On the other hand, only using R-IoU features considers the spatial position information of bounding boxes, but unable to effectively handle partial occlusion scenarios. When a target is partially occluded, the R-IoU value may drop significantly, leading to incorrect matches or tracking loss. However, combining R-IoU and optical flow features can address or mitigate some of the above issues to some extent. Optical flow provides rich information about the target's motion, and even in the case of partial occlusion. Because optical flow can still be inferred effectively based on the motion patterns in the unoccluded regions. It handles situations where the target is partially occluded, thereby reducing incorrect matches or tracking loss. By fusing these two features, the robustness and accuracy of tracking can be improved. Additionally, the data in Table 2 also shows that the baseline with the fusion of R-IoU and optical flow features can achieve excellent scores in IDF1 and IDs.

Table 2. Impact of the addition of the R-IoU and optical flow feature fusion model on the overall tracking algorithm performance (↑: The higher the parameter, the better. ↓: The lower the parameter, the better. ✓: indicates selection. Bold font: indicates the best value in that column).

Method	R-IoU	Opt-Flow	MOTA ↑	IDF1 ↑	IDs ↓
Baseline	✓		91.147	84.693	339
		✓	68.384	63.784	3446
	✓	✓	**91.404**	**85.744**	**320**

The ablation experiment in Table 3, which involves adding the R-IoU and optical flow feature fusion model in stages, shows that solely integrating our module into either the first or second stage of the two-stage tracker does not achieve the most ideal results. Only by using the R-IoU spatial features and optical flow temporal features in both stages can the test benchmark metrics reach relatively better results.

Table 3. The impact of applying the R-IoU and optical flow feature fusion model at different stages of the tracking framework on the overall tracking algorithm performance (↑: The higher the parameter, the better. ↓: The lower the parameter, the better. ✓: indicates selection. Bold font: indicates the best value in that column).

Method	Stage 1	Stage 2	MOTA ↑	IDF1 ↑	IDs ↓
Baseline + Fusion model	✓		**91.425**	85.557	329
		✓	91.368	84.513	360
	✓	✓	91.404	**85.744**	**320**

3.6. Qualitative Analysis of Multi-Object Tracking Performance

3.6.1. Performance Analysis of Handling Specific Target Occlusion Problems

To more intuitively validate the performance of our multi-object tracking method in handling occlusion issues during target tracking, we have applied this method to videos from the MOT15 and MOT17 datasets.

In Figure 5 we used the popular BYTEtrack method as well as our IOF-Tracker method to test the tracking performance on two videos from the MOT15 and MOT17 datasets where occlusion is more pronounced. The figure shows that when using the BYTEtrack method, the pedestrian with ID 19 in the MOT15 dataset and the pedestrian with ID 3 in the MOT17 dataset are assigned new IDs of 39 and 34, respectively, after being occluded. However, when using our IOF-Tracker method, the pedestrian with ID 19 in the MOT15 dataset and the pedestrian with ID 3 in the MOT17 dataset retain their original IDs of 19 and 3, respectively, even after being occluded. Therefore, our method demonstrates superior performance in handling occlusion issues during multi-object tracking.

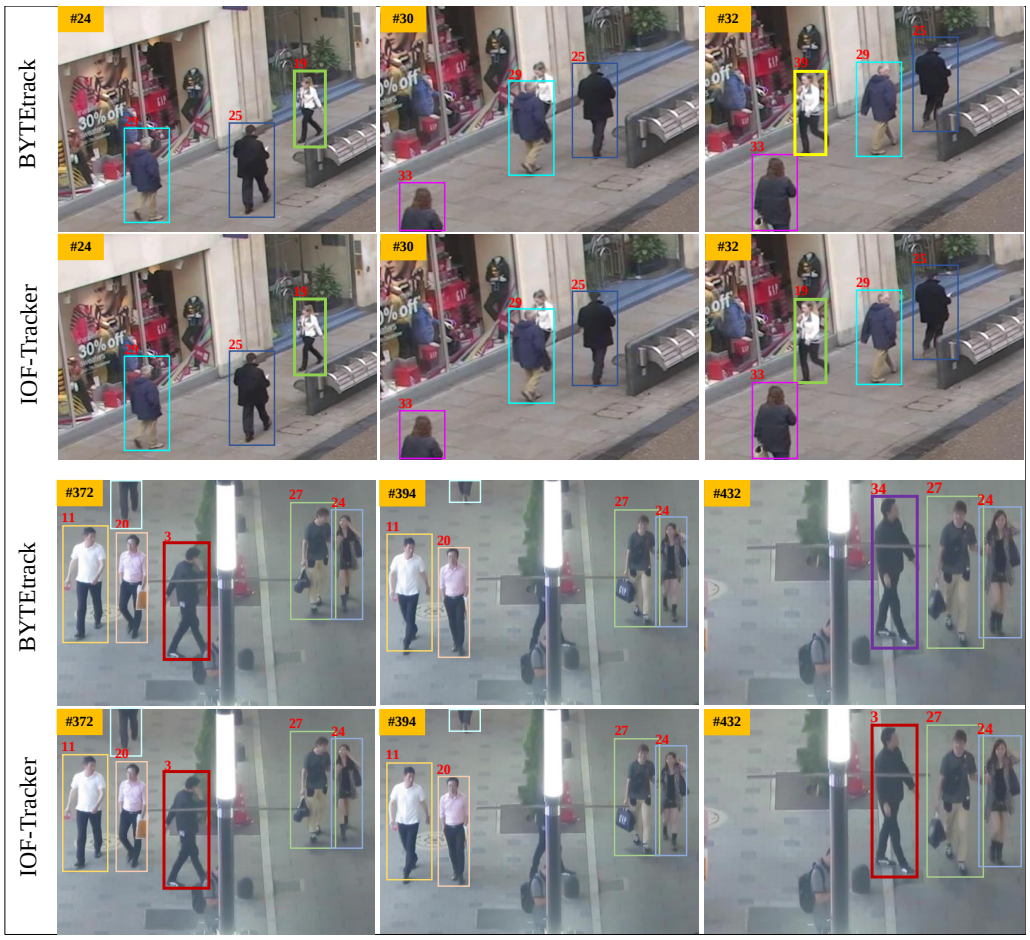

Figure 5. Performance analysis of handling occlusion problems (Different colored target boxes represent different ID information).

3.6.2. Performance Analysis of Multi-Object Tracking

To visually validate the performance of our multi-object tracking method in handling real-world tracking tasks, we present the test tracking results in MOT17 datasets, as shown in Figure 6.

Figure 6 presents a qualitative comparison of the performance between our IOF-Tracker method and the two best-performing multi-object tracking methods currently available. As seen in the figure, BYTEtrack assigns the pedestrian labeled as 25 in frame

88 an ID of 88 after prolonged occlusion. In contrast, both our IOF-Tracker method and BoTSORT correctly identify the pedestrian as number 20 in frame 88 and maintain this ID after the long occlusion. This indicates that BYTEtrack undergoes multiple ID switches before frame 88 video information arrives and performs poorly in handling occlusions, while our IOF-tracker method and BoTSORT handle occlusion issues better in the tracking process. Upon comparing the newly appearing pedestrian target in frame 382, we find that BYTEtrack and BoTSORT assign pedestrian IDs of 139 and 98, respectively, whereas our IOF-Tracker method assigns the ID of 91. This suggests that while BoTSORT maintains longer continuous tracking than BYTEtrack, it still generates a significant number of IDs during the tracking process. In contrast, our IOF-Tracker method clearly addresses the issue of frequent ID switching during long continuous tracking. Therefore, our tracking method demonstrates significant effectiveness in handling occlusion during multi-object tracking in real-time scenarios and managing issues related to frequent ID switching.

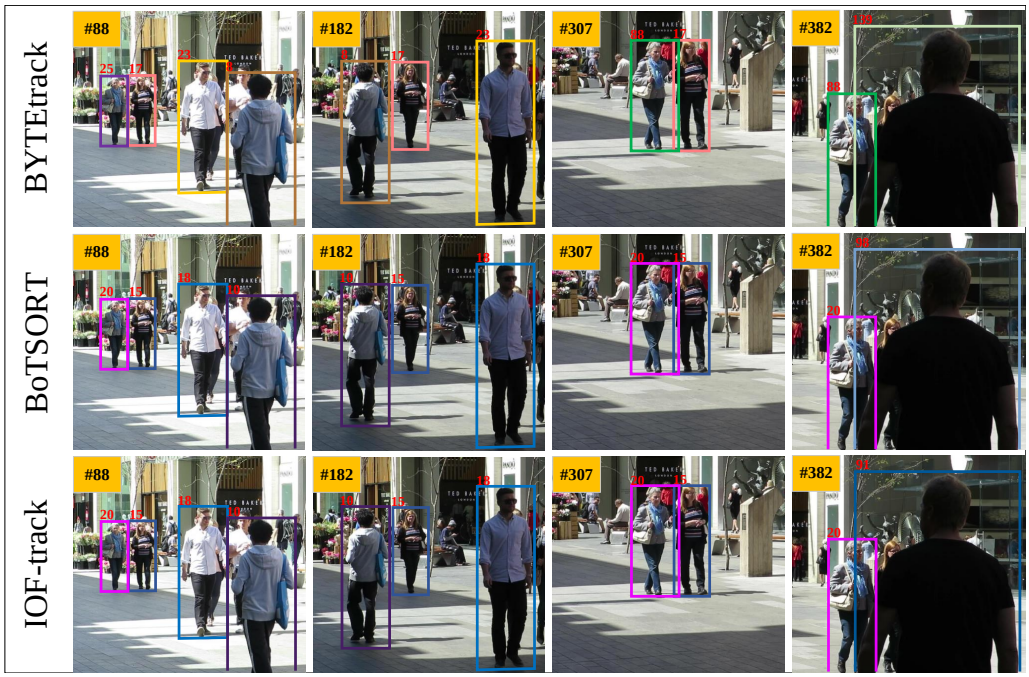

Figure 6. Qualitative analysis of multi-object tracking performance (Different colored target boxes represent different ID information).

3.7. Benchmark Evaluation

We conducted tests on our IOF-Tracker algorithm using the MOT17 and MOT20 test sets, comparing the performance of our feature fusion combined tracker in Tables 4 and 5. Our tests all used publicly available detectors, and all results were directly obtained from the official MOTChallenge evaluation server. Since the speed of each method depends on the device they are implemented on, and tracking based on detection usually does not calculate the time spent on detection, we do not compare the FPS performance here.

Through the benchmark scores in Table 4, on MOT17, our feature fusion model combined tracker performs excellently in the main evaluation metrics, namely MOTA, IDF1, IDs, and HOTA, outperforming all other advanced trackers. This indicates that the multi-object tracker with our feature fusion module can exhibit a certain degree of stability in tracking some difficulties. In Table 5, on the MOT20 dataset, the only metrics that reached the optimal performance were MOTA and HOTA. We analyzed the reasons for this: firstly, the population density in the MOT20 dataset is much larger than that in MOT17, resulting in more frequent crowd interactions and a greater impact on the optical flow field, causing frequent interactions between pedestrians and leading to chaos in the optical flow field.

Secondly, some scenes in MOT20 are in dimly lit night conditions with large variations in lighting intensity. Changes in lighting intensity (such as shadows, reflections, etc.) can cause variations in surface brightness, which poses a challenge for optical flow calculation. Therefore, using our module for multi-object tracking in extremely crowded scenarios like MOT20 presents certain challenges.

Table 4. Results on MOT17 challenge test set (↑: The higher the parameter, the better. ↓: The lower the parameter, the better. Bold font: indicates the best value in that column).

Tracker	MOTA ↑	IDF1 ↑	HOTA ↑	FP ↓	FN ↓	IDs ↓
UTM [20]	63.5	65.1	52.5	33,683	170,352	1686
CenterTrack [21]	67.8	64.7	52.2	18,498	160,332	3039
SOTMOT [22]	71.0	71.9	-	39,537	118,983	5184
TransCenter [23]	73.2	62.2	54.5	23,112	123,738	4614
FairMOT [24]	73.7	72.3	59.3	27,507	117,477	3303
SiamMOT [25]	76.3	72.3	-	-	-	-
TransMOT [26]	76.7	75.1	61.7	36,231	93,150	2346
OCSORT [27]	78.0	77.5	63.2	**15,129**	107,055	1950
StrongSORT [28]	78.3	78.5	63.5	27,876	86,205	1446
ByteTrack [3]	78.9	77.2	62.8	25,491	83,721	2196
FeatureSORT [29]	79.6	77.2	63.0	29,588	83,132	2269
Ours	**80.5**	**79.9**	**64.9**	27,245	**81,653**	**1370**

Table 5. Results on MOT20 challenge test set (↑: The higher the parameter, the better. ↓: The lower the parameter, the better. Bold font: indicates the best value in that column).

Tracker	MOTA ↑	IDF1 ↑	HOTA ↑	FP ↓	FN ↓	IDs ↓
MLT [30]	48.9	54.6	43.2	45,660	216,803	2187
FairMOT [24]	61.8	67.3	54.6	103,440	88,901	5243
TransCenter [23]	61.9	50.4	-	45,895	146,347	4653
SiamMOT [25]	67.1	69.1	-	-	-	-
SOTMOT [22]	68.6	71.4	-	57,064	101,154	4209
OCSORT [27]	75.7	**76.3**	62.4	19,067	105,894	942
ByteTrack [3]	75.7	74.9	60.9	26,249	**87,594**	1223
StrongSORT [28]	72.2	75.9	61.5	**16,632**	117,920	**770**
FeatureSORT [29]	76.6	75.1	61.3	25,083	95,027	1081
Ours	**77.7**	75.0	**62.0**	25,019	88,959	1530

4. Conclusions

In this paper, we propose the IOF-Tracker, a two-stage multi-object tracking method that integrates spatial and temporal feature fusion, aiming to enhance tracking accuracy and robustness by combining static spatial positions with dynamic temporal motion. To reduce the impact of noise on optical flow information, we estimate the temporal features of the target using the average optical flow of each pixel. We represent the spatial information using the R-IoU value between the target detection box and the predicted box. By fusing temporal and spatial features and applying them to a two-stage association tracking algorithm, the tracker can fully utilize the spatiotemporal features of the target for association matching during the tracking process, thereby reducing the occurrence of ID switches due to target occlusion and interaction in multi-object tracking. The robustness of optical flow vectors in estimating subtle movements across consecutive frames also contributes to high scores in the tracking benchmark IDF1. In practical applications, such as autonomous driving or intelligent security scenarios, it is only necessary to replace the corresponding dataset and train the detector so that it can be used for the detection stage of the tracking targets. This allows the IOF-Tracker multi-object tracking method to perform real-time

multi-object tracking in different application scenarios. Although this method can address tracking issues due to occlusion to some extent, especially in sparsely populated scenes where it performs exceptionally well, in densely populated scenes, frequent interactions between targets can affect the stability of the optical flow field, potentially leading to less than ideal tracking performance. Therefore, in future work, we will continue to improve the spatiotemporal feature fusion module and introduce an adaptive weight adjustment mechanism. This will enable the fusion module to dynamically adjust the weights of spatial and temporal features according to different tracking scenarios. This enhancement aims to improve the robustness of the feature fusion module in handling dense target scenarios, thereby further addressing the issue of lower tracking performance in such scenarios.

Author Contributions: Conceptualization, H.L.; Methodology, H.L. and Y.Z.; Software, Y.Z.; Validation, Y.Z.; Formal analysis, H.L.; Investigation, Y.W. and X.G.; Resources, H.L.; Data curation, P.D. and X.G.; Writing—original draft preparation, Y.Z.; Writing—review and editing, H.L. and Y.Z.; Visualization, P.D. and X.G.; Supervision, H.L.; Project administration, H.L.; Funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Shandong Province Natural Science Foundation under grant ZR2021QF077, the Foundation for the Key Research and Promotion of Henan Province (Science and Technology) under grant 242102210118, the Key R & D Plan of Shandong Province (Innovation Ability Improvement Project of Small and Medium-sized Science and Technology Enterprises) under grant 2024TSGC0123, the Doctoral Research Fund Project of Shandong Jianzhu University under grant XNBS20081.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors express their sincere gratitude to Hanbin Bao from Bresee Technology Co., Ltd. for providing the experimental equipment, technical support, and comprehensive training on theoretical content essential for this study. Additionally, the authors would like to extend their heartfelt thanks to all individuals who contributed assistance throughout the research.

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

References

1. Larsen, M.V.; Rolfsjord, S.; Gusland, D.; Ahlberg, J.; Mathiassen, K. BASE: Probably a Better Approach to Multi-Object Tracking. *arXiv* **2023**, arXiv:2309.12035.
2. Liu, Z.; Wang, X.; Wang, C.; Liu, W.; Bai, X. SparseTrack: Multi-Object Tracking by Performing Scene Decomposition based on Pseudo-Depth. *arXiv* **2023**, arXiv:2306.05238.
3. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-Object Tracking by Associating Every Detection Box. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23 October 2022; pp. 1–21.
4. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv* **2022**, arXiv:2206.14651.
5. Xiao, X.; Feng, X. Multi-Object Pedestrian Tracking Using Improved YOLOv8 and OC-SORT. *Sensors* **2023**, *23*, 8439. [CrossRef] [PubMed]
6. Bae, S.H.; Yoon, K.J. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 595–610. [CrossRef] [PubMed]
7. Song, Y.M.; Jeon, M. Online Multiple Object Tracking with the Hierarchically Adopted GM-PHD Filter Using Motion and Appearance. In Proceedings of the 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Republic of Korea, 26–28 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–4.
8. Ince, S.; Konrad, J. Occlusion-Aware Optical Flow Estimation. *IEEE Trans. Image Process.* **2008**, *17*, 1443–1451. [CrossRef] [PubMed]
9. Zach, C.; Pock, T.; Bischof, H. A Duality Based Approach for Realtime TV-L1 Optical Flow. In Proceedings of the 29th DAGM Symposium on Pattern Recognition, Heidelberg, Germany, 12–14 September 2007; pp. 214–223.

10. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
11. Liu, H.; Chang, F.; Liu, C.; Liang, F. Occlusion-Resistant Multi-Target Tracking Based on Spatiotemporal Progressive Feature Model. *Control Decis.* **2019**, *34*, 2171–2177.
12. Shen, J.; Yang, H.; Song, W. Panoramic Multi-Target Tracking Model Incorporating Human Features and Motion Model. In Proceedings of the 2024 4th International Conference on Computer Communication and Artificial Intelligence (CCAI), Xi'an, China, 4 May 2024; pp. 100–105.
13. Liang, C.; Zhang, Z.; Zhou, X.; Li, B.; Zhu, S.; Hu, W. Rethinking the competition between detection and reid in multiobject tracking. *IEEE Trans. Image Process.* **2022**, *31*, 3182–3196. [CrossRef] [PubMed]
14. Sharmin, N. Optimal Filter Estimation for Lucas-Kanade Optical Flow. *Sensors* **2012**, *12*, 12694–12709. [CrossRef]
15. Ma, Z.; Wang, T.; Xu, S.; Mu, X.; Wang, Q.; Guo, Q. Moving object Detection Based on Farneback Optical Flow. In Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023; pp. 7350–7355.
16. Jia, S.; Song, Y.; Ma, C.; Yang, X. Iou Attack: Towards Temporally Coherent Black-Box Adversarial Attack for Visual Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6709–6718.
17. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A Benchmark for Multi-Object Tracking. *arXiv* **2016**, arXiv:1603.00831.
18. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; Leal-Taixé, L. MOT20: A Benchmark for Multi-Object Tracking in Crowded Scenes. *arXiv* **2020**, arXiv:2003.09003.
19. 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]
20. You, S.; Yao, H.; Bao, B.K.; Xu, C. UTM: A Unified Multiple Object Tracking Model with Identity-Aware Feature Enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 21876–21886.
21. Yang, H.S.; Sim, C.B.; Jung, S.H. CenterTrack-EKF: Improved Multi Object Tracking with Extended Kalman Filter. *Smart Media J.* **2024**, *13*, 9–18.
22. Zheng, L.; Tang, M.; Chen, Y.; Zhu, G.; Wang, J.; Lu, H. Improving Multiple Object Tracking with Single Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2453–2462.
23. Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; Alameda-Pineda, X. Transcenter: Transformers with Dense Queries for Multiple-Object Tracking. *arXiv* **2021**, arXiv:2103.15145.
24. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. Fairmot: On the Fairness of Detection and Re-Identification in Multiple Object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
25. Liang, C.; Zhang, Z.; Zhou, X.; Li, B.; Hu, W. One More Check: Making "Fake Background" Be Tracked Again. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 22–29 June 2022; Volume 36, pp. 1546–1554.
26. Chu, P.; Wang, J.; You, Q.; Ling, H.; Liu, Z. Transmot: Spatial-Temporal Graph Transformer for Multiple Object Tracking. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 1–5 January 2023; pp. 4870–4880.
27. Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; Kitani, K. Observation-Centric Sort: Rethinking Sort for Robust Multi-Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 19–25 June 2023; pp. 9686–9696.
28. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. Strong Sort: Make DeepSort Great Again. *IEEE Trans. Multimed.* **2023**, *25*, 8725–8737. [CrossRef]
29. Hashempoor, H.; Koikara, R.; Hwang, Y.D. FeatureSORT: Essential Features for Effective Tracking. *arXiv* **2024**, arXiv:2407.04249.
30. Zhang, Y.; Sheng, H.; Wu, Y.; Wang, S.; Ke, W.; Xiong, Z. Multiplex Labeling Graph for Near-Online Tracking in Crowded Scenes. *IEEE Internet Things J.* **2020**, *7*, 7892–7902. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.