electronics

MDPI

Article

DetTrack: An Algorithm for Multiple Object Tracking by Improving Occlusion Object Detection

Xinyue Gao [1,2], Zhengyou Wang [1,*], Xiaofan Wang [3], Shuo Zhang [1], Shanna Zhuang [1] and Hui Wang [1]

[1] School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, China; 1202010040@student.stdu.edu.cn (X.G.); 1202110098@student.stdu.edu.cn (S.Z.); zhuangsn@stdu.edu.cn (S.Z.); wangh@stdu.edu.cn (H.W.)
[2] Tangshan Research Institute, Beijing Institute of Technology, Tangshan 063099, China
[3] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China; wangxf@ios.ac.cn
* Correspondence: zhengyouwang@stdu.edu.cn

Abstract: Multi-object tracking (MOT) is an important problem in computer vision that has a wide range of applications. Currently, object occlusion detecting is still a serious challenge in multi-object tracking tasks. In this paper, we propose a method to simultaneously improve occluded object detection and occluded object tracking, as well as propose a tracking method for when the object is completely occluded. First, motion track prediction is utilized to improve the upper limit of occluded object detection. Then, the spatio-temporal feature information between the object and the surrounding environment is used for multi-object tracking. Finally, we use the hypothesis frame to continuously track the completely occluded object. Our study shows that we achieve competitive performances compared to the current state-of-the-art methods on popular multi-object tracking benchmarks such as MOT16, MOT17, and MOT20.

Keywords: computer vision; multiple object tracking; object detection; Kalman filtering; data association

check for updates

Citation: Gao, X.; Wang, Z.; Wang, X.; Zhang, S.; Zhuang, S.; Wang, H. DetTrack: An Algorithm for Multiple Object Tracking by Improving Occlusion Object Detection. *Electronics* **2024**, *13*, 91. https://doi.org/10.3390/electronics13010091

Academic Editors: Kaihua Zhang, Shunli Zhang, Yang Yang and Xin Yu

Received: 9 November 2023
Revised: 4 December 2023
Accepted: 15 December 2023
Published: 25 December 2023

Copyright: © 2023 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

Occlusion is a typical problem in multi-object tracking, and it is also one of the major challenges for object tracking in real-life scenarios [1]. Researchers often use feature similarity to solve occlusion problems in multi-object tracking, such as color features, contour features, and introducing attention mechanisms [2]. Another solution is a multi-object tracking algorithm based on particle filtering [3]. Fazli et al. [4] proposed a particle algorithm that combines scale features with object color information, but the tracking performance decreased if the object colors were similar. Roccetti [5] detected the convergence of two trajectories by preserving the time evolution of extreme point trajectories, thereby tracking the occluded object. Liu et al. [6] proposed an adaptive template update algorithm to solve tracking occlusion problems, but the computational complexity was high and the practical execution was time-consuming. Currently, there are relatively few algorithms to solve the problem of complete occlusion.

In traditional multi-object tracking algorithms, data association problems are transformed into various problems. For example, the minimum cost flow problem, the discrete continuous optimization problem, the maximum multi-clique problem, or other methods can be used to solve the problem of multi-object tracking data association. Bewley et al. [7] first proposed a deep learning-based Sort algorithm in 2016; they also added appearance features and proposed the classic DeepSort algorithm in 2017 [8]. Thus far, the DeepSort algorithm is still receiving great attention from researchers. Furthermore, those who study it usually improve it.

Recently, some multi-object tracking algorithms have been based on Convolutional Neural Networks (CNNs) or pedestrian recognition networks. Yin et al. [9] combined a motion model and similarity model into a network structure. Wang et al. [10] proposed

a new multi trajectory object tracking framework based on dynamic attention guidance, which induces more dynamic appearance models. Although researchers have proposed a large number of multi-object tracking algorithms [11–13], there are still certain serious challenges in multi-object tracking tasks.

Multi-object tracking performance is often affected by object detection: the upper limit of object detection determines the upper limit of the multi-object tracking performance. For the problem of occlusion in multi-object tracking, we propose a tracking algorithm, DetTrack, to improve occlusion object detection. Since object detection has upper limits in tracking performance, we combined spatio-temporal features with track prediction to restore low-scoring object detection and enhance data association. When the object is completely occluded, the object motion track prediction box is used as a hypothesis box to continue tracking. Therefore, this paper will comprehensively solve the problem of occlusion object detection loss and occlusion multi-object tracking.

2. Materials and Methods

In this section, we mainly introduce YOLO for track prediction and multi-object tracking algorithms.

2.1. YOLO

Object detection algorithms include R-CNN series algorithms and YOLO series algorithms. Compared to R-CNN series algorithms, the YOLO series algorithms have a faster detection speed; thus, we will pay more attention to the YOLO series algorithms. The YOLO algorithm detects objects by dividing an image into grids, and it then uses the modified GoogLeNet [14] as the backbone network to extract features. The feature map of the output layer is designed to output the coordinates of the object frame, the object frame score, and the category score. YOLOv3 extracts features from the Darknet-53 network, and it uses sampling and cascading methods to retain fine-grained features for small object detection. Additionally, it is detected on 3 different scales. This allows YOLOv3 to detect various objects. The detection speed is as fast as YOLO and YOLOv2, and the accuracy is almost the same as SSD but the speed is nearly three times faster [15]. YOLOv5 will be open source and, soon after the release of YOLOv4 in 2020, will improve the accuracy of the network's detection of objects of different sizes to a certain extent. In 2021, on the basis of YOLOv3, YOLOX was proposed by the Vision Research Institute. It involved building an anchor-free end-to-end object detection framework, which also reached the first-class detection level.

2.2. Kalman Filtering

Kalman filtering is an algorithm that uses the state equation of a linear system to optimally estimate the state of the system input and output observations. A typical example of Kalman filtering is to predict the coordinate position and velocity of an object from a finite, noise-containing sequence of observations of the object's position. It can be found in many engineering applications (including radar and computer vision). The Kalman filter is a linear optimal filter that can be computed iteratively over time and is suitable for object tracking in-depth maps. In order to solve the tracking problem of object occlusion, the Kalman filter method is used to obtain the object's motion trajectory information, and this motion trajectory information is used to cope with the object being occluded. Furthermore, the online trajectory record is used to help the tracker localization, and the tracker position can better update the observation of the Kalman filter to achieve a mutually reinforcing effect.

2.3. Multi-Object Tracking

Multi-object tracking can be divided into online tracking and offline tracking. The main difference between them is whether the information of future video frames is used to improve the current tracking results in tracking process. Online multi-object tracking only

uses the information of current frame and history frame, while offline tracking method can also use the information of future frames [16,17]. Generally, due to less information needed for an offline muti-object tracker, the performance of an offline multi-object tracker is superior to that of an online multi-object tracker. However, offline multi-object trackers cannot be used for real-time tasks. Therefore, in practical application, online multi-object trackers are mainly used for real-time tracking tasks [18], while offline multi-object trackers are just used to analyze recorded videos.

3. Methods

The flow chart for our method is shown in Figure 1. The red detection box in the F_t frame extracts the spatio-temporal features of the object, while the F_{t-1} frame records the historical features of the object trajectory and can predict the position of the object in the F_t frame through Kalman filtering. We use spatio-temporal features and track prediction for data association to solve the problem of occluded object detection and multi-object tracking in severe occlusion situations. If the correlation matching score reaches the threshold, the tracking task is completed; otherwise, to deal with the reappearance of objects with complete occlusion, and the tracks recovery when the object is completely occluded again, the historical trajectory information is retained for continue tracking, and predict frames are as hypothetical frames for continuous tracking. Algorithm 1 presents how to use spatio-temporal features and track prediction to solve the problem of multi-object tracking in severe occlusion situations.

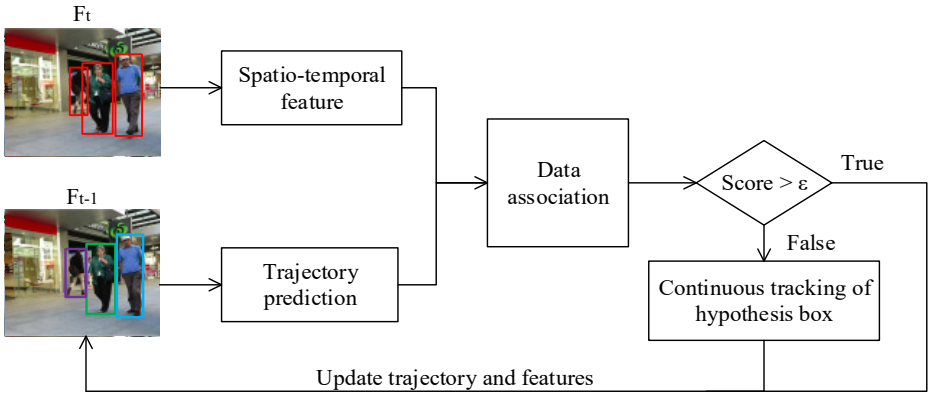


Figure 1. Method flow chart of DetTrack.

The pseudo-code of DetTrack is shown in Algorithm 1. The inputs of DetTrack includes a video sequence V, an object detector Det, the Kalman filter KF, τ_{high}, τ_{low}, ε and σ. τ_{high} is the high detection score thresholds, τ_{low} is the low detection score thresholds, ε is the tracking score threshold and σ is the retention frame threshold. The output of DetTrack is the tracks T of video, where each track contains the bounding box and identities of the object in each frame.

For each frame in the video, we predict the detection boxes and scores by using the detector Det. According to the detection score thresholds τ_{high} and τ_{low}, we separate all the detection boxes into two parts: D_{high} and D_{low}. We use the Kalman filter KF to predict the new locations of each track in T (line 3 to 16 in Algorithm 1).

The first association is performed between the high score detection boxes D_{high} and all tracks T. Association is mainly calculated through appearance feature matching and IOU similarity. The unmatched detections are put into D_{remain} and the unmatched tracks are put into T_{remain} (line 17 to 21 in Algorithm 1). The second association is performed between the low score detection boxes D_{low} and the remaining tracks T_{remain} after the first association. Since low score detection boxes often contain severe occlusion or motion blur, and the appearance features are unreliable, the second association matching is mainly performed through the spatial position relationship between the object and the surrounding environment (line 22 to 25 in Algorithm 1).

For the unmatched tracks $T_{re-remain}$ after the second association, we use the prediction box generated by Kalman filtering as a hypothesis detection box to continue tracking them. If the object still does not appear after continuous tracking of σ frames, we delete it from the tracks T (line 26 to 32 in Algorithm 1). Finally, we initialize new tracks from the unmatched high score detection boxes (line 33 to 38 in Algorithm 1).

Algorithm 1: DetTrack

Input: A video sequence V; object detector Det; Kalman Filter KF; high score detection threshold τ_{high}; low score detection threshold τ_{low}; tracking score threshold ε; retention frame threshold σ

Output: Tracks T of the video

1 Initialization: $T \leftarrow \emptyset$
2 **for** *frame* F_k in V **do**
/* predict detection boxes & scores, record appearance features and spatial positional relationships */
3 $D_k \leftarrow \mathrm{Det}(F_k)$
4 $D_{high} \leftarrow \emptyset$
5 $D_{low} \leftarrow \emptyset$
6 **for** d in D_k **do**
7 **if** $d.score > \tau_{high}$ **then**
8 $D_{high} \leftarrow D_{high} \cup \{d\}$
9 **end**
10 **else if** $d.score > \tau_{low}$ **then**
11 $D_{low} \leftarrow D_{low} \cup \{d\} \cup \{d\}\ D_{hypoth}$
12 **end**
13 **End**

/* Using Kalman filter KF to predict the current frame position of each track in T */
14 **for** t in T **do**
15 $t \leftarrow \mathrm{KF}(t)$
16 **End**

/* High score detection boxes are associated with trajectories, **focusing on improving the speed of multi object tracking** */
/* For non-occlusion or local occlusion, association mainly using motion prediction and appearance features */
17 $\mathrm{A}(T, D_{high}) \leftarrow$ Associate T and D_{high} using feature matching and IoU distance
18 **if** $\mathrm{A}(T, D_{high}).score < \varepsilon$ **then**
19 $D_{remain} \leftarrow$ remaining object boxes from D_{high}
20 $T_{remain} \leftarrow$ remaining tracks from T
21 **End**

/* Low score detection boxes are associated with trajectories, **focusing on solving serious occlusion problems** */
/* For severe occlusion, mainly through the spatial position relationship between the object and the surrounding environment */
22 $\mathrm{A}(T, D_{low}) \leftarrow$ Associate T_{remain} and D_{low} using spatio-temporal features matching
23 **if** $\mathrm{A}(T, D_{low}).score < \varepsilon$ **then**
24 $T_{re-remain} \leftarrow$ remaining tracks from T_{remain}
25 **End**

/* Unmatched tracks using **hypothesis box continuous tracking mechanism** to continue tracking σ frame */
/* Kalman filter prediction box as the current frame object hypothesis detection box */
26 **if** $D_{hypoth}.count < \sigma$ **then**
27 $D_{hypoth} \leftarrow \mathrm{KF}(T_{re-remain})$
28 $D_{hypoth}.count \leftarrow D_{hypoth}.count + 1$

Algorithm 1: ***Cont.***

```
    End

    /* Continuous tracking σ Frame, Tre−remain still not present will be deleted */
    else if D_hypoth.count > σ then
        T ← T \ Tre−remain
    End

    /*Initialize high score unmatched detection as a new track*/
    for d in D_remain do
        if d.score > ε then
            T ← T ∪ {d}
        end
    end
end
Return: T
```

3.1. Tracking Lifting Occlusion Object Detection

Since the Kalman filter has the motion prediction ability in linear systems, the tracks of moving objects are predicted by using the Kalman filter to obtain the current prediction frame of the object. Like [7], we use the Kalman filter to predict the location of the tracks in the new frame. We combine spatio-temporal features with track prediction to restore low-level object detection, and then to distinguish whether the low-level detection frame is a tracking object or a background. This step will not affect the high score detection frame, but only to improve the upper limit of object detection, restore low score detection objects which is mistakenly considered as background, and improve the comprehensive performance of multi-object tracking. The flow chart for low score detection is illustrated in Figure 2.

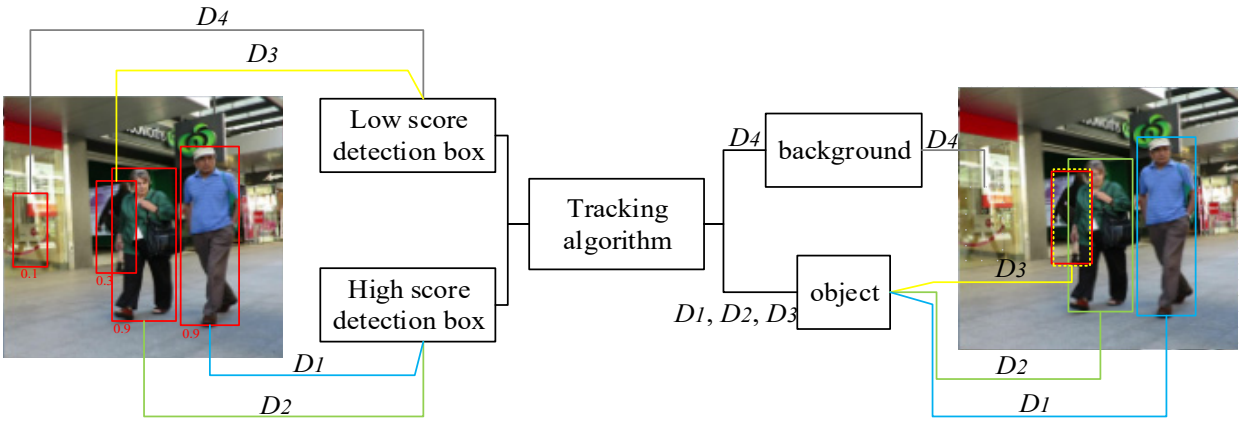


Figure 2. Tracking algorithm low score detection flow chart.

The low score detection box is either the background (D_4 in Figure 2) or the object (D_3 in Figure 2). If the low score detection box is mistaken for the background, the object will be lost that will affect the track integrity of the object, and even affect the subsequent identification (D_4 in Figure 2). Therefore, we distinguish the low score detection frame by using a tracking algorithm. If the historical track is successfully matched with the low score object detection frame, the low score detection frame will be restored as the determined object (D_3 in Figure 2). If the low score detection frame does not match the historical track, the low score detection frame is considered as the background and the object detection is discarded (D_4 in Figure 2). In this step, the matching in the tracking algorithm is still applied to the high score detection box (D_1, D_2 in Figure 2), the upper limit of object detection is improved, the low score detection objects which are mistakenly considered as

background are restored, and the comprehensive performance of multi-object tracking is also improved.

For each input video frame $I_t \in \mathbb{R}^{H\times W\times 3}$, the object detector performs all object detection for the current frame $\mathcal{D}_t = \left\{d_t^i\right\}_{i=1}^{N}$ where $d_t^i = (x_t^i, y_t^i, w_t^i, h_t^i)$. The existing tracks are represented by $\mathcal{T}_{t-1} = \left\{T_{t-1}^j\right\}_{j=1}^{M}$ where $T_{t-1}^j = \left\{d_1^j, \ldots, d_{t-2}^j, d_{t-1}^j\right\}$. The affinity matrix $A \in \mathbb{R}^{N\times M}$ is then estimated by pairwise comparison of the detection and the existing tracks. The formula considers the appearance features and location information.

$$A_{ij} = \mathrm{dist}\left(f\left(d_t^i\right), \hat{f}\left(T_{t-1}^j\right)\right) + \alpha \mathrm{IoU}\left(d_t^i, \hat{d}_t^j\right), \tag{1}$$

where $f\left(d_t^i\right)$ is the appearance feature detected in the current frame, $\hat{f}\left(T_{t-1}^j\right)$ is the discriminative feature of historical tracks, d_t^i is the current frame detection box, $\hat{d}_t^j$ is the current frame prediction box and α is a proportional coefficient. By matching cosine distance with appearance features in Hungary, IoU matches position information.

3.2. Spatio-Temporal Feature Extraction Based on Local Correlation

To improve the accuracy of data association, we use a spatial correlation layer to record the relationship between the object and the relative spatial location. For solving the problem of non-effective tracking caused by independent frame input in traditional MOT, by inputting multiple adjacent frames and object saliency feature frames, the approximate correlation between the object and the entire global context is obtained, while maintaining compactness and efficiency. Simultaneously, we consider the spatial position relationship between the object and the surrounding environment to deal with the tracking of the object in severe occlusion situations The network structure of spatio-temporal feature extraction based on local correlation is shown in Figure 3.

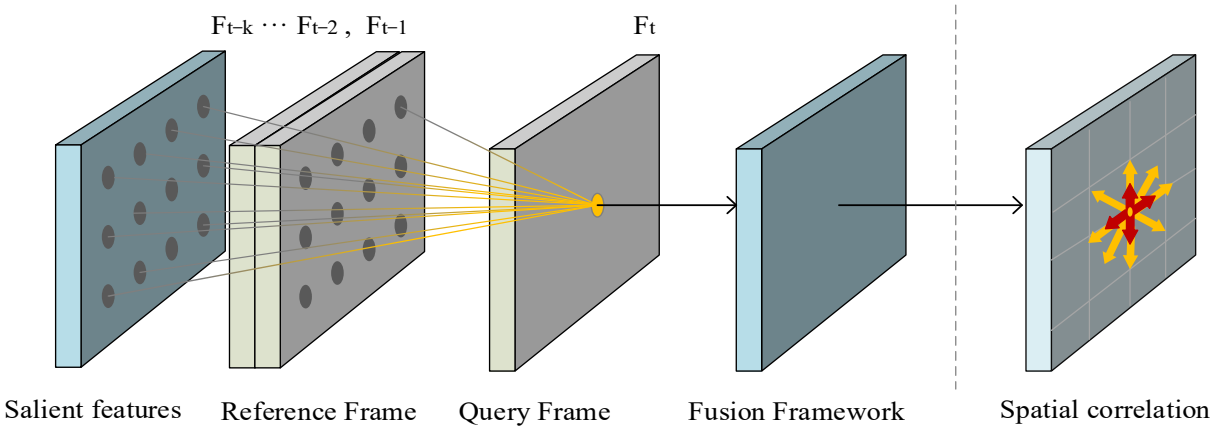


Figure 3. Structure of spatio-temporal local correlation network.

In the spatial local correlation module, the spatial local correlation layer is used to model the relationship structure to associate an object with its surrounding "neighbors". In addition, relevance learning is not limited to the objects of interest. Backgrounds such as vehicles will also be modeled to facilitate object recognition and relational reasoning, establish the corresponding relationship between each spatial location and its environment, and explicitly constrain the relevant volume through self-monitoring learning. In the local correlation layer, feature similarity is only evaluated near the object image coordinates. To obtain long range spatial correlation as much as possible, local correlation operation is performed on the feature pyramid. On the other hand, from the point of view of set

matching, we also need to consider the global characteristics. l represents the level in the feature pyramid. The related volume C^l between the query feature $F_q^l \in \mathbb{R}^{H_l \times W_l \times d_l}$ and the reference feature $F_r^l \in \mathbb{R}^{H_l \times W_l \times d_l}$ is defined as follows:

$$C^l(F_q, F_r, x, d) = F_q^l\Big(x\Big)^T F_r^l(x + d), \| d \|_\infty \leq R, \tag{2}$$

where $x \in \mathbb{Z}^2$ is a coordinate in the query feature map, and $\| d \|_\infty \leq R$ is the displacement from this position. The displacement is limited to R, that is, the maximum movement in any direction is C^l, and the obtained size of the three-dimensional correlation volume is $H^l \times W^l \times (2R+1)^2$.

Time series information is usually ignored in multi-object tracking. But the detector with single frame input is hard to be consistent in time sequence, such that the performance of the tracker degraded when encountering occlusion, blurred scene and small object. Therefore, the spatial local correlation module is extended to the time dimension, and the correlation of goals in different frameworks is established. The correlation learning between two frames can be regarded as another form of motion information learning. This correlation is used to enhance the feature representation and improve the detection accuracy. Specifically, multi-scale correlation learning between different frames can help the tracker to overcome the problems of object occlusion and motion blur and enhance the consistency of detection and recognition features.

3.3. The Hypothesis Box Continuous Tracking Mechanism

A hypothesis box continuous tracking mechanism is proposed to address the issues such as object loss, tracks fragmentation, and ID identity information jumping when conducting multi-object tracking under complete occlusion. That is, first using the prediction box that determines the historical tracks as the assumption box for the current frame of the object for continuous tracking δ Frame (after experimental testing, δ Set to 50 frames). When the object reappears in the field of view during this period, the ID remains unchanged, and the assumption is considered correct. Through tracks repairing, the complete tracking object tracks can be depicted. If the object does not appear again in the frame, the assumed tracks and assumption box are discarded, and the feature information of the object is also discarded to ensure the tracking system's running speed. Due to the hypothesis box continuous tracking mechanism, when the object is completely occluded and then severely occluded, the hypothesis box from the previous frame can be modeled as an undirected graph with other object positions, and matched with the current frame detection undirected graph to achieve continuous tracking when the object starts to appear again and there are no obvious features in severe occlusion.

As shown by the yellow box in Figure 4, frame F_{t-2} is a case of local occlusion, while frame F_{t-1} is a case of severe occlusion. In the F_{t-1} frame, the red solid box represents the object motion prediction box, and the yellow dashed box represents our hypothesis box. From F_t frame to F_{t+2} frame, the continuous tracking mechanism for a hypothesis box is required. The prediction box is used as the hypothesis object box for continuous tracking until the object reappears in F_{t+3} frame, which can retain the object identity and continue tracking, and then repair the complete tracks.



Figure 4. The hypothesis box continuous tracking mechanism.

4. Experimental Results

MOTA (MOT Accuracy) and IDF1 (IDF1 value) are the main indicators to evaluate the tracking performance algorithm. We will conduct ablation experiments on the MOT17 dataset and conduct comparative experiments with advanced tracking algorithms on the MOT16, MOT17 and MOT20 datasets.

4.1. MOT Challenge

MOT challenge is the most common dataset used in multi-object tracking, and is specially used for pedestrian tracking scenarios. Each dataset includes training data and testing data. Training data provides object detection results and real tracking results, testing data only provides object detection results. The newest MOT challenge datasets includes MOT15 [19], MOT16/MOT17 [20], MOT20 [21] and other data.

We use MOT dataset standard evaluation indicators [22] to evaluate different aspects of tracking performance. The indictors are as follows: MOTA (MOT Accuracy), IDF1 (IDF1 value) [23], MT (Mostly Tracked), ML (Mostly Lost), FP (False Positive), FN (FALSE Negative) and IDS (ID Switches). MOTA is calculated based on FP, FN and IDS. FN denotes missing detection which indicates that the GT box is not detected. FP represents false detection which indicates that the detected box is not in the GT. IDS is the number of times that ID has changed. IDS value is larger, the tracked track is more deviated from GT. MT represents the proportion of tracks satisfied the requirement that at least 80% tracking objects are successfully matched by GT. ML indicates the proportion of tracks satisfied the requirement that at least 20% tracking objects are successfully matched by GT. IDF1 evaluates the ability of identity preservation.

4.2. Experimental Details

To improve occlusion object detection by using our multi-object tracking algorithm, the high score detection threshold is set to 0.5, and the low score detection threshold is 0.3, and the historical confirmation track information retention frame $\sigma = 50$. When the track in the history confirmation status has not been successfully matched by the detection, the track will be discarded.

In the experimental training, YOLOX is used as the backbone network, and the pre-trained model parameters on the COCO dataset are used to initialize our model. The size of the input image is adjusted to 1088 by using standard data enhancement techniques, including rotation, scaling, and color 1088 $\times$ 608, the resolution of the feature map is

272 × 152. On two RTX2080Ti graphics processors, 30 rounds of training were conducted, and the training steps spent about 30 h. The computer operating system is Ubuntu 18.04, and the central processing unit (CPU) is Intel core i9-10900K. Building a running environment with a PyTorch version number of 1.7.0, a Torch vision version number of 0.8.0, and a CUDA version number of 10.2.

4.2.1. Selection of Detection Threshold

In the experiment of selecting high score detection threshold values, first, we temporarily set the low score detection threshold to 0.3. When the score of the detection box is less than 0.3, the content of the detection box is background, and the detection box should be discarded. Then, under the condition of fixed low score detection threshold, the tracking results with high score detection threshold ranging from 0.4 to 0.9 were tested separately. When the score of the detection box is greater than the low score detection threshold (the lowest score of the low score detection box) and less than the high score detection threshold (the lowest score of the high score detection box), it is considered as a low score detection box. When the score of the detection box is greater than the high score detection threshold, it is considered as a high score detection box. In the evaluation indicators of multi-object tracking algorithms, MOTA and IDF1 indicators are paid more attention.

Figure 5a shows a line chart for selecting high score detection thresholds under the same experimental environment, where the horizontal axis represents the high score detection threshold from 0.4 to 0.9, and the vertical axis represents the percentage of evaluation index scores. The blue line and the red line represent the results of the tracking algorithm experiments by using the MOTA evaluation index and the IDF1 evaluation index, respectively. From the graph, when the high score detection threshold is set to 0.5, MOTA reaches 74.5% and IDF1 reaches 73.8%, achieving the best tracking performance compared to other threshold settings. Therefore, the high score detection threshold for this experiment is set to 0.5. Figure 5b shows the performance analysis experiment with a low score detection threshold of 0.2. When the high score detection threshold is 0.5, the performance is optimal, with MOTA reaching 73% and IDF1 reaching 70.9%. To be more rigorous, comparative experiments were also conducted in the low score detection threshold experiment, where the low score threshold was, respectively, set to 0.1 and 0.4, but the tracking performance IDF1 results were not ideal. Therefore, the optimal tracking performance threshold is selected that 0.5 for high score detection and 0.3 for low score detection.

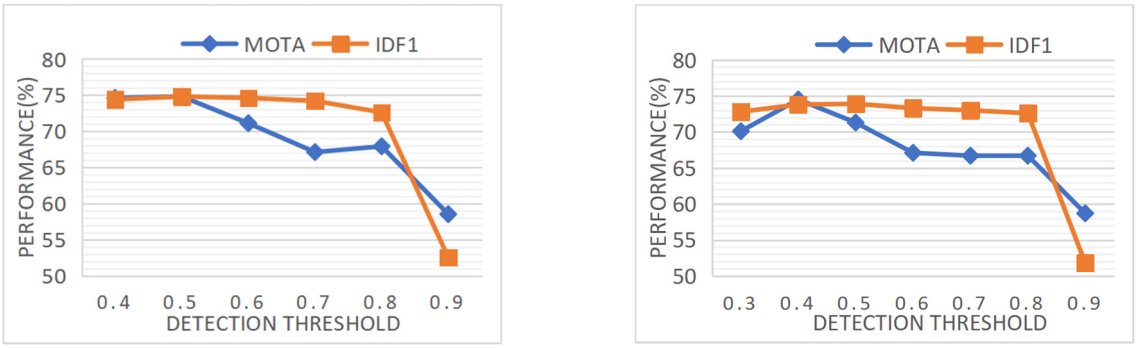


Figure 5. Detection threshold selection experiment. (**a**) Performance analysis experiment with a low score detection threshold of 0.3. (**b**) Performance analysis experiment with a low score detection threshold of 0.2.

4.2.2. Association Threshold Selection

This section presents the experiment of selecting threshold for data association matching. In the tracking algorithm, if the score of data association matching is greater than the threshold, the matching is considered successful. Figure 6a shows the first data association

threshold selection experiment. The horizontal axis represents the first data association matching threshold, and the vertical axis represents the percentage of tracking performance. This article mainly focuses on the results of MOTA and IDF1 for tracking performance, so the blue line represents the evaluation indicator MOTA, and the red line represents the evaluation indicator IDF1. From Figure 6a, under the same execution, MOTA achieves the optimal tracking accuracy, which far exceeds other thresholds, when the first data association matching threshold is selected as 0.6. Therefore, in this article, the data association matching threshold was selected as 0.6.

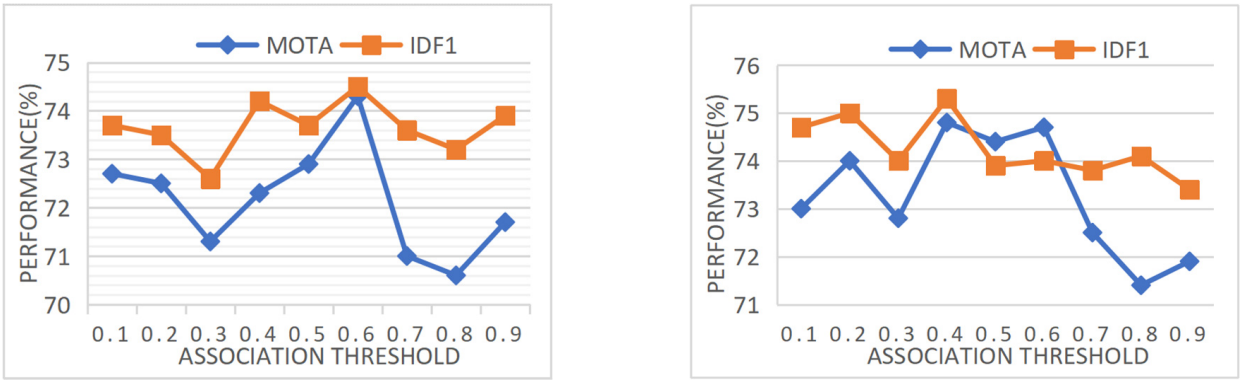


Figure 6. Data association threshold selection experiment. (**a**) Performance analysis experiment on the first data association threshold selection. (**b**) Performance analysis experiment on the second data association threshold selection.

Figure 6b shows the second data association threshold selection experiment. The horizontal axis represents the matching threshold for the second data association, and the vertical axis represents the percentage of tracking performance. This article mainly focuses on the results of MOTA and IDF1 for tracking performance, so the blue line represents the evaluation indicator MOTA, and the red line represents the evaluation indicator IDF1. Under the same execution, when the second data association matching threshold is selected as 0.4, MOTA achieves the optimal tracking accuracy, which far exceeds other thresholds. Meanwhile, the evaluation indicator IDF1 reaches a high percentage. Therefore, the second data association matching threshold was selected as 0.4.

4.3. The Ablation Experiment

We conducted the ablation experiment on the MOT17 dataset for the components proposed in our method. We eliminate each component: (1) the tracking algorithm restores low score detection (Tdec); (2) the spatio-temporal feature module (STC); (3) the hypothesis box continuous tracking mechanism (Hypoth).

As shown in Table 1, "↑" indicates that the larger the indicator value, the better performance for the tracking; "↓" indicates that the smaller indicator value, the better performance for the tracking. Compared with the classical multi-object tracking algorithm DeepSORT as the baseline (line 1), the overall multi-object tracking algorithm has improved 5.4% in MOTA indicators and 15.0% in IDF1 indicators. The tracking algorithm recovery low score detection (Tdec) module has improved MOTA by 2.5% and IDF1 by 4.0%. The space-time feature module (STC) mainly improves MOTA by 3.9% and IDF1 by 13.3%. The results show that (1) the tracking algorithm recovers low score detection (Tdec), (2) the spatio-temporal feature module (STC), and (3) the hypothesis box continuous tracking mechanism (Hypoth) are effective.

Table 1. Ablation studies on the MOT17 validation set.

Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	IDS ↓
Baseline (DeepSort)	69.6	61.2	129	59	595
Baseline + TDec	72.1	65.2	155	48	783
Baseline + STC	73.5	74.5	157	44	495
Baseline + TDec + STC	75.0	76.2	170	40	429
Baseline + TDec + STC + Hypoth	76.3	76.8	187	37	301

4.4. Comparison with State-of-the-Art Methods

We compared our tracking algorithm DetTrack with other advanced methods on the MOT16, MOT17 and MOT20 test sets. From the perspective of processing methods, multi-object tracking is mainly divided into online tracking and offline tracking. The main difference between online and offline tracking is whether information from the future frames is used to improve tracking performance. Since offline tracking requires the use of future frame information and can not achieve real-time object tracking, it can only be applied to pre-recorded video sequences. Our proposed algorithm DetTrack belongs to the online multi-object tracking algorithm and can be applied to the field of video surveillance to achieve real-time multi-object tracking. We compare DetTrack with other popular association methods including JDE [24], TraDes [25], FairMOT [26], MOTR [27], CenterTrack [28] and QuasiDense [29]. Among these trackers, JDE, FairMOT, and TraDes use a combination of motion and ReID similarity. QuasiDense uses Re-ID separatly. CenterTrack and TraDes predict the motion similarity by the learning networks. MOTR use the attention mechanism to propagate boxes among frames. We also compared DetTrack with other popular algorithms on each dataset. Since datasets have different birth times, many advanced algorithms have only been experimented on a few datasets. All results are directly obtained from the official MOT challenge evaluation server.

Table 2 shows the evaluation results of the MOT16 dataset and compares them with other advanced multi-object tracking algorithms. The best performance indicators have been displayed in bold. Our DetTrack achieves a tracking accuracy of 78.0% on MOTA indicators. The IDFI index reached 78.3%. This represents that the algorithm in this paper has achieved good results in object detection and object correlation performance. MT hits the pointer much higher than other tracking algorithms, which means that the track successfully matched in at least 80% of the time accounts for a higher proportion of all tracked objects. FN index performance is better than other tracking short shots, which means that our algorithm can achieve more object truth detection and reduce missed detection. If our multi-object tracking algorithm is further studied in the feature extraction part in the future, our tracking performance will be further improved.

The experimental results on the MOT17 dataset compared with other advanced algorithms are shown in Table 3. The best performance indicators have been displayed in bold. Our method DetTrack achieves the MOTA index of 78.0% on MOT17 and the IDF1 index of 78.3%. Compared with the FairMOT algorithm, the IDF1 index is improved by 5.7%, and the MOTA index is improved by 5.3%. Compared with other advanced algorithms such as TraDes, MOTR and LMOT, MOTA, IDF1, MT and FN, our algorithm has achieved better performance.

Compared with MOT17, MOT20 is more densely populated. In the MOT20 test set, 170 pedestrians appeared simultaneously in one frame. The best performance indicators have been displayed in bold. The experimental results of comparison with other advanced algorithms on the MOT20 dataset are shown in Table 4. Our algorithm DetTrack achieves an MOTA index of 72.8% and an IDF1 index of 73.4 in the MOT20 dataset. Compared with other advanced algorithms, it achieves better performance in MOTA, IDF1 and MT indicators. The DetTrack algorithm in this paper outperforms other multi-object trackers in IDF1 and ML indicators, and maintains comparability in ML, FP, FN and IDS indicators, which further shows that the DetTrack algorithm we proposed to solve occlusion problems is very effective.

Table 2. Tracking results on the MOT16 test set.

Method	Mode	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Hz ↑
Lif_T [30]	Offline	61.3	64.7	27.0	34.0	4844	65,401	389	0.5
TraDes [25]	Offline	70.1	64.7	37.3	20.0	-	-	1144	-
GSDT [31]	Offline	74.5	68.1	41.2	17.3	-	-	1299	-
Trackor [32]	Online	56.2	54.9	20.7	35.8	**2394**	76,844	617	1.8
SORT [7]	Online	59.8	53.8	25.4	22.7	8698	63,245	1424	**60**
DeepSORT [8]	Online	61.4	62.2	32.8	18.2	12,852	56,668	781	40
JDE [24]	Online	64.4	55.8	35.4	20.0	-	-	1544	18.5
Tube_TK [33]	Online	66.9	62.2	39.0	16.1	11,544	47,502	1236	1.0
CTracker [34]	Online	67.6	57.2	32.9	23.1	8934	48,305	1897	6.8
LMOT [35]	Online	73.2	72.3	44.0	17.0	-	-	669	-
FairMOT [36]	Online	74.9	72.8	44.7	15.9	10,163	34,484	1074	25.4
DetTrack (ours)	Online	**78.0**	**78.3**	**54.7**	14.1	12,981	**25,660**	1538	22.3

Table 3. Tracking results on the MOT17 test set.

Method	Mode	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Hz ↑
LPT [37]	Offline	57.3	57.7	23.3	36.9	15,187	224,560	1424	-
Lif_T [30]	Offline	60.5	65.6	27.0	33.6	**14,966**	206,619	**1189**	0.5
QuasiDense [29]	Offline	68.7	66.3	40.6	21.9	26,589	146,643	3378	-
TraDes [25]	Offline	69.1	63.9	36.4	21.5	20,892	150,060	3555	-
GSDT [31]	Offline	73.2	66.5	41.7	17.5	-	-	3891	-
Tube_TK [33]	Online	63.0	58.6	31.2	19.9	27,060	177,483	4137	3.0
MOTR [27]	Online	65.1	66.4	33.0	25.2	45,486	149,307	2049	-
CTracker [34]	Online	66.6	57.4	32.2	24.2	22,284	160,491	5529	34.4
CenterTrack [28]	Online	67.8	64.7	34.6	25.6	18,498	160,332	3039	3.8
Tracktor [32]	Online	56.3	55.1	21.1	35.3	8866	235,449	1987	1.8
MAT [38]	Online	69.5	63.1	-	-	30,660	138,741	2844	9.0
SOTMOT [39]	Online	71.0	71.9	42.7	15.3	39,537	118,983	5184	16.0
LMOT [35]	Online	72.0	70.3	45.4	17.3	-	-	3071	-
FairMOT [36]	Online	73.7	72.3	43.2	17.3	27,507	117,477	3303	25.9
TrackFormer [40]	Online	74.1	68.0	47.3	**10.4**	-	-	2829	-
DetTrack (ours)	Online	**78.0**	**77.6**	**54.2**	14.6	33,835	**85,660**	4878	**35.5**

Table 4. Tracking results on the MOT20 test set.

Method	Mode	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Hz ↑
LPT [37]	Offline	57.9	53.5	39.0	22.8	9980	205,949	1827	-
SORT [7]	Online	42.7	45.1	16.7	26.2	27,521	264,694	4470	-
Tracktor [32]	Online	52.6	52.7	29.4	26.7	**6930**	236,680	1648	-
ArTist [41]	Online	53.6	51.0	31.6	28.1	7765	230,576	1531	-
LMOT [35]	Online	59.1	61.1	25.1	23.0	-	-	**1398**	-
FairMOT [36]	Online	61.8	67.3	68.8	**7.6**	103,440	88,901	5243	13.2
DetTrack (ours)	Online	**75.6**	**75.3**	**70.4**	9.3	37,706	**86,692**	1640	**16.9**

4.5. Visualization Results

All visualization results are from the MOT challenge official website. Figure 7 shows the visualization results of our tracking algorithm on the MOT17 test set. We have bolded the borders for the objects that we are more concerned about, and the video frame rate is displayed with white numbers in the upper left corner. In the screenshot of frame 143 of the MOT17-08 dataset, the historical track of the object remains unchanged under severe occlusion. In frame 226, the object can be tracked as quickly as possible when severe occlusion occurs that verifies the effectiveness of our algorithm. In the 188th frame of the MOT17-12 dataset, object detection can still be recovered under severe occlusion that shows the effectiveness of our tracking algorithm to recover low score detection (Tdec).

It can be found from the object with track ID 9 in the MOT17-03 dataset that when it is frequently occluded or even completely occluded in a large area, our method can detect slightly changed objects and keep the original track ID unchanged. In other visualization results, we can find that our method can not only adapt to local occlusion, but also adapt to occlusion for multi-object tracking.



Figure 7. Example trace results on the MOT17 test set.

Figure 8 shows the visualization results obtained by our algorithm in tracking the MOT20 test set in the official MOT. We have bolded the borders for the objects that we are more concerned about, and the video frame rate is displayed with white numbers in the upper left corner. The object with ID 112 in the MOT20-07 dataset experienced local occlusion, severe occlusion, and complete occlusion before reappearing. The algorithm in this paper can continuously track the object and successfully detect and maintain the ID unchanged in severe occlusion situations. In other visualization results, it can be found that the method proposed in this paper can not only adapt to local occlusion situations, but also adapt to severe occlusion, complete occlusion, and long-term occlusion situations for object tracking, and restore severely occluded object detection while maintaining the identity of the object trajectory unchanged.



Figure 8. Example trace results on the MOT20 test set.

5. Conclusions

In this work, we propose a method to simultaneously improve occluded object detection and occluded object tracking, and propose a tracking method when the object is completely occluded. First, motion track prediction is utilized to improve the upper limit of occluded object detection. Then, multi-object tracking is performed based on the spatio-temporal feature information of the object. Since unable to extract appearance features for severe occlusion of the object, we mainly utilize the spatial position relationship between the object and the surrounding environment to solve the problem of multi-object tracking in severe occlusion situations. Finally, we use the hypothesis frame to continuously track the completely occluded object. The experiments of ablation study and comparison with other state-of-the-art methods both demonstrate the effectiveness of our method. Assuming that the box continuous tracking mechanism can be ported to other multi-object tracking algorithms to solve complete occlusion situations, then the first data association can be replaced with another multi-object tracking algorithm. Since the success of this tracker mainly depends on the quality of input detections, so we plan to explore training object detection jointly with our network flow framework in the future. We also plan to utilize the spatial position relationship between the object and the surrounding environment, and use adjacent object spatial position relationships for graph matching for severely occluded objects to further improve tracking performance.

Author Contributions: Conceptualization, X.G., Z.W. and S.Z. (Shanna Zhuang); methodology, X.G. and Z.W.; software, X.G.; validation, Z.W., X.W. and S.Z. (Shuo Zhang); formal analysis, Z.W. and H.W.; investigation, X.G. and S.Z. (Shuo Zhang); resources, X.W.; data curation, S.Z. (Shanna Zhuang); writing—original draft preparation, X.G.; writing—review and editing, Z.W. and X.W.; visualization, X.G. and S.Z. (Shuo Zhang); supervision, S.Z. (Shanna Zhuang) and H.W.; project administration, Z.W. and H.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Nature Science Foundation of China (No. 61972267) and the Nature Science Foundation of Hebei Province (No. F2019210306).

Data Availability Statement: The MOT16 dataset is available at https://motchallenge.net/data/MOT16 (accessed on 11 November 2023). The MOT17 dataset is available at https://motchallenge.net/data/MOT17 (accessed on 11 November 2023). The MOT20 dataset is available at https://motchallenge.net/data/MOT20 (accessed on 11 November 2023). The data that support the findings of this study are available from https://motchallenge.net/method/MOT=5905&chl=5 (accessed on 11 November 2023).

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

References

1. Mondal, A. Neuro-probabilistic model for object tracking. *Pattern Anal. Appl.* **2019**, *22*, 1609–1628. [CrossRef]
2. Wang, C.; Zhang, J.; Li, M. Object Tracking Algorithm Combining Convolutional Transformer. *Comput. Eng.* **2023**, *49*, 281–288.
3. Sui, Y.; Wang, G.; Zhang, L. Correlation filter learning toward peak strength for visual tracking. *IEEE Trans. Cybern.* **2017**, *48*, 1290–1303. [CrossRef] [PubMed]
4. Fazli, S.; Pour, H.M.; Bouzari, H. Particle filter based object tracking with sift and color feature. In Proceedings of the International Conference on Machine Vision, Dubai, United Arab Emirates, 28–30 December 2009.
5. Roccetti, M.; Gustavo, M.; Angelo, S. Playing into the wild: A gesture-based interface for gaming in public spaces. *J. Vis. Commun. Image Represent.* **2012**, *23*, 426–440. [CrossRef]
6. Liu, F.; Huang, G.; Lu, L. Robust Object Tracking Algorithm with Adaptive Template Update. *Comput. Sci. Explor.* **2019**, *13*, 83–96.
7. Bewley, A.; Ge, Z.; Ott, L. Simple online and realtime tracking. In Proceedings of the International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016.
8. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.
9. Yin, J.; Wang, W.; Meng, Q. A unified object motion and affinity model for online multi-object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
10. Wang, X.; Chen, Z.; Tang, J. Dynamic attention guided multi-trajectory analysis for single object tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4895–4908. [CrossRef]
11. Hou, J.; Yu, N.; Xiang, J. Multi object tracking based on motion model and data association. *J. South-Cent. Univ. Natl. Nat. Sci. Ed.* **2023**, *42*, 260–266.
12. Carballo, J.A.; Bonilla, J.; Fernández-Reche, J. Cloud Detection and Tracking Based on Object Detection with Convolutional Neural Networks. *Algorithms* **2023**, *16*, 487. [CrossRef]
13. Yang, J.; Pan, Z.; Liu, Y. Single Object Tracking in Satellite Videos Based on Feature Enhancement and Multi-Level Matching Strategy. *Remote Sens.* **2023**, *15*, 4351. [CrossRef]
14. Chen, J.; Sheng, H.; Zhang, Y.; Xiong, Z. Enhancing detection model for multiple hypothesis tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017.
15. Xiao, X.; Feng, X. Multi-Object Pedestrian Tracking Using Improved YOLOv8 and OC-SORT. *Sensors* **2023**, *23*, 8439. [CrossRef]
16. Perera, C.J.; Premachandra, C.; Kawanaka, H. Enhancing Feature Detection and Matching in Low-Pixel-Resolution Hyperspectral Images Using 3D Convolution-Based Siamese Networks. *Sensors* **2023**, *23*, 8004. [CrossRef] [PubMed]
17. Shi, L.; Tan, J. Discovery. Recurrence, and Inhibition of Motion-Blur Hysteresis Phenomenon in Visual Tracking Displacement Detection. *Sensors* **2023**, *23*, 8024. [CrossRef] [PubMed]
18. Lee, H.; Cho, H.; Noh, B.; Yeo, H. NAVIBox: Real-Time Vehicle–Pedestrian Risk Prediction System in an Edge Vision Environment. *Electronics* **2023**, *12*, 4311. [CrossRef]
19. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. Motchallenge: Towards a benchmark for multi-object tracking. *arXiv* **2015**, arXiv:1504.01942.
20. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
21. Dendorfer, P.; Osep, A.; Milan, A.; Schindler, K.; Cremers, D.; Reid, I.; Leal-Taixé, L. MOTChallenge: A Benchmark for Single-Camera Multiple object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 845–881. [CrossRef]
22. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP J. Image Video Process.* **2008**, *2008*, 246309. [CrossRef]
23. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-object, multi-camera tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 15–16 October 2016.
24. Sridhar, V.H.; Roche, D.G.; Gingins, S. Tracktor: Image-based automated tracking of animal movement and behaviour. *Methods Ecol. Evol.* **2019**, *10*, 815–820. [CrossRef]
25. Hornakova, A.; Henschel, R.; Rosenhahn, B.; Swoboda, P. Lifted disjoint paths with application in multiple object tracking. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020.
26. 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.

27. Pang, B.; Li, Y.; Zhang, Y.; Li, M.; Lu, C. Tubetk: Adopting tubes to track multi-object in a one-step training model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
28. Peng, J.; Wang, C.; Wan, F.; Wu, Y.; Wang, Y.; Tai, Y.; 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.
29. 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, Online, 19–25 June 2021.
30. Mostafa, R.; Baraka, H.; Bayoumi, A. LMOT: Efficient Light-Weight Detection and Tracking in Crowds. *IEEE Access* **2020**, *10*, 83085–83095. [CrossRef]
31. Wang, Y.; Kitani, K.; Weng, X. Joint object detection and multi-object tracking with graph neural networks. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021.
32. 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]
33. Wang, Q.; Zheng, Y.; Pan, P.; Xu, Y. Multiple object tracking with correlation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021.
34. Li, S.; Kong, Y.; Rezatofighi, H. Learning of Global Objective for Network Flow in Multi-Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
35. Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; Wei, Y. Motr: End-to-end multiple-object tracking with transformer. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October2022.
36. 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.
37. Pang, J.; Qiu, L.; Li, X.; Chen, H.; Li, Q.; Darrell, T.; Yu, F. Quasi-dense similarity learning for multiple object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021.
38. Han, S.; Huang, P.; Wang, H.; Yu, E.; Liu, D.; Pan, X. Mat: Motion-aware multi-object tracking. *Neurocomputing* **2022**, *476*, 75–86. [CrossRef]
39. 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, Online, 19–25 June 2021.
40. Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; Feichtenhofer, C. Trackformer: Multi-object tracking with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
41. Saleh, F.; Aliakbarian, S.; Rezatofighi, H.; Salzmann, M.; Gould, S. Probabilistic tracklet scoring and inpainting for multiple object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021.

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.