
Article:
Li, J., Ding, Y., Wei, H. orcid.org/0000-0002-4704-7346 et al. (2 more authors) (2022) SimpleTrack : rethinking and improving the JDE approach for multi-object tracking. Sensors, 22 (15). 5863.

https://doi.org/10.3390/s22155863

---

sensors

MDPI

Article

SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking

Jiaxin Li [1], Yan Ding [1,*], Hua-Liang Wei [2], Yutong Zhang [1] and Wenxiang Lin [1]

[1] Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education, School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China
[2] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, UK
* Correspondence: dingyan@bit.edu.cn; Tel.: +86-139-1175-6630

Abstract: Joint detection and embedding (JDE) methods usually fuse the target motion information and appearance information as the data association matrix, which could fail when the target is briefly lost or blocked in multi-object tracking (MOT). In this paper, we aim to solve this problem by proposing a novel association matrix, the Embedding and GioU (EG) matrix, which combines the embedding cosine distance and GioU distance of objects. To improve the performance of data association, we develop a simple, effective, bottom-up fusion tracker for re-identity features, named SimpleTrack, and propose a new tracking strategy which can mitigate the loss of detection targets. To show the effectiveness of the proposed method, experiments are carried out using five different state-of-the-art JDE-based methods. The results show that by simply replacing the original association matrix with our EG matrix, we can achieve significant improvements in IDF1, HOTA and IDsw metrics, and increase the tracking speed of these methods by around 20%. In addition, our SimpleTrack has the best data association capability among the JDE-based methods, e.g., 61.6 HOTA and 76.3 IDF1, on the test set of MOT17 with 23 FPS running speed on a single GTX2080Ti GPU.

Keywords: multiple object tracking; association matrix; joint detection and embedding; decoupling representation

Citation: Li, J.; Ding, Y.; Wei, H.-L.; Zhang, Y.; Lin, W. SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking. *Sensors* **2022**, *22*, 5863. https://doi.org/10.3390/s22155863

Academic Editors: Xian Tao, Qingyi Gu and Hu Su

Received: 4 July 2022
Accepted: 2 August 2022
Published: 5 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

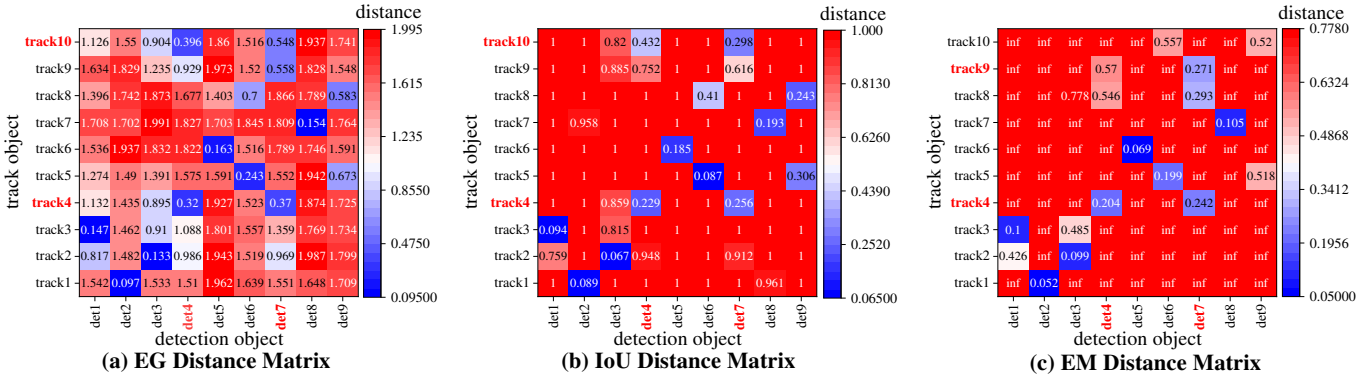
1. Introduction

Multi-object tracking (MOT), aiming to estimate the locations and identity of multiple targets in a video sequence, is a fundamentally challenging task in computer vision [1]. Recently, the Intersection over Union (IoU) and Hungarian method have been commonly used in the tracking phase, among many tracking-by-detection paradigms [2–10]. However, when the target is occluded or lost for a period of time, it is difficult to retrieve the correct identity only using the IoU distance. As a result, the identity switching of targets occurs from time to time. To alleviate this problem, many methods have started to introduce the re-identity feature of targets. Among them, the JDE-based methods [11–17] have become popular due to their simplicity and efficiency.

In part of the data association, the accuracy of similarity measurement determines the tracking performance. Most detection-based methods use the IoU distance as the similarity matrix in the cascade matching strategy, while JDE-based methods fuse the motion information and appearance information as the similarity matrix for the linear assignment in the first matching and use the IoU distance in the next matching. However, none of these existing methods provides the best expression of the similarity matrix according to our experiments.

When objects are occluded due to interlacing, it will produce confusing sets, which are difficult to allocate correctly, e.g., the set {det4, det7, track4, track10} in Figure 1a,b, and the set {det4, det7, track4, track9} in Figure 1c. When assigning these confusing sets, the inaccurate similarity distance leads to tracking failure. Based on the Hungarian method,

the IoU distance matrix tends to match det4 with track4 and det7 with track10, and the EM distance matrix tends to match det4 with track4 and det7 with track9. Both of them lead to target identity switching. The principal reason for these matching failures is the inaccurate prediction from the Kalman filter as the time of target loss becomes longer. Clearly, this results in an inaccurate IoU distance and motion information distance, which leads to the problem of linear allocation errors.



(a) EG Distance Matrix **(b) IoU Distance Matrix** **(c) EM Distance Matrix**

Figure 1. Example of heatmaps for different association matrices in frame 560 of MOT17 sequence 11. (**a**) shows our EG matrix, which combines the embedding cosine distance and the GioU distance. (**b**) shows the IoU distance matrix, i.e., the detection-based methods. (**c**) shows the EM matrix, which usually combines the motion distance and the embedding cosine distance, i.e., the JDE-based methods. In these heatmaps, the red cells indicates that the similarity distance between detection targets and tracking targets is farther, and the blue cells show that the similarity distance is closer.

To solve this problem, we propose the EG matrix, which utilizes the embedding cosine distance for the long-range tracking of targets and the GioU distance for limiting the matching range of embedding. To illustrate the robustness of the EG matrix, we apply it to five different JDE-based methods. As can be seen in Section 4.3, our implementations obtain improvements in MOT metrics, including tracking speed, HOTA, IDF1 and IDsw metrics.

To further explore the good properties of the EG matrix, we propose a simple tracking framework named SimpleTrack. In this framework, we design a bottom-up branch to represent Re-ID features. Different from the fusion method of detection features, it pays more attention to the high-level semantic layers. For the tracking part of SimpleTrack, we propose a novel tracking retrieval mechanism and design a new tracking strategy based on our EG matrix. The experimental results show that our tracking strategy can surpass the JDE-based methods in most metrics, including tracking speed. Compared with the current SOTA method BYTE, our tracking strategy can also improve the performance in terms of HOTA, IDF1 and IDsw metrics.

Our main contributions are as follows:

1. We adopt different feature fusion structures for feature detection and feature re-identification, respectively, to decouple them.

2. We propose a novel association matrix named the embedding and GioU matrix, which can directly replace the original association matrix in JDE-based methods. It can not only reduce time costs, but also improves the tracking metrics of the model.

3. We design a new tracking strategy that can alleviate the problem of tracking target loss.

4. The code and model are available at https://github.com/1vpmaster/SimpleTrack (accessed on 3 July 2022).

The remainder of the paper is arranged as follows. Section 2 summarizes the related work, including JDE-based methods, similarity matrices and tracking strategies. Section 3 describes the method of SimpleTrack, including the decoupling module, embedding and Giou matrix and a novel tracking strategy. In Section 4, experimental results are provided to verify the performance of the proposed SimpleTrack. Section 5 discusses the performance

and speed of the EG matrix and association methods. Section 6 briefly summarizes the work and considers the future work.

2. Related Work

2.1. Joint Detection and Embedding

JDE-based methods typically employ a single network to directly predict detection and appearance features [11–19]. In general, these methods employ a single backbone to predict both object bounding boxes and appearance features. For example, FasterVideo [18] and Online Tracker [19] adopt Faster R-CNN [20] and Yolov5 for feature detection and feature re-identification, respectively. Although their pipelines are relatively simple, the competitive relationship between detection and identification harms the optimization procedure in the multi-task learning of object detection and appearance feature extraction.

Recently, to tackle this problem, CSTrack [13] was proposed, which first uses a decoupling module to enhance the learned representation for both object detection and appearance identification. RelationTrack [21] uses a channel attention mechanism to decouple detection and re-identity. However, the two methods do not take into account the essential differences between detection features and re-identity features. Different from CSTrack and RelationTrack, the decoupling strategy adopted in our SimpleTrack focuses on the essence of the appearance feature. We start decoupling from the feature layer fusion of the network. In contrast to the detection feature fusion, we adopt a bottom-up fusion method.

2.2. Similarity Matrices

Location, motion and appearance are the most common cues in multi-object tracking. They are also combined together for the linear assignment. Detection-based methods [10] utilize the IoU distance as the similarity matrix and the tracking accuracy mainly depends on the detector. SORT [2] fuses position and motion cues as the similarity matrix, which can achieve good results in short-range matching. DeepSORT [7] improves the long-range tracking ability of trackers by merging appearance and motion cues, which is usually used in JDE-based methods [11–17].

All these methods use location cues or fuse appearance and motion information as the similarity matrix, as shown in Figure 2. However, the motion information estimated by linear motion models is not accurate in some scenes containing complex motion behaviors. In addition, it is time-consuming to integrate motion information and appearance information according to Section 5.2.2. Different from all the aforementioned methods, we design the similarity matrix combined with appearance and location information and use the GioU distance matrix as the location cue instead of the common IoU matrix.

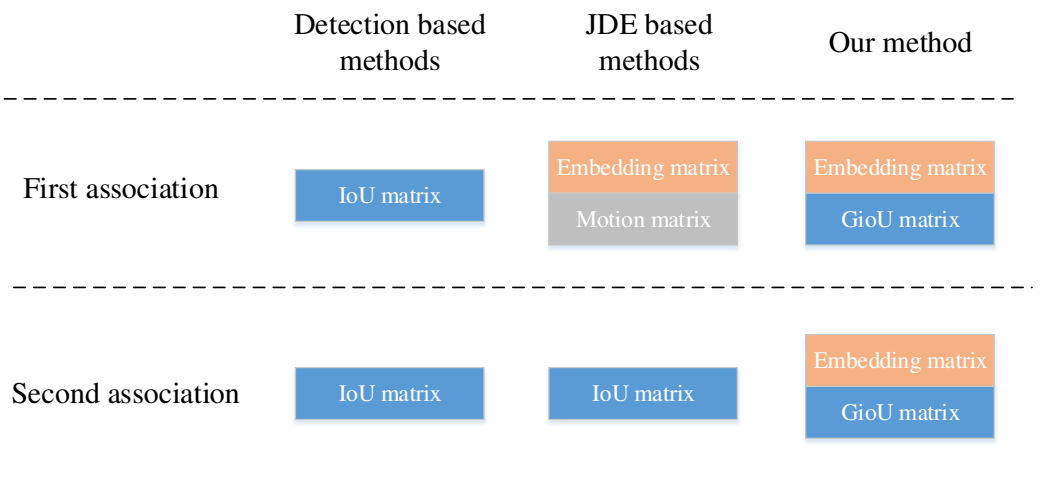


Figure 2. Association matrices used in cascade matching of different tracking methods.

2.3. Tracking Strategy

The assignment problem of target tracking and detection can be solved by the Hungarian algorithm [22] based on different similarity matrices. SORT associates the detection objects with the tracking objects by one-time matching. DeepSORT adopts a cascade matching method that reduces unmatched tracking targets. MOTDT [23] first uses the appearance similarity matrix and the IoU distance matrix as the similarity matrix for cascade matching, respectively. All of these methods assume that the detection targets are equally important and match them uniformly with the similarity matrix.

Recently, BYTETrack [10] proposed to use low-confidence detection results for secondary matching, which reduces the problem of target detection failure due to occlusion. Thereby, the occurrence of long-range tracking could be reduced, making the linear assignment based on the IoU distance matrix more effective. MAA [24] adopts different strategies for the blurred detection of targets and tracking targets in the similarity matrix. The method can alleviate the inaccuracy of the similarity distance caused by the ambiguous targets. Both of the two methods aim to make up for the shortcomings of the similarity matrix and do not pay attention to how to retrieve the lost detection targets. Based on the idea of BYTE [10], we redesign the similarity matrix for the JDE-based method and construct a new matching strategy.

3. SimpleTrack

In this section, we present the technical details of SimpleTrack, as illustrated in Figure 3. It is composed of feature decoupling, a similarity matrix as well as a tracking strategy.

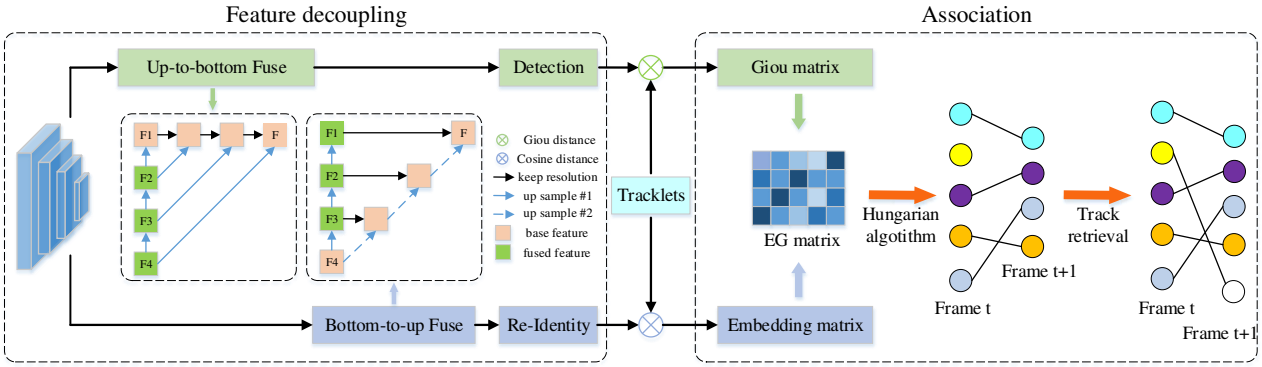


Figure 3. The overall pipeline of SimpleTrack. The input image is first fed to a backbone network to extract high-resolution feature maps. Then, we use different feature fusion methods for detection and re-identity separately, and combine the embedding and GioU distance matrix as the similarity matrix. At the end of the association phase, the tracking retrieval mechanism is used to recover the undetected targets.

3.1. Feature Decoupling

We adopt DLA-34 as a backbone in order to strike a good balance between accuracy and speed. For feature decoupling, we employ different feature fusion methods for detection and Re-ID representation. As illustrated in Figure 3, for the detection branch, the feature fusion method still adopts the structure of IDA-up in FairMOT [12]. We call it the up-to-bottom fusion method, based on low-level feature maps and continuously fusing higher-level feature maps.

However, Re-ID features tend to learn higher-level semantic features to distinguish different features among homogeneous objects. Therefore, we take a simple bottom-up approach to fusing feature maps. Denote the input feature maps by $\mathbf{F} = \{\mathbf{F_i}\}_{i=1}^{N}$, where N is the number of feature layers of different resolutions extracted by the backbone network. Then, the process of the bottom-up fusion method can be expressed as

$$\{\hat{\mathbf{F}}_i\}_{i=N}^{1} = \begin{cases} \mathbf{F}_i, & \text{if } i = N \\ \mathbf{F}_i \cdot \sigma(Conv_{1\times1}(UpSample(\hat{\mathbf{F}}_{i+1}))), & \text{otherwise} \end{cases} \tag{1}$$

where $UpSample(\cdot)$ represents an upsampling operation composed of the deformable convolution and the deconvolution, $Conv_{1\times1}$ denotes a 1×1 convolution layer for changing channels of features, $\sigma(\cdot)$ represents the Sigmoid activation layer.

It could be observed from Equation (1) that the fusion process is from bottom to top, and the previously fused feature map guides the lower-level feature map until the final fusion result is obtained. As will be shown by the experimental results in Section 4, the computational cost required by this fusion method is minimal.

3.2. Embedding and GioU Matrix

The similarity matrix is usually constructed from location, motion and appearance information. Let **L**, **M**, **E** denote the location distance matrix, the motion distance matrix and the appearance distance matrix, respectively. We fuse **L** and **E** as the similarity matrix, called the EG matrix. Moreover, **L** can be represented as

$$\mathbf{L} = \mathbf{1} - \left(\frac{|\ \mathbf{A} \cap \mathbf{B}\ |}{|\ \mathbf{A} \cup \mathbf{B}\ |} - \frac{|\ \mathbf{C} \backslash (\mathbf{A} \cup (\mathbf{B}))\ |}{|\ \mathbf{C}\ |}\right) \tag{2}$$

where **A** and **B** represent the bounding boxes of the tracking objects and the bounding boxes of the detection objects, respectively, and **C** is the minimum enclosing rectangle sets of the above bounding boxes.

E can be represented as

$$\mathbf{E} = \frac{\mathbf{O}_e^1 \cdot \mathbf{O}_e^2}{\|\ \mathbf{O}_e^1\ \| \| \mathbf{O}_e^2\ \|} \tag{3}$$

where $\mathbf{O}_e^1$ and $\mathbf{O}_e^2$ represent different appearance embedding vectors.

Note that the matrix **L** in Equation (2) is actually the GioU distance matrix and that the matrix **E** in Equation (3) defines the cosine distance matrix. Then, the embedding and GioU matrix, which is also denoted as **EG**, can be represented as

$$\mathbf{EG} = \lambda_1 \mathbf{E} + \lambda_2 \mathbf{G} \tag{4}$$

where $\lambda_1 = 1.0$ and $\lambda_2 = 0.5$ represent two hyperparameters, **G** denotes the GioU distance matrix and $\mathbf{G} = \mathbf{L}$.

3.3. Tracking Strategy in SimpleTrack

Inspired by BYTE [10], we develop a tracking strategy based on our EG matrix. As shown in Algorithm 1, we follow the idea of secondary matching with low-confidence detection adopted in BYTE, and use the EG matrix to replace the similarity matrix in the cascade matching. In addition, after the secondary matching, we utilize the cosine distance to retrieve the unmatched tracklets.

Algorithm 1: Pseudo-code of SimpleTrack

Input: A video sequence V; object detector Det; Kalman filter KF; detection score threshold τ_{high}, τ_{low}; tracking score threshold ϵ; tracking retrieval threshold ϵ_r

Output: Tracks $\mathcal{T}$ of the video

for *frame* f_k *in* V **do**
  $\mathcal{D}_k \leftarrow Det(f_k)$;
  $\mathcal{D}_{high} \leftarrow \emptyset$;
  $\mathcal{D}_{low} \leftarrow \emptyset$;
  for d *in* $\mathcal{D}_k$ **do**
    if $d.score > \tau_{high}$ **then**
      $\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}$;
    if $d.score > \tau_{low}$ **then**
      $\mathcal{D}_{low} \leftarrow \mathcal{D}_{low} \cup \{d\}$;
  for t *in* $\mathcal{T}$ **do**
    $t \leftarrow KF(t)$;
  `// first association with EG matrix`
  Associate $\mathcal{T}$ and $\mathcal{D}_{high}$ using EG matrix;
  $\mathcal{D}_{remain} \leftarrow$remaining object boxes from $\mathcal{D}_{high}$;
  $\mathcal{T}_{remain} \leftarrow$remaining tracks from $\mathcal{T}$;
  `// second association with EG matrix`
  Associate $\mathcal{T}$ and $\mathcal{D}_{low}$ using EG matrix;
  $\mathcal{T}_{re-remain} \leftarrow$remaining tracks from $\mathcal{T}$;
  $\mathcal{T}_u = \mathcal{T} - \mathcal{T}_{remain} - \mathcal{T}_{re-remain}$;
  `// tracking retrieval`
  for t_u *in* $\mathcal{T}_u$ **do**
    Find the embedding vector of $\mathbf{E_u}$ of t_u corresponding to the previous frame;
    Find surrounding embedding vectors $\mathbf{E_d}$ with the center point of t_u in the detection frame;
    Select the most similar appearance embedding vector $\mathbf{E_d^s}$ based on the cosine similarity;
    Record the coordinates (x_E, y_E) of the center point corresponding to $\mathbf{E_d^s}$;
    if $\mid E_u - E_d^s \mid < \epsilon_r$ **then**
      Update the coordinates of the center point in t_u with (x_E, y_E);
      $\mathcal{T}_{reback} \leftarrow t_u$;
  `// delete unmatched tracks`
  $\mathcal{T} \leftarrow \mathcal{T} \backslash \mathcal{T}_{\nabla]-\nabla]\Updownarrow\dashv\rangle\backslash}$;
  `// initialize new tracks`
  for d *in* $\mathcal{D}_{remain}$ **do**
    if $d.score > \epsilon$ **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$;
final ;
return $\mathcal{T}$;

As shown in Figure 4, when the target is blocked and the detector fails, we use a Kalman filter to predict the center point position of the unmatched tracking targets. In order to compensate for the drift of the Kalman filter, we use appearance information to modify the prediction results of the Kalman filter. We select the appearance embedding vectors in the 3×3 range around the prediction center point (C_i, C_j) by the Kalman filter. Denoting

the embedding vector of the unmatched tracking target by $\mathbf{E_u}$, we follow Equation (5) to determine whether the unmatched tracking target can be retrieved.

$$S(t_u) = \begin{cases} \text{retrieved}, & \text{if } Dis_{min} < \epsilon_r \\ \text{unretrieved}, & \text{otherwise} \end{cases} \tag{5}$$

where Dis_{min} represents the minimum cosine distance among the 3×3 range around (C_i, C_j), ϵ_r denotes the tracking retrieval threshold. By denoting the appearance embedding vector on pixel (i, j) by $E_d^{(i,j)}$, Dis_{min} can then be represented by Equation (6).

$$Dis_{min} = Min_{i,j}(\{F_{cd}(\mathbf{E_d^{i,j}}, \mathbf{E_u})\}_{i \in [C_i - 1, C_i + 1], j \in [C_j - 1, C_j +]}) \tag{6}$$

where $F_{cd}(\cdot)$ indicates the results of the cosine distance between two vectors. Afterward, if the state of the unmatched tracking target is judged to be retrieved, we obtain (i_{min}, j_{min}) according to Dis_{min}. Finally, set (i_{min}, j_{min}) as the center point of the retrieval box, and make the width and height of the retrieval box consistent with the tracked target in the previous frame.

With the tracking retrieval mechanism, we can recover the occluded (failed) detection boxes by using the predictions of the Kalman filter. At the same time, the embedding information can be used to correct the predicted position of the Kalman filter, so as to update the parameters of the Kalman filter and reduce the accumulated error of the Kalman filter.

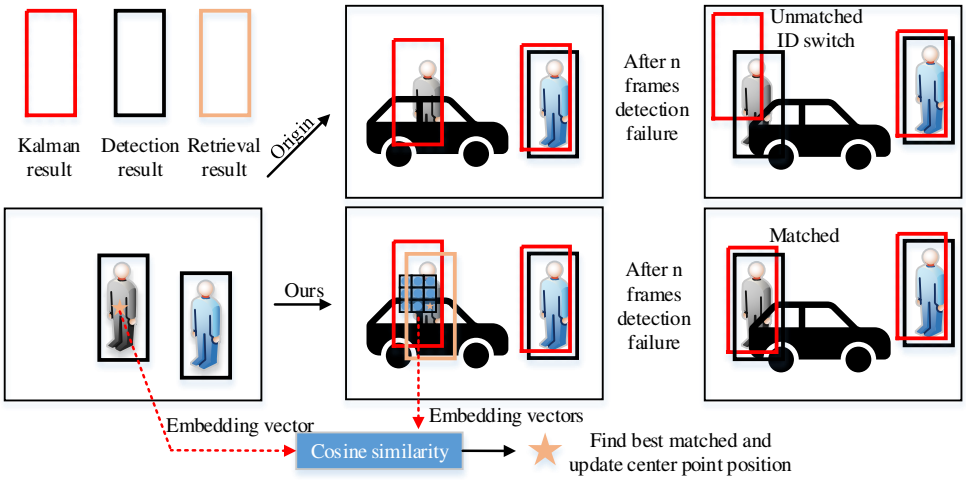


Figure 4. Tracking retrieval process. The five-pointed star indicates the position of the best matching embedding vector.

4. Experiments

4.1. Datasets and Metrics

4.1.1. Datasets

We evaluate SimpleTrack on private detection tracks of the MOT17 [25] and MOT20 [26] datasets. The former contains 14 different video sequences for multi-target tracking, recorded by fixed or moving cameras. The latter consists of 8 video sequences with a fixed camera focusing on tracking in very crowded scenes, 4 for training and testing each. For ablation studies, we follow [27–31] and split the train set into two parts for ablative experiments as the annotations of the test split are not publicly available. We fuse the CrowdHuman [32] and MOT17, with half as the training dataset for ablation experiments following [10,30,31,33,34]. We add the ETH [35], CityPerson [36], CalTech [37], CUHK-SYSU [38] and PRW [39] datasets for training following [11–13] when testing on the test set of MOT17.

4.1.2. Evaluation Metrics.

To evaluate the tracking performance, we use TrackEval to evaluate all metrics, including MOTA [40], IDF1 [41], false positives (FP), false negatives (FN), identity switches (IDSW) and the recently proposed HOTA [42]. HOTA can comprehensively evaluate the performance of detection and data association. IDF1 focuses more on the association performance and MOTA evaluates the detector ability and focuses more on detection performance.

4.2. Implementation Details

4.2.1. Tracker

In the tracking phase, the default high detection score threshold τ_{high} is 0.3, the low threshold τ_{low} is 0.2, the trajectory initialization score ϵ is 0.6, and the trajectory retrieval score ϵ_r is 0.1, unless otherwise specified. In the linear assignment step, for the high-confidence detection, the assignment threshold is 0.8, and for the low-confidence detection, the assignment threshold is 0.4.

4.2.2. Detector and Embedding

We use SimpleTrack to extract the location features and appearance features of objects. For SimpleTrack, the backbone is DLA-34, which initializes weights with a COCO-pretrained model. The training schedule is 30 epochs on the combination of MOT17, CrowdHuman and other datasets mentioned above. The input image size is 1088×608. Rotation, scaling and color jittering are adopted as data augmentation techniques during our training phase. The model is trained on 4 NVIDIA TITAN RTX with a batch size of 32. The optimizer is Adam and the initial learning rate is set to 2×10^{-4}, which decays to 2×10^{-5} in the 20th epoch. The total training time is around 25 h. FPS is measured with a single NVIDIA RTX2080Ti and the batch size is set to 1.

4.3. Ablation Studies

4.3.1. Ablation on SimpleTrack

The innovation of SimpleTrack is mainly composed of bottom-up decoupling, the EG similarity matrix and tracking retrieval. We conduct ablation experiments on the MOT17 validation set for these three modules. The results are shown in Table 1. It can be observed that adding bottom-up decoupling to FairMOT increases IDF1 and MOTA. In addition, after replacing the similarity matrix of JDE-based methods with the EG matrix, the strategy improves IDF1 from 76.1 to 78.1, MOTA from 71.4 to 72.5 and HOTA from 60.2 to 61.5 and decreases IDs from 451 to 186. After further adding the tracking retrieval mechanism, the IDF1 metric increases from 78.1 to 78.5 and HOTA from 61.5 to 61.7, and the IDs metric decreases from 186 to 182. These results prove that the modules proposed in SimpleTrack are necessary and effective.

Table 1. Ablation experiment on SimpleTrack. ✓ denotes addition of this module to the baseline, which is FairMOT. BU-D, EG and TR stand for bottom-up decoupling, EG similarity matrix and tracking retrieval strategy, respectively. The best results are shown in **bold**.

Model Settings			**Evaluation Indicators**						
BU-D	**EG**	**TR**	**IDF1↑**	**MOTA↑**	**HOTA↑**	**IDs↓**	**FP↓**	**FN↓**	**FPS↑**
			75.6	71.1	-	327	-	-	-
✓			76.1	71.4	60.2	451	3319	11,655	19.7
✓	✓		78.1	72.5	61.5	186	3260	**11,430**	**24**
✓	✓	✓	**78.5**	**72.5**	**61.7**	**182**	**3212**	11,456	23.8

4.3.2. Analysis of the Hyperparameters of EG Matrix

We test different sets of hyperparameters of Equation (4) in Figure 5. Set the parameter λ_1 of embedding similarity to 1 and increase the parameter λ_2 of GioU similarity from 0.1 to 4. It can be observed that the tracking performance of the algorithm is better when λ_1 is set to 1.0 and λ_2 is set from 0.5 to 0.9, because the interval of embedding similarity is [0, 1], and the interval of GioU similarity is [0, 2]. Therefore, in order to balance the weights of embedding similarity and GioU similarity, EG matrix set $\lambda_1 = 1.0$ and $\lambda_2 = 0.5$.

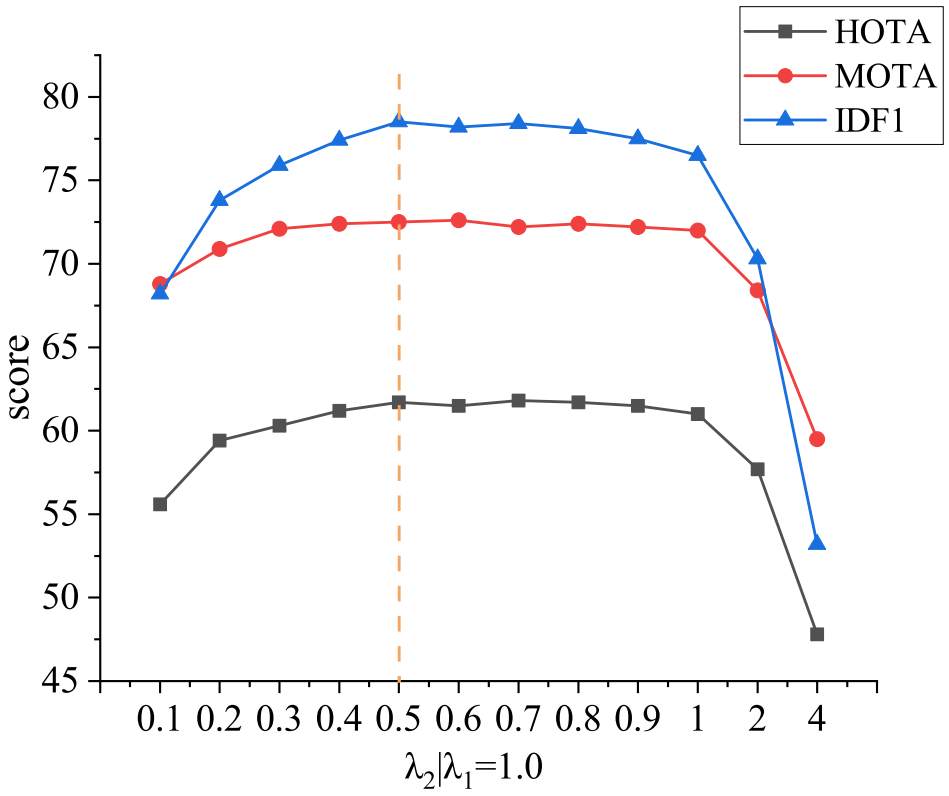


Figure 5. Experiments for hyperparameters in EG matrix in MOT17-half val. The blue, red and black lines represent IDF1, MOTA and HOTA indicators, respectively. The parameters selected in the paper are shown by the dotted line.

4.3.3. Comparison with Preceding SOTAs

In this part, we compare the performance of SimpleTrack with preceding SOTA methods on MOT17 and MOT20. The results are reported in Tables 2 and 3, respectively. As shown in these two tables, SimpleTrack showed the best results in various metrics and surpassed the contrasted counterparts by large margins, especially on the HOTA, IDF1 and IDS metrics. Moreover, compared with other MOT tracking methods, SimpleTrack has an obvious speed advantage.

Table 2. Comparison of the state-of-the-art methods under the "private detector" protocol on the MOT17 test set. The best results are shown in **bold**. MOT17 contains rich scenes and half of the sequences are captured with camera motion. * indicates the addition of linear interpolation and † indicates JDE-based methods.

Method	HOTA↑	IDF1↑	MOTA↑	IDs↓	FP↓	FN↓	FPS↑
TraDes [30] †	52.7	63.9	69.1	3555	20,892	150,060	17.5
MAT[43]	53.8	63.1	69.5	2844	30,660	138,741	9.0
QuasiDense [16] †	53.9	66.3	68.7	3378	26,589	146,643	20.3

Table 2. *Cont.*

Method	HOTA↑	IDF1↑	MOTA↑	IDs↓	FP↓	FN↓	FPS↑
SOTMOT [44]	-	71.9	71.0	5184	39,537	118,983	16.0
TransCenter [45]	54.5	62.2	73.2	4614	23,112	123,738	1.0
GSDT [46] †	55.2	66.5	73.2	3891	26,397	120,666	4.9
PermaTrackPr [47]	55.5	68.9	73.8	3699	28,998	115,104	11.9
TransTrack [33]	54.1	63.5	75.2	3603	50,157	86,442	10.0
FUFET [28]	57.9	68.0	76.2	3237	32,796	98,475	6.8
FairMOT [12] †	59.3	72.3	73.7	3303	27,507	117,477	18.9
CSTrack [13] †	59.3	72.6	74.9	3567	23,847	114,303	15.8
Semi-TCL [48]	59.8	73.2	73.3	2790	22,944	124,980	-
ReMOT [49]	59.7	72.0	**77.0**	2853	33,204	**93,612**	1.8
CrowdTrack [50]	60.3	73.6	75.6	2544	25,950	109,101	-
CorrTracker [29] †	60.7	73.6	76.5	3369	29,808	99,510	15.6
RelationTrack [21] †	61.0	74.7	73.8	1374	27,999	118,623	8.5
SimpleTrack(Ours) †	61.0	75.7	74.1	1500	**17,379**	127,053	**22.53**
SimpleTrack(Ours) *	**61.6**	**76.3**	75.3	**1260**	22,317	116,010	-

Table 3. Comparison of the state-of-the-art methods under the "private detector" protocol on the MOT20 test set. The best results are shown in **bold**. The scenes in MOT20 are much more crowded than those in MOT17. * indicates the addition of linear interpolation and † indicates JDE-based methods.

Method	HOTA↑	IDF1↑	MOTA↑	IDs↓	FP↓	FN↓	FPS↑
MLT [51]	43.2	54.6	48.9	2187	45,660	216,803	3.7
FairMOT [12] †	54.6	67.3	61.8	5243	103,440	**88,901**	**13.2**
TransCenter [45]	-	50.4	61.9	4653	45,895	146,347	1.0
TransTrack [33]	48.5	59.4	65.0	3608	27,197	150,197	7.2
Semi-TCL [48]	55.3	70.1	65.2	4139	61,209	114,709	-
CorrTracker [29] †	-	69.1	65.2	5183	79,429	95,855	8.5
CSTrack [13] †	54.0	68.6	66.6	3196	25,404	144,358	4.5
GSDT [46] †	53.6	67.5	67.1	3131	31,913	135,409	0.9
SiamMOT [17] †	-	67.8	70.7	-	22,689	125,039	6.7
RelationTrack [21] †	56.5	70.5	67.2	4243	61,134	104,597	2.7
SOTMOT [44]	-	**71.4**	68.6	4209	57,064	101,154	8.5
SimpleTrack(Ours) †	56.6	69.6	70.6	2434	**18,400**	131,209	7.0
SimpleTrack(Ours) *	**57.6**	70.2	**72.6**	**1785**	25,515	114,463	-

4.3.4. Visualization Results

We show some scenarios that are prone to identity switching in Figure 6, which contains three sequences from the half validation set of MOT17. We use different tracking strategies to generate the visualization results. It can be observed that SimpleTrack can effectively deal with the identity switching problem caused by the occlusion of the tracking targets. In addition, some tracking examples on the MOT17 test datasets are shown in Figure 7.

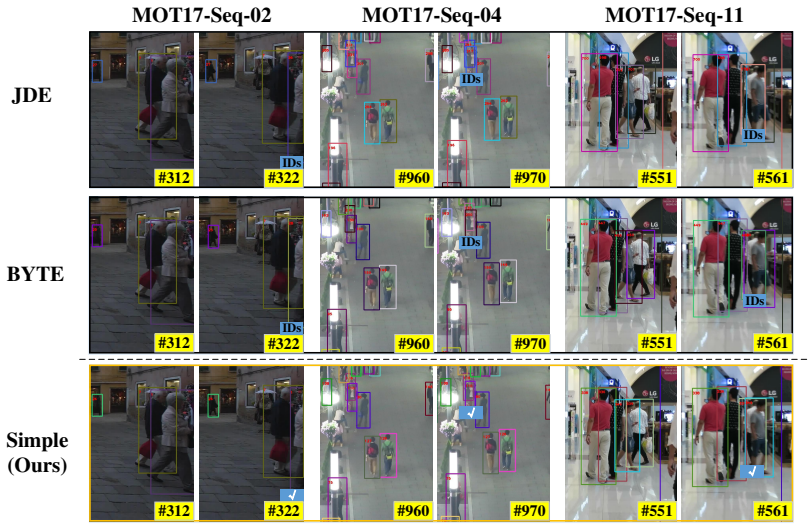


Figure 6. Robustness of our tracking strategy compared to BYTE and JDE-based methods. Boxes with the same color indicate that the tracking targets have the same identity; IDs indicates that the tracking targets have switched their identities. The check mark indicates that the identity of the target has not changed. The #number indicates that the frame number in mot17 video sequence.

Figure 7. Tracking results of SimpleTrack on the MOT17 test dataset. The #number indicates that the frame number in mot17 video sequence.

5. Discussion

This section mainly discusses the similarity matrix and the tracking association method. Section 5.1 analyzes and compares the performance of our proposed EG matrix with other existing similarity metrics, and applies the EG matrix to other JDE-based methods to analyze the universality of the EG matrix. Section 5.2 compares the speed and accuracy of our proposed tracking strategy with other existing tracking methods.

5.1. Analysis of the Similarity Matrix

5.1.1. Performance Compared with Other Similarity Metrics

We employ different distance matrices as the similarity measure and evaluate their data association ability on the half validation set of MOT17. It can be obtained from Table 4 that only using the GioU or embedding matrix for data association does not result in good performance. Moreover, the table shows that the combination of the embedding matrix and IoU matrix can improve the association effect but reduces the result of MOTA. Compared with the IoU matrix used in detection-based methods, our EG matrix improves the IDF1 from 75.7 to 78.5 and HOTA from 60.4 to 61.7 and decreases IDs from 285 to 182. Compared with the embedding and motion matrix used in JDE-based methods, our EG matrix improves both the MOT metrics and tracking speed.

Table 4. Data association comparison of different similarity matrices. The best results are shown in **bold**.

Similarity Matrix	IDF1↑	MOTA↑	HOTA↑	IDs↓	FP↓	FN↓	FPS↑
IoU	75.7	72.5	60.4	285	3510	11,048	**25**
GioU	66.4	70.4	54.8	378	4631	**10,956**	23.6
Embedding	64.1	65.0	53.4	749	6120	12,012	24.2
Embedding and Motion	76.1	71.4	60.2	451	3319	11,655	19.7
Embedding and IoU	77.2	72.3	61.4	263	**2560**	12,144	24
Embedding and GioU	**78.5**	**72.5**	**61.7**	**182**	3212	11,456	23.8

5.1.2. Applications in Other JDE-Based Trackers

We apply our EG matrix to five different JDE-based trackers, including JDE [11], FairMOT [12], CSTrack [13], TraDes [30] and QuasiDense [16]. Among these trackers, JDE, FairMOT, CSTrack, TraDes merge the motion and Re-ID similarity and the first three methods follow the same fusion strategy. QuasiDense uses Re-ID similarity alone. It can be observed from Table 5 that using the EG matrix instead of the EM matrix can enhance the tracking performance and improve the tracking speed. Taking the JDE [11] method as an example, only using the EG matrix to replace the EM matrix can improve the HOTA from 50.1 to 50.9, IDF1 from 63 to 64.4, MOTA from 59.3 to 59.5 and FPS from 16.64 to 21.29 and decreases the IDs from 621 to 558. Combined with the BYTE strategy, our EG matrix still improves the HOTA from 50.4 to 50.9, IDF1 from 64.1 to 64.4 and FPS from 18.52 to 25.48 and decreases the IDs from 437 to 388.

5.2. Analysis of the Association Methods

5.2.1. Accuracy Compared with Other Association Methods

We compare SimpleTrack with other association methods, including the recent SOTA algorithm BYTE and the tracking algorithm used in JDE-based methods [11–13,17]. As shown in Table 6, SimpleTrack improves the IDF1 metric of JDE from 76.1 to 78.5, MOTA from 71.4 to 72.5 and HOTA from 60.2 to 61.7 and decreases IDs from 451 to 182. Compared with BYTE, we can see that SimpleTrack improves the IDF1 from 75.7 to 78.5 and HOTA from 60.4 to 61.7, and decreases IDs from 285 to 182. These demonstrate that our tracking method is more effective than the JDE strategy, and it can improve the accuracy of data association compared to the BYTE strategy.

Table 5. Results of applying SimpleTrack to five different JDE-based trackers on the MOT17 validation set. Blue represents the tracking method using only the EG matrix, and red represents the tracking method combining the EG matrix and BYTE.

Method	Similarity	w/BYTE	HOTA↑	IDF1↑	MOTA↑	IDs↓	FPS↑
JDE [11]	EM	-	50.1	63.0	59.3	621	16.64
	EG	-	50.9	64.4	59.5	558	21.29
	EM	✓	50.4	64.1	60.2	437	18.52
	EG	✓	50.9	64.8	60.1	388	25.48
FairMOT [12]	EM	-	57.0	72.4	69.1	372	21.01
	EG	-	57.5	73.3	69.5	236	25.18
	EM	✓	-	74.2	70.4	232	-
	EG	✓	58.5	74.5	70.6	188	24.70
CSTrack [13]	EM	-	58.7	72.0	67.9	423	20.39
	EG	-	59.3	73.0	68.2	322	24.3
	EM	✓	59.8	73.9	69.2	298	20.72
	EG	✓	60.0	73.8	69.6	249	24.25
TraDes [30]	EM	-	58.6	71.7	68.3	293	15.8
	EM	✓	58.4	71.2	68.9	263	16.22
	EG	✓	59.0	71.5	68.5	483	16.5
QuasiDense [16]	EM	-	56.2	67.7	67.1	386	4.10
	EM	✓	58.5	71.9	67.4	295	4.80
	EG	✓	57.9	70.9	67.5	252	4.80

Table 6. Comparison of different association methods on the MOT17 validation set. JDE expresses the tracking strategy employed by [11–13,17] and BYTE expresses the tracking strategy employed by [10]. The best results are shown in **bold**.

Tracking Method	IDF1↑	MOTA↑	HOTA↑	IDs↓	FP↓	FN↓	FPS↑
JDE	76.1	71.4	60.2	451	3319	11,655	19.7
BYTE	75.7	72.5	60.4	285	3510	**11,048**	**25**
SimpleTrack (Ours)	**78.5**	**72.5**	**61.7**	**182**	**3212**	11,456	23.8

5.2.2. Speed Compared with Other Association Methods

From Tables 4 and 6, we can observe that our SimpleTrack algorithm utilizes the embedding information but is still nearly 20% faster than the JDE-based tracking strategy. A more detailed comparison of different video sequences can be observed in Figure 8a. It can be observed that our tracking algorithm is only slightly slower than BYTE, which does not utilize the embedding information. According to Figure 8b, we can see the time consumption of the main modules in the tracking phase. It shows that the JDE-based tracking strategy spends a lot of time in fusing the embedding and motion information, which is represented by the orange dotted square in Figure 8b. For the EG matrix, we only need to calculate the GioU distance and add it to the embedding distance. The time consumption is represented by the orange dotted star in Figure 8b.

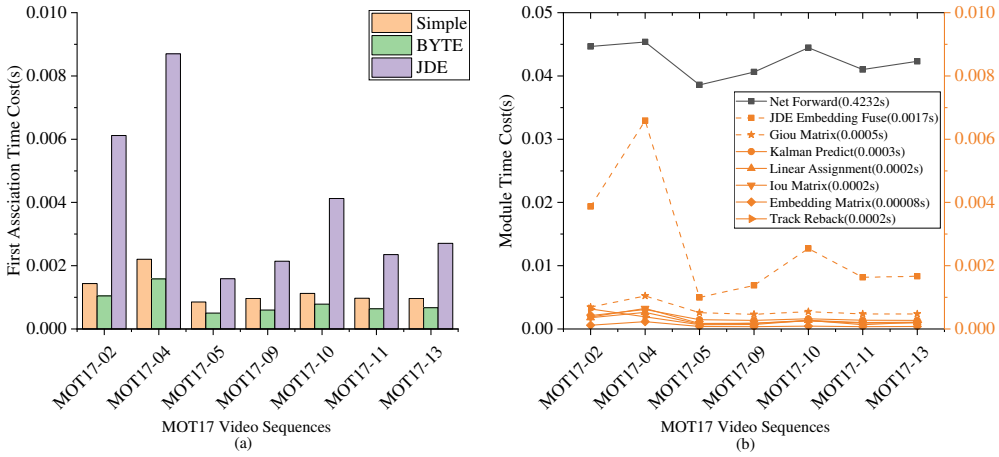


Figure 8. Comparison of different tracking algorithm speeds. (**a**) shows the tracking speed of different tracking algorithms. (**b**) shows the time consumption of several main modules in the tracking phase.

6. Conclusions and Future Work

We propose a simple yet effective data association matrix, the EG matrix, for JDE-based multi-object tracking methods. The EG matrix can be easily applied to existing trackers and improves not only the tracking effect but also the speed of JDE-based tracking methods. In addition, we design a bottom-up feature fusion module for decoupling Re-ID and detection tasks, and present a novel tracklet retrieval strategy for mitigating the loss of detection targets. These innovations together form our SimpleTrack, which achieves 61.6 HOTA and 76.3 IDF1 on the test set of MOT17 with 23 FPS, ranking first among all the JDE-based methods.

SimpleTrack has a strong data association ability due to adopting the EG matrix and decoupling feature extraction module, which can be applied to multi-target tracking in some complex scenes. In the future work, we will consider the enhancement of target features in the time dimension and design an anti-occlusion feature extraction network based on our SimpleTrack framework. Moreover, we hope that the EG matrix can become the standard association matrix of JDE-based methods in multi-object tracking.

Author Contributions: Conceptualization J.L.; investigation J.L., Y.D. and Y.Z.; methodology J.L. and Y.D.; project administration Y.D.; software J.L.; supervision Y.D.; validation J.L.; visualization J.L. and W.L.; writing—original draft J.L., Y.D. and H.-L.W.; writing— review and editing Y.D., H.-L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets in Experiment and Discussion are available at https://motchallenge.net.

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

References

1. Vandenhende, S.; Georgoulis, S.; Van Gansbeke, W.; Proesmans, M.; Dai, D.; Van Gool, L. Multi-task learning for dense prediction tasks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3614–3633. [CrossRef] [PubMed]
2. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016 ; IEEE: Piscataway, NJ, USA , 2016; pp. 3464–3468.
3. Bochinski, E.; Eiselein, V.; Sikora, T. High-speed tracking-by-detection without using image information. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September, 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.

4. Liu, Q.; Chu, Q.; Liu, B.; Yu, N. GSM: Graph Similarity Model for Multi-Object Tracking. In Proceedings of the IJCAI, Yokohama, Japan, 11–17 July 2020; pp. 530–536.
5. Specker, A.; Stadler, D.; Florin, L.; Beyerer, J. An occlusion-aware multi-target multi-camera tracking system. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4173–4182.
6. Tang, S.; Andriluka, M.; Andres, B.; Schiele, B. Multiple people tracking by lifted multicut and person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3539–3548.
7. 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; IEEE: Piscataway, NJ, USA, 2017; pp. 3645–3649.
8. Xiao, B.; Wu, H.; Wei, Y. Simple baselines for human pose estimation and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 466–481.
9. Xu, J.; Cao, Y.; Zhang, Z.; Hu, H. Spatial-temporal relation networks for multi-object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 3988–3998.
10. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2021**, arXiv:2110.06864.
11. 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; Springer: Berlin/Heidelberg, Germany, 2020; pp. 107–122.
12. 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]
13. Liang, C.; Zhang, Z.; Lu, Y.; Zhou, X.; Li, B.; Ye, X.; Zou, J. Rethinking the competition between detection and reid in multi-object tracking. *IEEE Trans. Image Process.* **2020**, *31*, 3182–3196. [CrossRef]
14. Lu, Z.; Rathod, V.; Votel, R.; Huang, J. Retinatrack: Online single stage joint detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 14668–14678.
15. Zhang, Y.; Wang, C.; Wang, X.; Liu, W.; Zeng, W. Voxeltrack: Multi-person 3d human pose estimation and tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1. [CrossRef]
16. 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, Nashville, TN, USA, 20–25 June 2021; pp. 164–173.
17. Liang, C.; Zhang, Z.; Zhou, X.; Li, B.; Lu, Y.; Hu, W. One More Check: Making" Fake Background" Be Tracked Again. *Proc. AAAI Conf. Artif. Intell.* **2021**, *36*, 1546–1554. [CrossRef]
18. Mouawad, I.; Odone, F. FasterVideo: Efficient Online Joint Object Detection and Tracking. In Proceedings of the International Conference on Image Analysis and Processing, Paris, France, 17–18 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 375–387.
19. Chan, S.; Jia, Y.; Zhou, X.; Bai, C.; Chen, S.; Zhang, X. Online Multiple Object Tracking Using Joint Detection and Embedding Network. *Pattern Recognit.* **2022**, *130*, 108793. [CrossRef]
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
21. Yu, E.; Li, Z.; Han, S.; Wang, H. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *IEEE Trans. Multimed.* **2022**. [CrossRef]
22. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
23. Chen, L.; Ai, H.; Zhuang, Z.; Shang, C. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
24. Stadler, D.; Beyerer, J. Modelling Ambiguous Assignments for Multi-Person Tracking in Crowds. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 133–142.
25. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
26. 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.
27. 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, Nashville, TN, USA, 20–25 June 2021; pp. 14329–14339.
28. Shan, C.; Wei, C.; Deng, B.; Huang, J.; Hua, X.S.; Cheng, X.; Liang, K. Tracklets predicting based adaptive graph tracking. *arXiv* **2020**, arXiv:2010.09015.
29. 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, Nashville, TN, USA, 20–25 June 2021; pp. 3876–3886.
30. Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; Yuan, J. Track to detect and segment: An online multi-object tracker. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12352–12361.

31. 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; Springer: Berlin/Heidelberg, Germany, 2020; pp. 474–490.
32. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv* **2018**, arXiv:1805.00123.
33. Sun, P.; Cao, J.; Jiang, Y.; Zhang, R.; Xie, E.; Yuan, Z.; Wang, C.; Luo, P. Transtrack: Multiple object tracking with transformer. *arXiv* **2020**, arXiv:2012.15460.
34. Zeng, F.; Dong, B.; Wang, T.; Zhang, X.; Wei, Y. Motr: End-to-end multiple-object tracking with transformer. *arXiv* **2021**, arXiv:2105.03247.
35. Ess, A.; Leibe, B.; Schindler, K.; Van Gool, L. A mobile vision system for robust multi-person tracking. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–8.
36. Zhang, S.; Benenson, R.; Schiele, B. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3213–3221.
37. Dollár, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: A benchmark. In Proceedings of the 2009 IEEE Conference on Computer vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 304–311.
38. Xiao, T.; Li, S.; Wang, B.; Lin, L.; Wang, X. Joint detection and identification feature learning for person search. In Proceedings of the IEEE Conference on Computer vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3415–3424.
39. Zheng, L.; Zhang, H.; Sun, S.; Chandraker, M.; Yang, Y.; Tian, Q. Person re-identification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1367–1376.
40. Bernardin, K.; Elbs, A.; Stiefelhagen, R. Multiple object tracking performance metrics and evaluation in a smart room environment. In Proceedings of the Sixth IEEE International Workshop on Visual Surveillance, in Conjunction with ECCV, Graz, Austria, 13 May 2006; Citeseer: Princeton, NJ, USA, 2006; Volume 90.
41. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 17–35.
42. 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]
43. Han, S.; Huang, P.; Wang, H.; Yu, E.; Liu, D.; Pan, X. Mat: Motion-aware multi-object tracking. *Neurocomputing* **2022**, *476*, 75–86. [CrossRef]
44. Zheng, L.; Tang, M.; Chen, Y.; Zhu, G.; Wang, J.; Lu, H. Improving multiple object tracking with single object tracking. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2453–2462.
45. 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.
46. Wang, Y.; Kitani, K.; Weng, X. Joint object detection and multi-object tracking with graph neural networks. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 13708–13715.
47. Tokmakov, P.; Li, J.; Burgard, W.; Gaidon, A. Learning to track with object permanence. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 10860–10869.
48. Li, W.; Xiong, Y.; Yang, S.; Xu, M.; Wang, Y.; Xia, W. Semi-tcl: Semi-supervised track contrastive representation learning. *arXiv* **2021**, arXiv:2107.02396.
49. Yang, F.; Chang, X.; Sakti, S.; Wu, Y.; Nakamura, S. ReMOT: A model-agnostic refinement for multiple object tracking. *Image Vis. Comput.* **2021**, *106*, 104091. [CrossRef]
50. Stadler, D.; Beyerer, J. On the Performance of Crowd-Specific Detectors in Multi-Pedestrian Tracking. In Proceedings of the 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Washington, DC, USA, 16–19 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–12.
51. 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]