Multi-Object Tracking with Confidence-Based Trajectory Prediction Scheme

Kai Yi , Jiarong Li , Yi Zhang *

Posted Date: 28 October 2025

doi: 10.20944/preprints202510.2145.v1

Keywords: multi-object tracking; confidence score; trajectory prediction; data association

 doi:10.20944/preprints202510.2145.v1

Article

Multi-Object Tracking with Confidence-Based Trajectory Prediction Scheme

Kai Yi [1], Jiarong Li [2] and Yi Zhang [2,*]

[1] Intelligent Policing Key Laboratory of Sichuan Province, China
[2] College of Computer Science, Sichuan University, Chengdu 610065, China
* Correspondence: yi.zhang@scu.edu.cn

Abstract

The goal of Multi-Object Tracking (MOT) is to associate multiple objects across frames and maintain continuous and stable trajectories. Currently, much attention has been paid to the data association problems, where many methods filter the detection boxes for object matching based on the confidence scores (CS) of the detectors without fully utilizing the detection results. On the other hand, Kalman filter (KF) is a traditional means for MOT, which matches updates a predicted trajectory with a detection box. However, under crowded scenes, the noise will create low confident detection boxes, causing identity switch (IDS) and tracking failure. In this paper, we thoroughly investigate the limitations of existing trajectory prediction schemes and prove that KF can still achieve competitive results if proper care is taken to handle the noise. We propose a Confidence-based Trajectory prediction scheme (dubbed ConfMOT) based on KF. The CS of the detection results is used to adjust the noise during updating KF and to predict the trajectories of the tracked objects. While a cost matrix (CM) is constructed to measure the cost of successful matching of unreliable objects. Meanwhile, each trajectory is labeled with a unique CS, while the lost trajectories that have not been updated for a long time will be removed. Our tracker is simple yet efficient. Extensive experiments have been conducted on mainstream datasets, where our tracker has exhibited superior performances than other advanced competitors.

Keywords: multi-object tracking; confidence score; trajectory prediction; data association

1. Introduction

MOT has been highly spotlighted recently, which deals with the detection of multiple objects across video frames and identifier assignment for each trajectory. It plays a vital role in intelligent video surveillance, military surveillance and autonomous driving etc. Over the past few years, MOT has benefited greatly from the rapid development of advanced object detection schemes. Especially, the tracking-by-detection (TBD) paradigm has achieved tremendous success in MOT. Most existing works divided MOT into detection and association tasks and solve them independently under TBD framework [1,2]. Typically, they firstly apply object detectors to detect known targets and infer the trajectories via different matching strategies for cross-frame data association. For instance, MOTRv2 [3] integrates YOLOX with MOTR tracker, where YOLOX creates high-quality object proposals to facilitate the association process by MOTR. However, TBD has a notable problem: the tracking performance is highly relied on the detection results (i.e. the core impact on the tracking performance is the threshold of the detector and subsequent matching mechanisms). In particular, the fixed threshold of the detector is difficult to cope with complex situations where multiple objects interact. Classical TBD based methods include SORT [4], DeepSORT [1], and ByteTrack [2] etc. Among them, SORT [4] simply uses intersection over union (IoU) and motion information to measure matching similarity and creates new trajectories and reject lost ones. DeepSORT [1] employs a matching cascade and calculates Mahalanobis distance between predicted track boxes and detection boxes (instead of IoU), since the accuracy of the inactive tracks decreases over time. However, the Mahalanobis distance is only used to avoid

 doi:10.20944/preprints202510.2145.v1

irrational assignments, which is not suitable for matching. ByteTrack [2] associates the low-confident detections that are unmatched in the first stage, which indeed pushes the record of MOT to a new level and thereby has been adopted by many following works. However, the high-confident detections are usually preferred over low-confident ones, while the low-confident detections will no longer be assigned to inactive trajectories. Fortunately, with the continuous development of object detection, the detection results are becoming increasingly reliable.

Apart from detection, frequent object occlusions under crowded scenes are still the major challenge for MOT[5], causing IDS. Various appearance models with effective feature learning methods are employed to calculate feature similarity so as to correct object ID when occlusions occur [6]. TransCenter [7] advocates the use of image-related dense detection queries and efficient sparse tracking queries under their query learning networks (QLN). AMtrack[8] is built upon an encoder-decoder Transformer architecture, which realizes data association across frames via evolving a set of track predictions. However, it has slower inference speed, and its predictive ability on linear motion is not as good as KF. DeNoising-MOT [9] learns denoising process in an encoder-decoder structure and develops a cascaded mask strategy to prevent the mutual suppression between neighboring trajectories. However, as the scenes become more complex, the tracking performances of the above methods will still be severely affected by occlusion. Although recent improvements in detector performance are conducive to correct object associations., even the best detectors make false predictions. There remains a room for improvement in existing methods using detections.

The linear KF algorithm has long been widely used in visual object tracking [2,10]. During the update process, the measurement noise is set to a constant value. For a reliable detection result, the measurement noise should be low accordingly. To obtain more accurate update results, we adjust the measurement noise based on CS (output by the detector) so as to ensure the quality of the tracking results. During the matching process between the prediction and detection boxes, IoU and feature similarities are usually used to form CM (to measure the discrepancy between the predicted target position and the detection box) of the Hungarian algorithm. When computing CM, the reliability of the current bounding box is often ignored. We therefore fuse CS into CM to improve its reliability. In some cases, some trajectories exist for only a period of time but discontinue in the following frames, which disturb the subsequent trajectory matching and cause incorrect association. We regard them lost and hence delete them in due course. Through analyzing the connection between the detection and associations, we believe the detection results have not been fully utilized. Our results show that the performance of the current TBD based methods can be further improved by combining CS of the detection results. Therefore, an appropriate detection processing scheme from tracking perspective is required to bridge the gap between detection and tracking applications.

In a nutshell, we have made contributions in 3 key stages of MOT, including data association, trajectory prediction and trajectory management:

1. In data association stage, confidence scores are integrated into the cost matrix to improve matching accuracy and achieve more reliable assignment results.

2. In trajectory prediction stage, a confidence based trajectory prediction scheme has been proposed based on KF, which achieves more accurate prediction performance by controlling the measurement noise in KF.

3. In trajectory management stage, a trajectory deletion scheme has been proposed to determine the duration of trajectories and delete less reliable trajectories to avoid possible incorrect matches.

The experimental results shows that our tracking scheme further improves the performances of the current TBD based methods. On the MOT Challenge benchmarks, ConfMOT ranks among the top on MOT17 [12] and MOT20 [13]. ConfMOT achieves 64.5% HOTA and 80.5% MOTA on the MOT17 dataset, and 62.9% HOTA and 78.3% of MOTA on the MOT20 respectively.

The rest of the paper is organized as follows: section 2 briefly reviews related work, section 3 describes our proposed method in details. Experimental results are provided in section 4 with ablation studies. Finally, a conclusion is drawn in section 5.

 doi:10.20944/preprints202510.2145.v1

2. Related works

With the rapid development of object detection [11], the current research hotspots of visual tracking [2,14] has been shifted from designing complex and individual trackers to subsequent data association schemes that are built upon advanced detectors. The main goal is to maintain the correct trajectories of the objects.

2.1. Motion Models

Most of the recent TBD based methods are based on motion models (under the assumption of constant velocity). As one of the classical models, KF is a Bayes filter which predicts and updates in a recursive way. The ground-truth state is an unobserved Markov process, while the measurement is observed from a hidden Markov model. The measurement noise is the noise exists in signal observation process. OC-SORT [14] uses virtual trajectories to smooth parameters for solving the cumulative error of KF. ByteTrack [2], on the other hand, uses a normal linear KF to associate every detection box to recover true objects and filter out the background elements. However, the above 2 methods apply the same KF to all targets, achieving trajectory prediction quickly and efficiently, but ignoring the differences between various detections. You et al. [15] proposes a novel Spatial-Temporal Topology-based Detector (STTD) algorithm and introduces a topology structure to dynamics of moving targets. It indeed reduces false positives, but since it only considers group motion between targets to build topology, it hence overlooks camera motion. Li et al. [6] presents a TBD based framework for MOT, where a deep association network is developed (followed by detection) to infer the association degree of objects. Despite some advantages in correlation operation, the object detection and tracking tasks are somehow independent of each other, where 2 deep feature extractions are needed in each stage. GIAOTracker [16] proposes a modified KF with confidence scores. Khurana et al. [17] adopts depth estimates from a monocular depth estimator to forecast the trajectories of occluded people. However, they rely heavily on appearance features, which introduces high computational cost. Wang et al. [18] publishes a novel approach to tackle long-term vehicle tracking without appearance information. But it cannot be extended to other types of objects. Tracktor++ [19] employs camera motion compensation for frame alignment. The former method performs global association by developing an appearance-free link model to address missing detection and missing association problems. While the latter employs global information and optimization strategies to associate object trajectories. But it is not applicable to online real-time tracking.

2.2. Cost Matrix

In order to update the objects' trajectories, the prediction boxes obtained from the KF prediction phase need to be matched with the detection boxes obtained from the detector. OC-SORT [14] claims that using IoU alone is sufficient to generate good results if the detector is accurate enough. But it ignores the impact of detection differences on the cost matrix. MOTDT [20] constructs the cost matrix of the tracking processes with appearance features and IoU respectively. FairMOT [21], on the other hand, fuses IoU and appearance feature, but it does not consider the negative impact of unreliable cost matrix. The above 2 methods combine multiple indicators to form a cost matrix, which can solve the confused assignment problem to some extent. However, extraction of appearance information also slows down the tracking speed in return.

2.3. Trajectories Management

Most trackers set different durations for lost trajectories for different datasets. For example, JDE [10] and FairMOT [21] save the unmatched trajectories for 30 frames, while ByteTrack [2] keeps lost trajectories for 14 frames for MOT17-06 and 25 frames for MOT17-14 according to the length of the videos. However, these parameter settings require pre-processing of the video, sine the time of disappearing trajectory varies due to different occlusion situations. It would be impractical to make different settings manually for different datasets. From a practical perspective, a unified trajectory

 doi:10.20944/preprints202510.2145.v1

processing scheme should be developed that is applicable to different datasets, while different time intervals should be assigned to different kinds of lost trajectories.

3. Method

The block diagram of our tracker is drawn in Figure 1. Basically, CS is incorporated into the tracking process, where the detection boxes with either high or low scores are taken care of by the first and the second association processes respectively to ensure correct associations of trajectories.

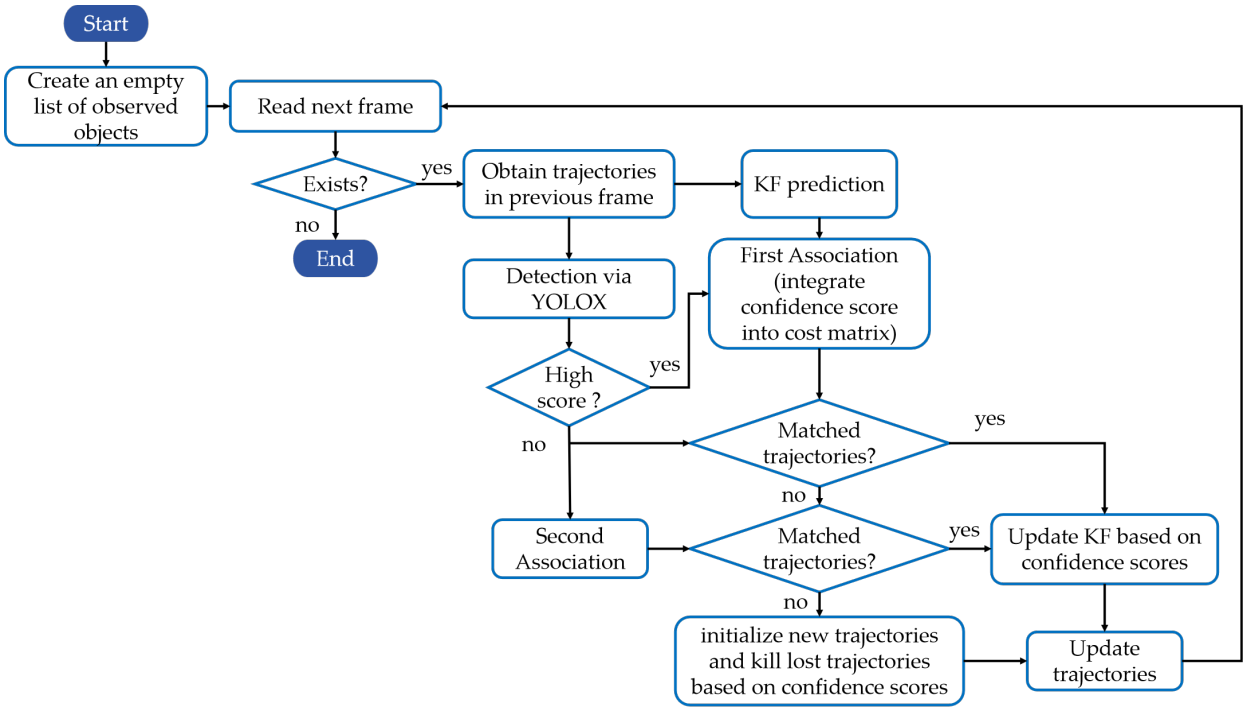

Figure 1. Block diagram of the proposed **ConfMOT** framework.

3.1. Confidence-Based Adaptive KF

KF (with a constant velocity) is commonly adopted in object tracking tasks in building a motion model. It follows the Markov assumption that the current system state is only related to the state of its previous moment, which is irrelevant to any previous states. Then the previous state vector and the state observed at the current frame are needed to estimate the current state.

KF consists of a prediction step and an update step. In the first step, KF generates the estimated state variables (along with their uncertainties), which will then be updated with a weighted average of the estimated state and measurement. Since there is no active control in multi-object tracking, the state vector $\hat{x}_{t-1|t-1}$ at frame $t-1$ is represented by:

$$\hat{x}_{t-1|t-1} = [\hat{u}_{t-1|t-1}, \hat{v}_{t-1|t-1}, a, \hat{h}_{t-1|t-1}, \dot{u}, \dot{v}, \dot{a}, \dot{h}]^T, \tag{1}$$

The update process of KF is expressed as:

$$\begin{aligned} K_t &= P_{t|t-1}H^T(HP_{t|t-1}H^T + (1-c_t)R_t)^{-1}, \\ \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t(z_t - H\hat{x}_{t|t-1}), \\ P_{t|t} &= P_{t|t-1} - K_tHP_{t|t-1}, \end{aligned} \tag{2}$$

$$H = \begin{bmatrix} I_{4\times4} & O_{4\times4} \end{bmatrix}, \tag{3}$$

 doi:10.20944/preprints202510.2145.v1

$$z_t = \left[z_x, z_y, z_w, z_h\right]^T, \tag{4}$$

where K_t is the Kalman gain and H is the observation matrix. c_t is the detection CS and R_t is the covariance of the measurement noise, which represents the level of measurement noise (of the detections in the current frame). A higher noise level indicates a lower weight of the measurement during the state update (i.e., higher uncertainty). z_t represents the observation state vector.

As can be seen, KF estimates the current state vector of the trajectory and updates it through R_t. In many previous works [14,21], R is a constant matrix (i.e. the value of the measurement noise remains constant for different detection results). Intuitively, different measurements should contain different levels of noise, and the measurement noise should also be adaptive to detection confidence. In this paper, we use CS to balance the measurement noise of different scales [16]. Specifically, we rectify the constant R_t into $(1 - c_t)R_t$ in Eqn. (2), so that a higher CS will result in lower noise (i.e., more reliable detection results). Conversely, a lower $(1 - c_t)R_t$ means that the detection will be assigned higher weight in the state update step.

3.2. Data Association

The trajectory of frame t is predicted by the trajectory of frame $t - 1$, which is used to calculate IoU with the detection of frame t. Then the IoU is used to construct CM. Then the trajectories will be associated with the detection boxes via Hungarian algorithm based on the CM and be updated via KF. The detection boxes with low confidence scores are regarded as unreliable, which should have higher cost for successful matching.

For ConfMOT, we use CS to rectify CM: the lower the confidence level of the detection, the higher the value of the corresponding element in the matrix will be. Formally:

$$C_{i,j} = 1 - \frac{1 + S_j}{2} \times (1 - c_{i,j}^{iou}), \tag{5}$$

where $C_{i,j}$ represents the confidence score of the j^{th} detection box and $c_{i,j}^{iou}$ represents the IoU cost between the i^{th} prediction box and the j^{th} detection box.

3.3. Deletion Strategy

In previous methods, the lost trajectories were usually kept for a certain number of frames, which may be recovered if matches were found, or otherwise deleted. However, for objects with lower CS, their trajectories (or recovered trajectories) are less reliable. We use the track score to define the reliability of a trajectory by summing up the CS of an object on its last tracklet. For a tracklet that lasts for n frames, its track score is expressed as:

$$S_{track} = \sum_{i=1}^{n} S_i, \tag{6}$$

where S_i represents the CS of an object in the i^{th} frame of the traklet. The lower the track score, the more unstable and less reliable the trajectory will be, which will be endowed shorter time of duration before it is labeled as lost. For the lost trajectories, we use the following equation to determine the time to delete them:

$$S_{\text{del}} = \min\{\alpha S_{\text{track}} + S_{\text{last}}, 1\} - \log(1 + \beta T_{\text{umatch}}), \tag{7}$$

where S_{last} is the confidence score of the last frame of the tracklet and T_{umatch} is the number of frames when the trajectory does not find a match in the following frames. α and β are two weights. A trajectory will be deleted if its confidence score S_{del} is lower than a certain threshold. To avoid a reliable trajectory with high confidence score but short duration from being deleted in an early stage (e.g. a newly generated but soon occluded trajectory), we sum the confidence score of the last frame with the track score to ensure that the track score is not too low. Meanwhile, different trajectories should

 doi:10.20944/preprints202510.2145.v1

have different time durations before deletion, so as to reduce the possibility of object drifting to other unreliable trajectories that last for long time.

3.4. Tracking Procedure

The pseudo-code of tracking is shown in Algorithm 1, and the corresponding diagram has been drawn in Figure 2 to better understand the process. Firstly, we adopt YOLOX as our detector to obtain the bounding boxes and CS, where the detection boxes are divided into 2 parts: $\mathcal{D}_{high}$ (yellow in Figure 2) and $\mathcal{D}_{low}$ (green in Figure 2) based on confidence thresholds τ_{high} and τ_{low}(which are set to 0.6 and 0.1 respectively according to OC-SORT[14]). The bad detection boxes with CS lower than τ_{low} are removed. Then we use KF to predict the new locations of each trajectory in $\mathcal{T}_{all}$ (labeled 1, 2, 3 in Figure 2).

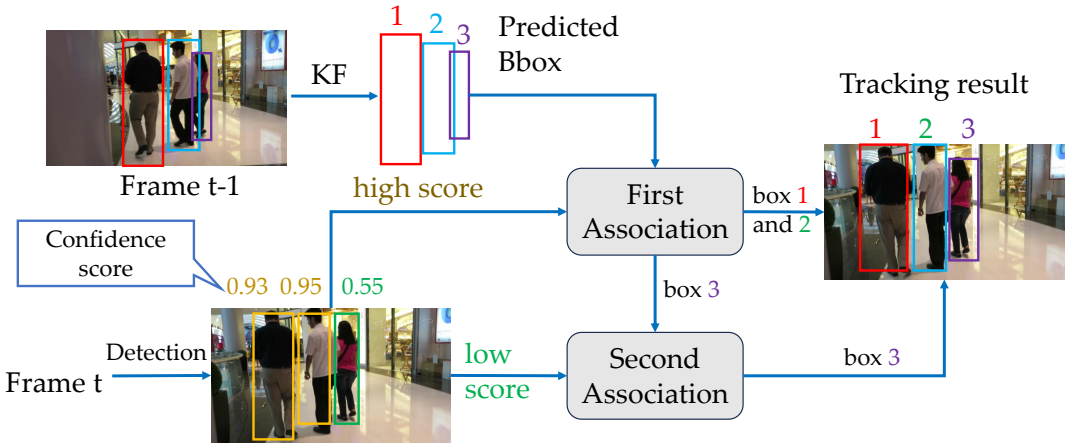

Figure 2. Illustration of the processing flow of the proposed method.

The first association is performed between the high score boxes $\mathcal{D}_{high}$ and all trajectories $\mathcal{T}_{all}$. We only use the IoU CM via Eqn. (5) and utilize Hungarian algorithm to assign the detection boxes to corresponding trajectories. In particular, if the IoU cost between the detection box and the prediction box is greater than a threshold, then the matching will be rejected. The unmatched detections in the first association are kept in $\mathcal{D}_{remain}$ and the unmatched trajectories (e.g. No. 3 in Figure 2) are kept in $\mathcal{T}_{remain}$ (line 14 to 18 in Algorithm 1), which will be matched with the low score boxes in the second stage association.

In the second association, $\mathcal{T}_{remain}$ will be associated with the low score boxes $\mathcal{D}_{low}$. The unmatched detection boxes in $\mathcal{D}_{low}$ are regarded as the background and unmatched trajectories are marked as lost and kept in $\mathcal{T}_{lost}$(line 19 to 21 in Algorithm 1), which will be deleted after a certain period of time.

The successfully matched trajectories are updated by KF through Eqn. (2). For trajectories in $\mathcal{T}_{lost}$, their CS ($\mathcal{S}_{del}$) will be calculated via Eqn. (7). If they are lower than τ_{del}, they will be moved from $\mathcal{T}_{lost}$ to $\mathcal{T}_{del}$. If a lost trajectory is recovered, it will be moved from $\mathcal{T}_{lost}$ to $\mathcal{T}$, and the track score will be recalculated.

Finally, for detection boxes in $\mathcal{D}_{remain}$, we initialize new trajectories if their CS are higher than τ_{new} and exist for 2 consecutive frames(line 29 to 31 in Algorithm 1).

4. Experiment

4.1. Datasets

The training process of MOT17 [12] is conducted jointly on CrowHuman [22], CityPersons [23], ETHZ [24] and the training set of MOT17. And we use CrowdHuman and the training set of MOT20 [13] to train MOT20. While testing is carried out on MOT17 and MOT20 under the “private detection" protocol. For ablation study, we combine CrowdHuman and half of the training set of MOT17 as the

 doi:10.20944/preprints202510.2145.v1

Algorithm 1: Pseudo-code of tracking

Input: A video sequence V; object detector Det; detection score threshold τ_{high}, τ_{low}; new trajectory score threshold τ_{new}; trajectory confidence score threshold τ_{del}
Output: Tracks $\mathcal{T}$ of the video

Initialization: $\mathcal{T}, \mathcal{T}_{lost} \leftarrow \emptyset$
for *frame* f_t *in* V **do**
  /* Handle new detections */
  $\mathcal{D}_t \leftarrow \texttt{Det}(f_t)$
  $\mathcal{D}_{high} \leftarrow \emptyset$
  $\mathcal{D}_{low} \leftarrow \emptyset$
  for d *in* $\mathcal{D}_t$ **do**
    if $d.score > \tau_{high}$ **then**
      /* Store high scores detections */
      $\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}$
    else if $d.score > \tau_{low}$ **then**
      /* Store low scores detections */
      $\mathcal{D}_{low} \leftarrow \mathcal{D}_{low} \cup \{d\}$
  $\mathcal{T}_{all} \leftarrow \mathcal{T} \cup \mathcal{T}_{lost}$
  /* Predict new locations of trajectories */
  for t *in* $\mathcal{T}_{all}$ **do**
    $t \leftarrow \texttt{KalmanFilter}(t)$
  /* First association */
  $C_{iou} \leftarrow IoU(\mathcal{T}_{all}, \mathcal{D}_{high})$
  $C_{final} \leftarrow \mathcal{D}_{high}.score$ with C_{iou} //Eq.(5)
  Linear assignment by Hungarian's alg. with C_{final}
  $\mathcal{D}_{remain} \leftarrow$ remaining object boxes from $\mathcal{D}_{high}$
  $\mathcal{T}_{remain} \leftarrow$ remaining trajectories from $\mathcal{T}_{all}$
  /* Second association */
  $C_{iou} \leftarrow IoU(\mathcal{T}_{remain}, \mathcal{D}_{low})$
  Linear assignment by Hungarian's alg. with C_{iou}
  $\mathcal{T}_{lost} \leftarrow$ remaining trajectories from $\mathcal{T}_{remain}$
  /* Update matched trajectories */
  $\mathcal{T}_{all} \leftarrow KalmanFilter(\mathcal{T}_{all}, \mathcal{D}_t.score)$// Eq.(2)
  Update tracklets appearance features.
  /* Delete unmatched trajectories */
  for t *in* $\mathcal{T}_{lost}$ **do**
    if $t.score < \tau_{del}$ **then**
      $\mathcal{T}_{del} \leftarrow \mathcal{T}_{del} \cup \{t\}$// Eq.(7)
  $\mathcal{T}_{lost} \leftarrow \mathcal{T}_{lost} \setminus \mathcal{T}_{del}$
  $\mathcal{T} \leftarrow \mathcal{T}_{all} \setminus \mathcal{T}_{lost}$
  /* Initialize new trajectories */
  for d *in* $\mathcal{D}_{remain}$ **do**
    if $d.score > \tau_{new}$ **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$
Return: $\mathcal{T}$

Note: Trajectory recovery process is included in the matching process of lost trajectories.

training set and test our tracker on the second half of MOT17. The initial parameters of our model is pre-trained on COCO [25].

4.2. Metrics

We adopt Higher Order Tracking Accuracy (HOTA) Multi-Object Tracking Accuracy (MOTA), ID F1 Score (IDF1) (the ratio of correctly identified detections over the average number of ground-truth and computed detections) and Number of Identity Switches (IDS) as the main metrics to evaluate the performance of our model. Considering MOTA is mainly used to assess detection performance while IDF1 characterizes the ability to keep constant identities, we use HOTA as the final evaluation metric

 doi:10.20944/preprints202510.2145.v1

for ranking, which is a more comprehensive indicator to reflect the general performance of detection accuracy, association and localization. In addition, we adopt Association Accuracy (AssA), Association Precision (AssPr), Association Recall (AssRe), Detection Accuracy (DetA) and Localization Accuracy (LocA) to further compare the trackers with similar performances. AssA, AssPr and AssRe are used to measure association performance while DetA for detection quality.

4.3. Implementation Details

All the experiments are implemented in PyTorch and under NVIDIA Tesla V100 GPU. We use YOLOX as the detector with YOLOX as the backbone. The datasets and training strategy are described in Section 4.1. The optimizer is SGD with weight decay of 5×10^{-4} and a momentum of 0.9 and the initial learning rate is set to 10^{-3}. The parameters of the tracker are the same as the baseline. The high and low confidence threshold τ_{high} and τ_{low} are empirically set to 0.6 and 0.1 respectively. If the CS of unmatched detections are higher than 0.7[2], we initialize new trajectories starting from them. The lost trajectories will be deleted if their CS are lower than 0.1. GSI is adopted as post-processing method.

4.4. Experimental Results

Extensive experiments have been conducted on MOT17 [12] and MOT20 [13] to testify the effectiveness of our tracker.

4.4.1. Resutls on MOT17

The video sequences in MOT17 are filmed by both static and moving cameras. The comparative results on MOT17 are listed in Table 1. Generally, ByteTrack [2], OC-SORT [14],SCTrack [43] etc. and ConfMOT outperform other trackers with some margins. Among them, ConfMOT surpass ByteTrack by 1.4% and 2.0% in HOTA and MOTA respectively, with lower IDS as well. Meanwhile, we lead OC-SORT in HOTA, MOTA and IDF1. Our obvious advantage in HOTA is attributed to the proposed early trajectory deletion strategy which ensures the tracking accuracy.

 doi:10.20944/preprints202510.2145.v1

Table 1. Evaluation on the test sets of MOT17 and MOT20. We compare our method with recent methods. The best results are shown in red and the second best results are in blue.

Methods	MOT17					MOT20				
	HOTA↑	MOTA↑	IDF1↑	IDS↓	FPS↑	HOTA↑	MOTA↑	IDF1↑	IDS↓	FPS↑
TADN[44]	-	69.0%	60.8%	-	-	-	68.7%	61.0%	-	-
DMMTracker[45]	52.1%	67.1%	64.3%	3135	16.1	48.7%	62.5%	60.5%	2043	9.7
TransTrack[26]	54.1%	75.2%	63.5%	3603	10.0	48.5%	65.0%	59.4%	3608	7.2
TransCenter[7]	54.5%	73.2%	62.2%	4614	1.0	43.5%	61.9%	50.4%	4653	1.0
MeMOT[27]	56.9%	72.5%	69.0%	2724	-	54.1%	63.7%	66.1%	1938	-
AMtrack[8]	58.6%	74.4%	71.5%	4740	-	56.8%	73.2%	69.2%	1870	-
DNMOT[9]	58.0%	75.6%	68.1%	2529	-	58.6%	70.5%	73.2%	987	-
MeMOTR[31]	58.8%	72.8%	71.5%	-	-	-	-	-	-	-
FairMOT[21]	59.3%	73.7%	72.3%	3303	25.9	54.6%	61.8%	67.3%	5243	13.2
DiffusionTrack[32]	60.8%	77.9%	73.8%	3819	-	55.3%	72.8%	66.3%	4117	-
STDFormer-LMPH[33]	60.9%	78.4%	73.1%	5091	-	60.2%	76.2%	72.1%	5245	-
RelationTrack[34]	61.0%	73.8%	74.4%	1374	7.4	56.5%	67.2%	70.5%	4243	2.7
BGTracker[46]	61.0%	75.6%	73.8%	3735	20.7	57.5%	71.6%	71.8%	2471	12.8
ColTrack[35]	61.0%	78.8%	73.9%	1881	-	-	-	-	-	-
JDT-NAS-T1[36]	-	74.3%	72.0%	2818	13.3	-	-	-	-	-
DcMOT[37]	61.3%	74.5%	75.2%	2682	20.4	53.8%	59.7%	67.4%	5636	10.6
MOTFR[28]	61.8%	74.4%	76.3%	2652	22.2	57.2%	69.0%	71.7%	3648	13.3
CorrTracker[38]	-	76.5%	73.6%	3369	14.8	-	65.2%	69.1%	5183	8.5
TransMOT[39]	-	76.7%	75.1%	2346	-	-	77.5%	75.2%	1615	-
MAA[29]	62.0%	79.4%	75.9%	1452	189.1	57.3%	73.9%	71.2%	1331	14.7
MOTRv2[3]	62.0%	78.6%	75.0%	-	-	61.0%	76.2%	73.1%	-	-
PID-MOT[40]	62.1%	74.7%	76.3%	1563	19.7	57.0%	67.5%	71.3%	1015	8.7
GHOST[41]	62.8%	78.7%	77.1%	2325	-	61.2%	73.7%	75.2%	1264	-
GGSTrack[30]	62.8%	80.2%	-	1689	58.0	61.8%	75.1%	-	1498	15.3
ScoreMOT[42]	63.0%	79.8%	76.7%	4007	25.6	62.3%	77.7%	75.6%	1440	16.2
ByteTrack[2]	63.1%	80.3%	77.3%	2196	29.6	61.3%	77.8%	75.2%	1223	17.5
OC-SORT[14]	63.2%	78.0%	77.5%	1950	29.0	62.1%	75.5%	75.9%	913	18.7
AM-SORT[47]	63.3%	78.0%	77.8%	-	-	62.0%	75.5%	76.1%	-	-
SCTrack[43]	63.5%	79.4%	77.7%	2022	-	61.4%	75.6%	76.1%	837	-
ConfMOT	64.5%	80.5%	79.3%	1980	26.1	62.9%	78.3%	76.1%	1359	15.2

To further investigate the subtle difference among ByteTrack [2], OC-SORT [14] and our tracker, we compare another 5 metrics: AssA, AssPr, AssRe, DetA and LocA. As mentioned in Section 4.2, the first 3 metrics reflect the data association ability of a tracker, while DetA and LocA demonstrate the accuracies of object detection and localization respectively. As shown in Table 2, except that we lag behind OC-SORT in AssPr, ConfMOT outperforms other 2 trackers in all other metrics. The reason is that OC-SORT is stronger than our trackers in association ability, but weaker in detection. It is worth noting that ConfMOT ranks first in AssRe, which reflects the accurate prediction of the target trajectories.

Table 2. Further comparison of different tracking methods on the MOT17 dataset. The best results are highlighted in red and the second-best results are in blue

Tracker	AssA (%)↑	AssPr (%)↑	AssRe (%)↑	DetA (%)↑	LocA (%)↑
ByteTrack [2]	62.0	76.0	68.2	64.5	83.0
OC-SORT [14]	63.4	80.8	67.5	63.2	83.4
ConfMOT (Ours)	63.8	77.9	70.0	64.9	83.4

A group of visualization results of ConfMOT on the test set of MOT17 is shown in Figure 3. MOT17-01 scene is a normal outdoor scenario with pedestrians walking around. Apparently, the color of the bounding boxes do not either change or swap, indicating constant identities during tracking process. MOT17-03 is a crowded outdoor scene, in which our tracker still completes tracking task without obvious miss detections or ID switches. MOT17-06 is a normal street scene, it's worth noting that the old man with number 27 dark blue box (in the middle of the scene) always remains unchanged after several overlap by other pedestrians, which indicates the ability of our tracker in dealing with occlusions. In MOT17-07, the video is taken by moving cameras, and there is a man sitting at the

 doi:10.20944/preprints202510.2145.v1

left corner in the first frame, who is moving closer in the following frames with a constant ID. The MOT17-8 shows a tracking scenario with varying lighting conditions. Still, our tracker demonstrates robustness in tracking multiple targets with fixed IDS. Lastly, MOT17-14 is a video sequence containing small targets which is captured on a moving box. Clearly, we can correctly detect and locate the small targets over moving viewpoint.

Figure 3. Demonstration of tracking results on the test set of MOT17.

4.4.2. Results on MOT20

MOT20 is a more challenging benchmark with crowded scenes. As shown in Table 1, ConfMOT ranks first in HOTA, MOTA and IDF1 due to our proposed deletion strategies, but it also causes slightly higher IDS, since the lost tracks are difficult to be matched. Like MOT17, we also compare AssA, AssPr, AssRe, DetA and LocA among OC-SORT, ByteTrack and ConfMOT. As shown in Table 3,

 doi:10.20944/preprints202510.2145.v1

ConfMOT surpasses both OC-SORT and ByteTrack in all 5 metrics, which reflects stronger association accuracy and more precise localization. The deletion of unreliable lost trajectories in the early stage ensures the overall tracking performances, while the integration of the confidence-based cost matrix guarantees the robustness of the tracker during long period of tracking process.

Table 3. Further comparison with ByteTrack on the testing set of MOT20. The best results are shown in **bold**.

Tracker	AssA↑	AssPr↑	AssRe↑	DetA↑	LocA↑
ByteTrack[2]	59.6%	74.6%	66.2%	63.4%	83.6%
OC-SORT[14]	60.5%	75.1%	67.1%	64.2%	83.9%
ConfMOT	**61.4**%	**77.1**%	**67.8**%	**64.6**%	**84.7**%

A group of visualization results of ConfMOT on the test set of MOT20 is shown in Figure 4, including indoor and outdoor scenarios under different lighting conditions. As mentioned earlier, MOT20 include crowded scenes. MOT20-04 is an outdoor scene at night, while MOT20-06 and MOT20-08 are daytime scenarios. No matter under which scenes, our tracker maintains excellent tracking performances, which is less sensitive to lighting conditions, and the ID of each target remain largely unchanged.

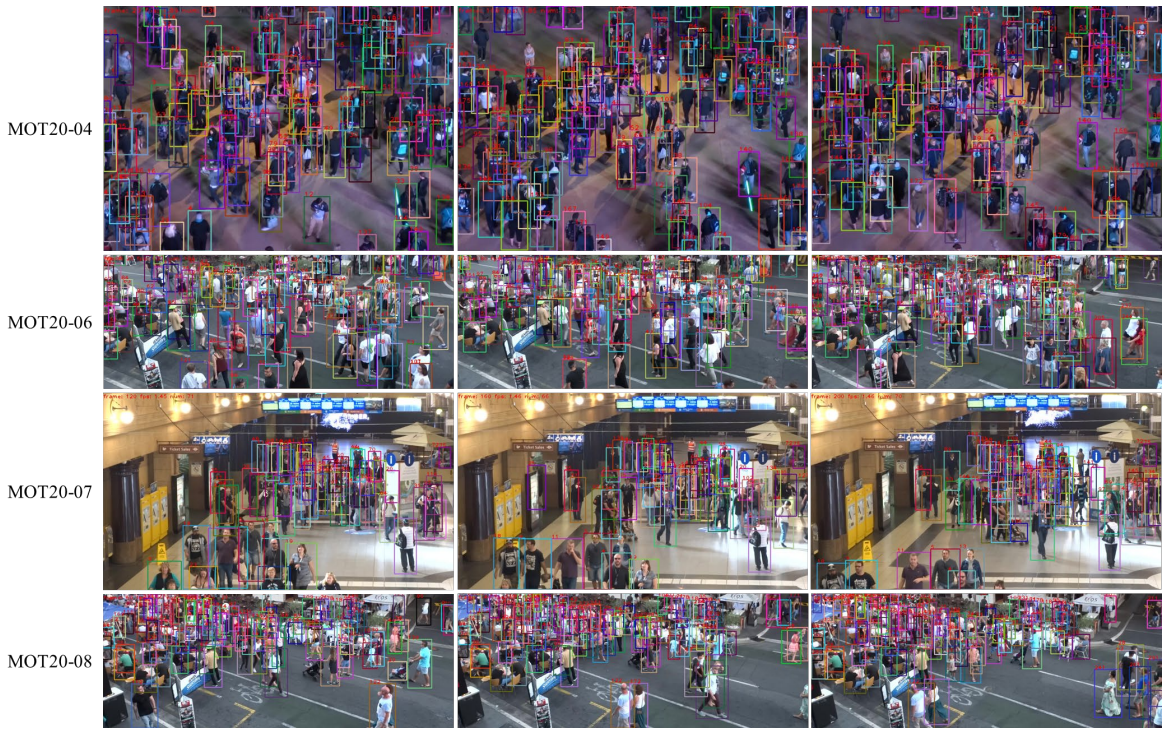

Figure 4. Demonstration of tracking results on the test set of MOT20.

4.4.3. Comprehensive Analysis

As mentioned earlier, ByteTrack [2], OC-SORT [14] and our tracker achieve better results than other competitors. Here, we further analyze the advantages / disadvantages of the 3 methods. ByteTrack [2] utilizes pure IoU to process all the detections. In contrast, we process the detection boxes using the cost matrix which is adjusted by the confidence score. Our scheme matches the high-quality detection boxes in a more effective way, resulting in more stable assignment results. As a result, we achieve the best overall secondary indexes (including AssA, AssPr, AssRe, DetA and LocA). Besides, we obtain more precise object location due to the proposed modified KF, which has been verified by the highest DetA (Detection Accuracy) we have reached. In summary, we have achieved the best overall results on MOT-17 and on MOT-20.

 doi:10.20944/preprints202510.2145.v1

4.4.4. Visualization of Trajectories

To further demonstrate the superiority of our tracker, a comparison among ByteTrack [2], OC-SORT [14] and ConfMOT is made in Figure 5, in which the trajectory of each object of the last 50 frames is drawn with different colors. Obviously, OC-SORT (shown in Figure 5(a)) has many sporadic segments of trajectories, which reflect target loss under crowded scene, while it has almost no trajectories at the left hand side of the scene. ByteTrack (shown in Figure 5(b)) has more trajectories, but there are some zigzags and stagnation, indicating incorrect matching and ID switches. In comparison, as shown in Figure 5(c), the trajectories of ConfMOT appear more stable, with almost identical trajectory lengths and no broken segments. In addition, the colors of the trajectories do not change within the dense areas, indicating the robustness of ConfMOT under crowded scenes.

4.5. Ablation Studies

In ablation experiment, we use YOLOX as the backbone and train our model jointly on CrowdHuman and the first half of the training set of MOT17. While the second half of the training set of MOT17 is used for validation. In this section, we will firstly verify the effectiveness of the core modules of our tracker, including KF, CM+CS and deletion strategy (DS). The comparative results on MOT17 validation set is shown in Table 4.

Table 4. Ablation study on the MOT17 validation set. The best results are shown in **bold**.

KF	DS	CM+CS	HOTA↑	MOTA↑	IDF1↑
			67.8%	77.9%	79.6%
✓			68.1%	77.9%	79.8%
✓	✓		68.3%	78.0%	80.2%
✓	✓	✓	**68.7%**	**78.4%**	**80.5%**

(**a**) OC-SORT (**b**) ByteTrack (**c**) ConfMOT

Figure 5. Comparison of trajectories generated by three tracking methods.

4.5.1. KF Based on the Confidence Score

As shown in Table 4, when KF (with CS) is integrated into the tracking network, we enjoy a slight increase in both HOTA and IDF1. By incorporating confidence, KF assigns weights to different detections based on their CS, making it more robust to noise and uncertain detections. During the update of KF, the high quality detection boxes will be assigned higher weights than the prediction boxes, making the detection boxes more dependent on the detection results. Reversely, for low quality detection results, the detection boxes will depend more on the prediction boxes, which will be assigned higher weights.

4.5.2. Deletion Strategy based on the confidence score

As shown in Table 4, when the proposed lost trajectory deletion strategy has been added into the tracking architecture, the HOTA and IDF1 value increase by 0.2% and 0.4%, respectively. This results illustrate the fact that when a lost trajectory has been deleted, the subsequent detected target cannot find a match and therefore a new trajectory will be created. Moreover, the increased IDF1 suggests that the deleted trajectories are indeed unreliable, and our deletion strategy reduces the likelihood of incorrect matches between a detected target and the wrong trajectory. Intuitively, if a trajectory

 doi:10.20944/preprints202510.2145.v1

remains unmatched for a certain period of time and is re-matched again, it's quite possible that it matches with a different target (instead of the previous one). Therefore, our deletion strategy becomes very necessary to avoid target drifting caused by re-matching.

4.5.3. Cost matrix based on the confidence score

When calculating the matching cost between the prediction and detection boxes, such cost will be higher for low-quality detection boxes. As shown in Table 4, after the CS incorporated into CM, both HOTA and MOTA increase by 0.4%. This result indicates that the integration of confidence score indeed reduces the probability of matching low-quality detection boxes and continuation of incorrect trajectory.

5. Conclusion

In this paper, we thoroughly analyze the limitations of current TBD based trackers, and present ConfMOT to ensure stable continuation of trajectories among different objects. To realize the goal, we adopt YOLOX as our detector to obtain the detection results. We adopt two-stage association strategy and predict the trajectories based on the confidence-score and Kalman filtering technique. Then a cost matrix has been constructed to measure the cost of unreliable matching. Finally, a deletion strategy has been proposed to determine the duration of a trajectory and delete the less reliable trajectories. Extensive experiments have been conducted on MOT17 and MOT20 datasets along with ablation studies to testify the efficacy of our tracker. The results prove that our trackers have obvious advantages in the HOTA metric compared to other methods, due to the cost matrix and trajectory deletion strategy we designed. However, the performance of our tracker in IDS is mediocre (ID switch caused by occlusion), since we delete unreliable trajectories in an early stage.

In future work, we will be engaged in developing light-weight appearance feature extraction module to improve the robustness of the model in dealing with fast appearance changes. We will also employ specific motion modelling schemes to ensure the stability and continuation of target trajectory.

Funding: This research was funded by the Intelligent Policing Key Laboratory of Sichuan Province, No. ZNJW2024KFMS004

References

1. N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE international conference on image processing (ICIP), IEEE, 2017, pp. 3645–3649.
2. Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang, Bytetrack: Multi-object tracking by associating every detection box, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII, Springer, 2022, pp. 1–21.
3. Y. Zhang, T. Wang, X. Zhang, Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 22056–22065.
4. A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE international conference on image processing (ICIP), IEEE, 2016, pp. 3464–3468.
5. Lifan Sun, Jiayi Zhang, Dan Gao, Bo Fan, Zhumu Fu. Occlusion-aware visual object tracking based on multi-template updating Siamese network. Digital Signal Processing. 148 (2024) 104440
6. H. Li, Y. Liu, X. Liang, Y. Yuan, Y. Cheng, G. Zhang, S. Tamura, Multi-object tracking via deep feature fusion and association analysis, Engineering Applications of Artificial Intelligence 124 (2023) 106527.
7. Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, X. Alameda-Pineda, Transcenter: Transformers with dense representations for multiple-object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).
8. Z. Liu, X. Huang, J. Sun, X. Zhang, AMtrack: Anti-occlusion multi-object tracking algorithm, *Signal, Image and Video Processing*, vol. 18, no. 12, pp. 9305–9318, 2024.
9. T. Fu, X. Wang, H. Yu, K. Niu, B. Li, X. Xue, Denoising-mot: Towards multiple object tracking with severe occlusions, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 2734–2743.

 doi:10.20944/preprints202510.2145.v1

10. Z. Wang, L. Zheng, Y. Liu, Y. Li, S. Wang, Towards real-time multi-object tracking, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, Springer, 2020, pp. 107–122.
11. Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, arXiv preprint arXiv:2107.08430 (2021).
12. A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler, Mot16: A benchmark for multi-object tracking, arXiv preprint arXiv:1603.00831 (2016).
13. P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, L. Leal-Taixé, Mot20: A benchmark for multi object tracking in crowded scenes, arXiv preprint arXiv:2003.09003 (2020).
14. J. Cao, J. Pang, X. Weng, R. Khirodkar, K. Kitani, Observation-centric sort: Rethinking sort for robust multi-object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9686–9696.
15. S. You, H. Yao, C. Xu, Multi-object tracking with spatial-temporal topology-based detector, IEEE Transactions on Circuits and Systems for Video Technology 32 (5) (2021) 3023–3035.
16. Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, J. Dong, Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2809–2819.
17. T. Khurana, A. Dave, D. Ramanan, Detecting invisible people, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 3174–3184.
18. G. Wang, R. Gu, Z. Liu, W. Hu, M. Song, J.-N. Hwang, Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9876–9886.
19. P. Bergmann, T. Meinhardt, L. Leal-Taixe, Tracking without bells and whistles, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 941–951.
20. L. Chen, H. Ai, Z. Zhuang, C. Shang, Real-time multiple people tracking with deeply learned candidate selection and person re-identification, in: 2018 IEEE international conference on multimedia and expo (ICME), IEEE, 2018, pp. 1–6.
21. Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, Fairmot: On the fairness of detection and re-identification in multiple object tracking, International Journal of Computer Vision 129 (2021) 3069–3087.
22. S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, J. Sun, Crowdhuman: A benchmark for detecting human in a crowd, arXiv preprint arXiv:1805.00123 (2018).
23. S. Zhang, R. Benenson, B. Schiele, Citypersons: A diverse dataset for pedestrian detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3213–3221.
24. A. Ess, B. Leibe, K. Schindler, L. Van Gool, A mobile vision system for robust multi-person tracking, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
25. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.
26. P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, P. Luo, Transtrack: Multiple object tracking with transformer, arXiv preprint arXiv:2012.15460 (2020).
27. J. Cai, M. Xu, W. Li, Y. Xiong, W. Xia, Z. Tu, S. Soatto, Memot: multi-object tracking with memory, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8090–8100.
28. J. Kong, E. Mo, M. Jiang, T. Liu, Motfr: Multiple object tracking based on feature recoding, IEEE Transactions on Circuits and Systems for Video Technology 32 (11) (2022) 7746–7757.
29. P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, P. Luo, Transtrack: Multiple object tracking with transformer, arXiv preprint arXiv:2012.15460 (2020).
30. S. Yan, Z. Wang, Y. Huang, Y. Liu, Z. Liu, F. Yang, W. Lu, D. Li, GGSTrack: Geometric graph with spatio-temporal convolution for multi-object tracking, *Neurocomputing*, 2025, p. 131234.
31. R. Gao, L. Wang, Memotr: Long-term memory-augmented transformer for multi-object tracking, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 9901–9910.
32. R. Luo, Z. Song, L. Ma, J. Wei, W. Yang, M. Yang, Diffusiontrack: Diffusion model for multi-object tracking, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 3991–3999.
33. M. Hu, X. Zhu, H. Wang, S. Cao, C. Liu, Q. Song, Stdformer: Spatial-temporal motion transformer for multiple object tracking, IEEE Transactions on Circuits and Systems for Video Technology 33 (11) (2023) 6571–6594.

 doi:10.20944/preprints202510.2145.v1

34. E. Yu, Z. Li, S. Han, H. Wang, Relationtrack: Relation-aware multiple object tracking with decoupled representation, IEEE Transactions on Multimedia 25 (2022) 2686–2697.
35. Y. Liu, J. Wu, Y. Fu, Collaborative tracking learning for frame-rate-insensitive multi-object tracking, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 9964–9973.
36. D. Chen, H. Shen, Y. Shen, Jdt-nas: Designing efficient multi-object tracking architectures for non-gpu computers, IEEE Transactions on Circuits and Systems for Video Technology 33 (12) (2023) 7541–7553.
37. K. Deng, C. Zhang, Z. Chen, W. Hu, B. Li, F. Lu, Jointing recurrent across-channel and spatial attention for multi-object tracking with block-erasing data augmentation, IEEE Transactions on Circuits and Systems for Video Technology 33 (8) (2023) 4054–4069.
38. Q. Wang, Y. Zheng, P. Pan, Y. Xu, Multiple object tracking with correlation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3876–3886.
39. P. Chu, J. Wang, Q. You, H. Ling, Z. Liu, Transmot: Spatial-temporal graph transformer for multiple object tracking, in: Proceedings of the IEEE/CVF Winter Conference on applications of computer vision, 2023, pp. 4870–4880.
40. W. Lv, N. Zhang, J. Zhang, D. Zeng, One-shot multiple object tracking with robust id preservation, IEEE Transactions on Circuits and Systems for Video Technology (2023).
41. J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi, L. Leal-Taixé, Simple cues lead to a strong multi-object tracker, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 13813–13823.
42. T. Zhao, G. Yang, Y. Li, M. Lu, and H. Sun, "Multi-object tracking using score-driven hierarchical association strategy between predicted tracklets and objects," *Image and Vision Computing*, vol. 152, p. 105303, 2024.
43. H. Li, S. Qin, S. Li, Y. Gao, and Y. Wu, "Synergistic-aware cascaded association and trajectory refinement for multi-object tracking," *Image and Vision Computing*, p. 105695, 2025.
44. A. Psalta, V. Tsironis, and K. Karantzalos, "Transformer-based assignment decision network for multiple object tracking," *Computer Vision and Image Understanding*, vol. 241, p. 103957, 2024.
45. Y.-F. Li, H.-B. Ji, W.-B. Zhang, and Y.-K. Lai, "Learning discriminative motion models for multiple object tracking," *IEEE Transactions on Multimedia*, 2024.
46. C. Zhou, M. Jiang, and J. Kong, "Bgtracker: cross-task bidirectional guidance strategy for multiple object tracking," *IEEE Transactions on Multimedia*, vol. 25, pp. 8132–8144, 2023.
47. V. Kim, G. Jung, and S.-W. Lee, "AM-SORT: adaptable motion predictor with historical trajectory embedding for multi-object tracking," in *Proc. Int. Conf. Pattern Recognition and Artificial Intelligence*, pp. 92–107, 2024.