2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

GroupTrack: Multi-Object Tracking by Using Group Motion Patterns

Xinglong Xu, Weihong Ren, Gan Sun, Haoyu Ji, Yu Gao, and Honghai Liu, *Fellow, IEEE*

***Abstract*— The main challenge of Multi-Object Tracking (MOT) lies in maintaining a distinctive identity for each target in dense crowds or occluded scenarios. Although the existing methods have achieved significantly progress by using robust object detectors or complex association strategies, they cannot effectively solve long-term tracking due to individually motion or appearance modeling for each single target. In this paper, we propose a novel 2D MOT tracker GroupTrack, to learn reliable motion state for each target using group motion patterns. Specifically, for each tracklet, we first choose its neighboring ones to form a group of motion patterns, which can provide informative clues for the motion estimation of the current tracklet. Then, we apply the group motion patterns to perform tracklet prediction and data association. By integrating prior from neighboring motion patterns into the data association process, GroupTrack provides a new paradigm for target motion modeling in extremely crowded and occluded scenarios. Through extensive experiments on the public MOT17 and MOT20 datasets, we demonstrate the effectiveness of our approach in challenging scenarios and show state-of-the-art performance at various MOT metrics.**

I. INTRODUCTION

Multi-Object Tracking (MOT) [1] is an essential task in the field of machine vision, and it aims to maintain a continuous trajectory for each detected target across a video sequence. MOT has widespread applications in various domains such as autonomous driving [2], [3], robot perception [4], and intelligent surveillance [5]. Although MOT has achieved impressive progress in recent years, it still faces challenges in dealing with dense crowds and extreme occlusion in complex scenarios.

Currently, the paradigm of tracking-by-detection has emerged as the most effective approach for MOT task. It first localizes all the objects within each video frame and then establishes trajectories across different frames by performing data association. With the great progress of object detection [6]–[12], recent MOT methods adopt powerful object detectors to achieve high tracking performance. However, in dense crowds or occluded scenarios, they cannot effectively solve long-term tracking due to individually motion or appearance modeling for each single target. As shown in Fig. 1 (a), the pedestrian in green box is heavily occluded by the one in orange box, and they have a high Intersection-over-Union (IoU). After updating the tracklets of the two pedestrians, their motion models exhibit very similar motion patterns, resulting in ID switches (green line turns into orange line). For our GroupTrack, it utilizes group motion patterns from the neighboring tracklets (light blue ones with arrows) to correct the low confidence box, and thus can learn a robust motion model. Note that the red dotted lines in Fig. 1 represent the Ground Truth (GT) trajectories. Also, for a long-term trajectory in Fig. 1 (b), the pedestrian in purple box will walk behind the billboard in a few frames, making it difficult to be detected. For traditional MOT trackers, they usually model target motion individually, e.g., Kalman Filter

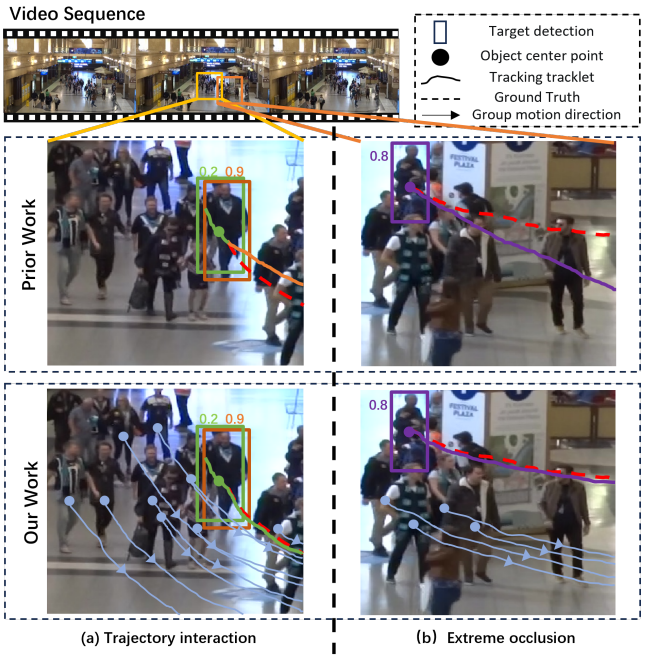

Fig. 1: Different methods of motion modelling in crowded scenes. For the trajectory interaction (a), the two targets have almost the same motion patterns after individually updating their motion models, respectively, causing false matching (green line turns into orange line). In the extreme occlusion (b), the individual motion model (e.g., Kalman Filter) gradually becomes inaccurate in the long-term tracking due to the occlusions, which leads to a drifting trajectory (see the purple line in the top-right). Using group motion patterns (light blue tracklets with arrows), our proposed GroupTrack can infer the location of undetected or occluded box which is employed to update the motion model during untracked phase. This strategy contributes to the success of re-association in complex scenarios (red dotted lines denoted GT trajectories).

This work was supported in part by the National Natural Science Foundation of China under Grants 62206075, 61733011, and 62261160652, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012028, and in part by the Shenzhen Science and Technology Program under Grants RCBS20221008093220004 and GXWD20231129125006001. *(Corresponding author: Weihong Ren)*

Xinglong Xu, Weihong Ren, Haoyu Ji, Yu Gao and Honghai Liu are with the State Key Lab of Robotics and Systems, School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China. (e-mail: xinglongxu@stu.hit.edu.cn, renweihong@hit.edu.cn, jihaoyu@stu.hit.edu.cn, gaoyu@stu.hit.edu.cn, honghai.liu@hit.edu.cn).

Gan Sun is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China. (email: sungan1412@gmail.com).

(KF), which can only infer target motion using previous target state. However, individual motion modelling causes prediction shift in long-term tracking due to the cumulative error. For our GroupTrack, it can aggregate motion prior from neighboring tracklets of a target, and thus works well in the occlusion scenarios (see the purple line in bottom-right).

To solve the problem of trajectory interaction and extreme occlusion, the recent works make efforts in three aspects: motion modelling, appearance learning, and association designing. For motion modelling, BoT-SORT [13] modifies the state vector of KF to generate a relatively accurate prediction, and also uses Camera Motion Compensation (CMC) to perform inter-frame alignment to tackle with dynamic scenes. Similarly, OC-SORT [14] adopts target observation to formulate a virtual tracklet throughout the period of occlusion, mitigating the accumulation of filter parameter errors, rather than solely depending on linear state estimation. To promote motion modeling with deep learning technique, MotionTrack [15] models complex motion in dense crowds through Interaction Module and Refind Module. However, the motion modules only model individually motion for each single target, ignoring the prior from group motion patterns, which may cause false matching in extremely occluded scenarios. For appearance learning, the works [16], [17] employ memory technology to retain a variety of features for each target, and subsequently identify different targets through a multi-query approach. [18] concentrates on exploring detailed representation, which thoroughly portrays appearance from both global and local perspectives. Though the above methods can improve the discriminability of appearance features, they need to consume extra resources for training and matching, which is not conducive to real-time tracking. For association designing, [19] conducts a comprehensive analysis of scenarios where the appearance of the target is insufficient, and determine when these shortcomings can be offset by the integration of motion data. [20] formulates a depth cascading matching algorithm that employs pseudo-depth information to transform a dense target set into several sparse target subsets for data association. The association strategies have been proven effective for MOT, but they usually need to carefully tune the hyper-parameters under different scenarios.

Different from the existing methods, in this paper, we propose a simple yet efficient MOT tracker, *i.e.*, GroupTrack, to learn reliable motion state for each target using group motion patterns. In particular, our GroupTrack follows the tracking-by-detection paradigm, in which group motion patterns are introduced to associate low-quality tracklets. For each tracklet, we choose its neighboring tracklets to form a group of motion patterns, which can provide informative clues for the motion estimation of the current tracklet. During the tracking process, we apply the group motion patterns to perform tracklet prediction and data association, which contributes to the success of re-association in dense crowds and occlusion scenarios. Experimental results on two public datasets (MOT17 and MOT20) confirm that our proposed GroupTrack surpasses the performance of prior state-of-the-art methods.

To summarize, our contributions are three-fold:

- Different from the traditional individual motion modelling, we reveal that MOT performance can be further improved by motion modelling using group motion patterns.
- By integrating prior from group motion patterns, our GroupTrack can correct object detection with low confidence, which creates a robust motion model in occlusion scenarios.
- Only using IoU as association metric, the proposed GroupTrack achieves state-of-the-art results on two public MOT benchmarks.

II. RELATED WORKS

Here, we briefly review the MOT works including tracking-by-detection, motion models and handling objection occlusion. Comprehensive reviews on MOT can be found in [21]–[23].

A. Tracking-by-Detection

With the rapid development of object detection, the MOT trackers based on the tracking-by-detection paradigm [24] have made significant progress. The current state-of-the-art object detectors [9], [12], [25] are used to provide detection boxes in each frame, and then data association with motion or appearance modelling is performed to build trajectories across different frames in a video sequence. The pioneer work SORT [26] first adopts the KF as a motion model to predict the location of each tracklet in the next frame, and then uses Hungarian algorithm [27] to associate detection boxes across the whole video, forming trajectories. DeepSort [28] further extend SORT by adding a Re-Identification (ReID) model to obtain appearance features, which can provide critical clues for long-term tracking. Because of the effectiveness of appearance cues, some works [29]–[31] employ the advanced ReID models [32]–[34] to generate high-quality instance appearance embeddings, resulting in significant improvement of MOT. The tracking-by-detection approach adopts two separate steps to perform object tracking, which is not easy to be applied to real-time applications. To improve the tracking efficiency, joint detection and ReID models [35]–[38] become more and more popular. For example, FairMOT [39] adds an extra ReID branch based on CenterNet [25], which is completely homogeneous with the detection branch. This method is essentially different from the previous method of performing detection and ReID in a two-step cascaded style. However, it is difficult to balance the detection task and the ReID task, leading to a suboptimal solution for MOT.

B. Motion Models

For MOT, most of the current matching strategies [20], [26], [40]–[42] are based on motion models, and they perform frame-by-frame data association directly using motion and location cues. Recently, ByteTrack [43] has achieved the state-of-the-art tracking performance by only using the strong

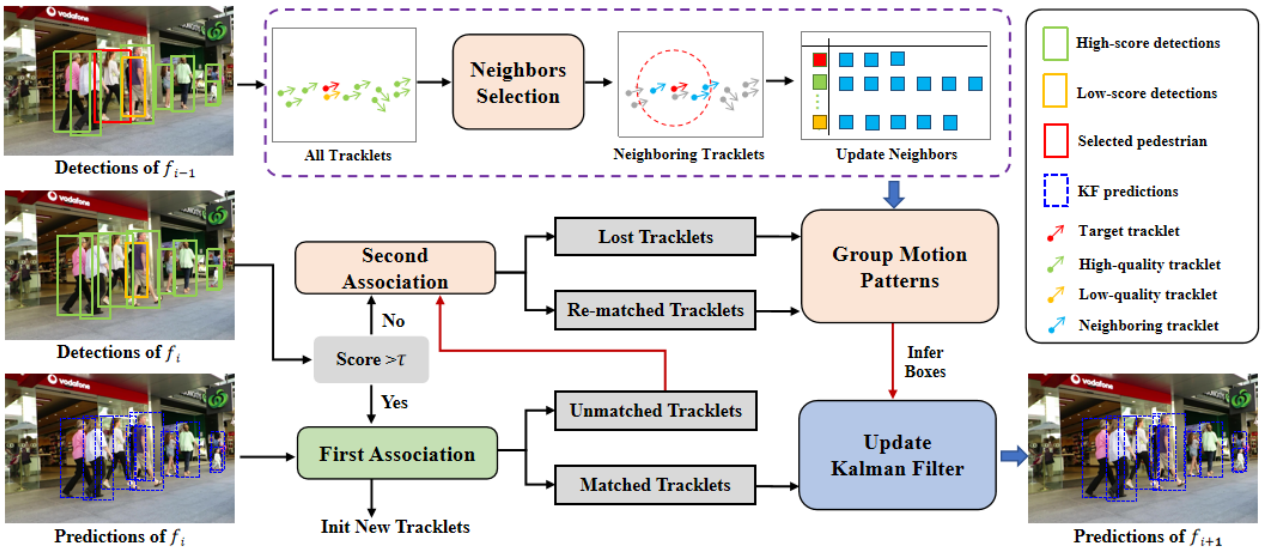

Fig. 2: The overall framework of GroupTrack. For the current frame f_i, we first separate the object detections into high-score and low-score sets. Then, we perform the first association using the high-score boxes. For the matched tracklets, we directly update their KF models. However, for the unmatched ones, we perform the second association with the low-score detections. When one tracklet is low-quality (lost track or re-matched track), we adopt the group motion patterns from its neighbors to infer the occluded or undetectable box which is finally used to update the corresponding KF motion model.

detector YOLOX [12] and motion model KF. Thus, a series of works [44]–[47] are emerging to focus on exploring robust motion models. For example, many studies use the variants of KF to obtain more accurate predictions, such as [46], [47]. They directly incorporate the detection scores into the KF, and try to improve the robustness of the motion model. To handle dynamic scenes, [13] adopts CMC by aligning frames via image registration, and [14] utilizes target observation to create a virtual trajectory during the occlusion period instead of only relying on the linear state estimate. Moreover, recent advances integrate motion and visual information together to provide better trajectory prediction. For example, the works [41], [48] build the tracking branches on the object detectors to predict the inter-frame displacements, aiming to generate prediction for each trajectory in the next frame. To further model surrounding interactions, ArTIST [49] treats target motion as a probability distribution to explore interactive information. In contrast, MotionTrack [15] adopts Graph Neural Networks (GNN) and self-attention mechanism to model the interaction among different trajectories, which can alleviate the motion modeling problem in long-term occlusion. The above approaches only focus on modeling individual motion pattern for each target, but ignore the informative clues from group motion patterns. As a result, in dense crowds, the individual motion model (e.g., KF) gradually becomes inaccurate in the long-term tracking due to the occlusion, which leads to many fragments for each trajectory.

C. Handling Object Occlusion

In MOT, the ability to handle occlusion is very important in crowd scenarios, and many studies are dedicated to address this problem. E.g., TBC [50] utilizes crowd density maps to integrate the modeling of detection, counting, and tracking of multiple targets into a network flow program, which can find the global optimum for detection and tracking across the entire video. However, the performance of TBC depends too much on the quality of the crowd density maps. Some works [16]–[18] attempt to distinguish the unoccluded and occluded targets by using attention mechanisms and contrastive learning. However, they inevitably introduce the additional computational costs, which is also not conducive to long-term tracking. Considering that a large number of detection boxes with low confidence are generated by occlusion, Bytetrack [43] utilizes the similarity between the low-score detection boxes and the untracked tracklets to recall occluded detections and filter out the background distractors. ByteTrack can significantly improve the tracking performance by a simple association strategy, but its motion model is prone to deterioration in trajectory interaction and extreme occlusion scenes due to the incorporation of detection boxes with low confidence. To address this problem, SparseTrack [20] decomposes dense target sets based on pseudo-depth information and then performs data association on sparse subsets, thus avoiding erroneous matches caused by excessively dense detection boxes. However, for highly dynamic scenes, this method becomes less reliable due to the inaccurate depth information. Different from the above methods, we propose a simple yet effective method, to learn reliable motion state for each target using group motion patterns. Our method aims to motivate other researchers to focus on motion modelling using group prior rather than only using individual state information.

III. GROUPTRACK

In this section, we introduce our MOT tracker GroupTrack, which can learn reliable motion state for each target using group motion patterns.

A. Overview

GroupTrack follows the tracking-by-detection paradigm and establishes trajectories across different frames by performing data association. The overall tracking framework is shown in Fig. 2. For each given frame f_i in a video, it is first processed by the YOLOX [12] detector to obtain object detections, which consist of bounding boxes (x, y, w, h) and confidence scores s. and then these object detections are associated together across the whole video to establish target trajectories. During tracking, for each tracklet, we choose its neighboring ones to form a group of motion patterns based on relative distance and moving velocity, and the specific details are described in Sec. III-B. In the data association process at frame f_i, we apply the prior from group motion patterns to infer accurate box predictions in the next frame for low-quality tracklets (unmatched tracklets in the first association), which avoids updating KF with inaccurate detections and thus can obtain robust motion models, and the details are described in Sec. III-C. The pseudo-code of GroupTrack is shown in Algorithm 1.

B. Neighboring Tracklets Selection

For traditional MOT trackers, they usually model target motion individually for each trajectory, and are inclined to cause trajectory drift. Here, we introduce how to aggregate prior from group motion patterns to perform tracklet prediction. Intuitively, a group of people nearby should have the similar motion patterns. Thus, we jointly consider relative distance and moving velocity to select neighboring tracklets for each tracklet.

Specifically, within a single frame, for the high-quality tracklets $\mathcal{T}_{matched}$, which are matched in the first association, we calculate a distance matrix using the Euclidean distance over center points of all the tracklets in $\mathcal{T}_{matched}$. The width of a detection box usually contains the information of camera perspective and also implicitly reflects the target depth, and we thus utilize it to construct Standardized Euclidean Distance (SED):

$$D(T^c, {}^mT^c) = \frac{\|T^c - {}^mT^c\|}{w}, \tag{1}$$

where T^c and ${}^mT^c$ represent the current center locations of the target tracklet T and the tracklet ${}^mT$ in $\mathcal{T}_{matched}$, respectively, and w is the width of the target tracklet T. Also, the neighboring tracklets should have the consistent motion patterns with the target one, and we further take the Velocity Cosine Distance (VCD) into consideration to filter out the distracted tracklets:

$$V(T^v, {}^mT^v) = 2 - \frac{T^v \cdot {}^mT^v}{\|T^v\| \|{}^mT^v\|}, \tag{2}$$

where T^v and ${}^mT^v$ are the current velocities of the target tracklet T and the tracklet ${}^mT$ in $\mathcal{T}_{matched}$, respectively.

To enhance the robustness, we calculate the trajectory similarity based on the temporal correlations rather than individual frame. A temporal window n is first chosen as:

$$n = \min\{\min\{\text{len}(T), \text{len}({}^mT)\}, 30\}, \tag{3}$$

where len(T) and len$({}^mT)$ represent the length of the target tracklet T and the tracklet ${}^mT$ in $\mathcal{T}_{matched}$ after their most recent re-matched, respectively.

Finally, we compute the similarity between a tracklet ${}^mT$ in $T_{matched}$ and the target tracklet T in the current frame I, which is expressed as:

$$S_{total}(T, {}^mT) = \frac{\sum_{i=I-n}^{I-1} S(T_i, {}^mT_i)}{n}, \tag{4}$$

$$S(T_i, {}^mT_i) = \begin{cases} -\infty & V(T_i^v, {}^mT_i^v) \geq 2 \\ \dfrac{1}{D(T_i^v, {}^mT_i^v) \cdot V(T_i^v, {}^mT_i^v)} & \text{otherwise} \end{cases}, \tag{5}$$

If $S_{total}(T, {}^mT)$ is greater than a given similarity threshold τ, the tracklet ${}^mT$ is then regraded as a neighbor tracklet for the target tracklet T.

Note that in the stage of neighboring tracklets selection, we only update the neighboring tracklets for the high-quality ones in $\mathcal{T}_{matched}$, and ensure that the neighboring tracklets must be high-quality, which can avoid damaging the group motion patterns. For each low-quality tracklet, we keep its neighboring tracklets unchanged at the current frame.

C. Data Association of GroupTrack

Based on the neighboring tracklets for each tracklet, we can build the group motion patterns which are then integrated into the data association process. Following ByteTrack [43], we first divide object detections into high-score detection set D_{high} and low-score detection set D_{low} based on confidence scores. For each tracklet in $\mathcal{T}$, we calculate its prior state estimation at frame t using the following formula:

$$\begin{cases} x_{t|t-1} = F_t x_{t-1|t-1} \\ P_{t|t-1} = F_t P_{t-1|t-1} F_t^\top + Q_t \end{cases}, \tag{6}$$

where $x_{t|t-1}$ is the prior state estimation, F is the state transition model, P is the covariance matrix, and Q is the process noise. Then, we perform the first data association using D_{high} by the Hungarian Matching (HM) algorithm, and output the matched tracklets $\mathcal{T}_{matched}$, unmatched tracklets $\mathcal{T}_{unmatched}$ and the unmatched detections $D_{unmatched}$. For each tracklet in $\mathcal{T}_{matched}$, we update the prior state estimation with its matching observation z_t, resulting in a posteriori state estimate $x_{t|t}$ as output:

$$\begin{cases} K_t = P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R_t)^{-1} \\ x_{t|t} = x_{t|t-1} + K_t(z_t - H_t x_{t|t-1}) \\ P_{t|t} = (I - K_t H_t) P_{t|t-1} \end{cases}, \tag{7}$$

where R is the observation noise, and K is the kalman gain.

Afterwards, the second data association is performed using the unmatched tracklets $\mathcal{T}_{unmatched}$ and the low-score detections D_{low}. Then, we can mark the re-matched tracklets

$\mathcal{T}_{re\_matched}$, and also find out the lost tracklets $\mathcal{T}_{lost}$. For the low-quality tracklets $\mathcal{T}_{low} \leftarrow \{\mathcal{T}_{re\_matched}, \mathcal{T}_{lost}\}$, it is not advisable to directly update the motion model KF, since the current target state is inaccurate (low-score or missed detection) due to the occlusion or dense crowds. Thus, we make use of the prior from Group Motion Patterns (GMP) to infer an accurate object detection for the low-quality tracklet. Suppose T_j is one of the neighboring tracklets of T_{low}, and the possible center location L_{low}^j for T_{low} can be inferred and updated by:

$$L_{low}^j = \tilde{T}_{low}^c + (\tilde{T}_{low}^v - \tilde{T}_j^v) + T_j^v, \tag{8}$$

where $\tilde{T}_{low}^c$ represents the center location of T_{low} in the previous frame, $\tilde{T}_{low}^v$ and $\tilde{T}_j^v$ are the previous velocities for T_{low} and T_j, respectively, and T_j^v is the current velocity of T_j. Usually, the objects in a group should have the similar motion patterns, and thus they have fixed relative velocity. For the target tracklet T_{low}, its current velocity is inaccurate due to missed or occluded detection. However, its neighboring tracklets have accurate motion models, and thus we can use the relative velocity in the previous frame to infer the target motion, namely $(\tilde{T}_{low}^v - \tilde{T}_j^v) + T_j^v$. The potential target location can be estimated by (8) from T_j. Note that the previous velocities are directly taken from KF motion models. When considering all the neighboring tracklets, in order to better aggregate the prior cues, the new center location of T_{low} can be updated by:

$$T_{low}^c = \frac{\sum_{j=1}^{N} L_{low}^j \cdot S(T_{low}, T_j)}{\sum_{j=1}^{N} S(T_{low}, T_j)}, \tag{9}$$

where N is the total number of neighboring tracklets for T_{low}. Here, we only infer the center location T_{low}^c, and the width and the height remain the same with the previous frame. For a tracklet, it has similar motion pattern with its neighbors, but the box scale may be very different due to the camera perspective.

The inferred boxes $\mathcal{D}_{infer}$ are then used to update the target state of $\mathcal{T}_{re\_matched}$ and $\mathcal{T}_{lost}$ by using Eq. (7). Unlike the first update, we utilize the more reliable inferred boxes $\mathcal{D}_{infer}$ as the observations. Finally, the new tracklets are added using the high-score detections in $\mathcal{D}_{unmatched}$. After achiving the tracking results in the current frame, we adopt (4) to update the neighboring tracklets for each high-qaulity tracklets in $\mathcal{T}_{matched}$ (see Algorithm 1 for more details).

Algorithm 1: The Data Association of GroupTrack.

Input: Video sequence V; object detector Det; high-score detection threshold τ; Kalman Filter KF
Output: Trajectories $\mathcal{T}$ of the video V

Initialization: $\mathcal{T} \leftarrow \emptyset$
for *frame* f_i *in* V **do**
  $D_i \leftarrow \texttt{Det}(f_i)$ /* detections per frame */
  /* divided into two parts by τ */
  $D_{high}, D_{low} \leftarrow D_i$
  /* predictions in f_i of $\mathcal{T}$ */
  $\mathcal{T} \leftarrow \texttt{KF\_Predict}(\mathcal{T})$
  /* first association */
  $\mathcal{T}_{matched}, \mathcal{T}_{unmatched}, D_{unmatched} \leftarrow \texttt{HM}(\mathcal{T}, D_{high})$
  /* updates in $\mathcal{T}_{matched}$ */
  $\mathcal{T}_{matched} \leftarrow \texttt{KF\_Update}(\mathcal{T}_{matched})$
  Infer boxes for low-quality tracklets using GMP, *i.e.*, (8) & (9)
  /* second association */
  $\mathcal{T}_{re\_matched}, \mathcal{T}_{lost} \leftarrow \texttt{HM}(T_{unmatched}, D_{low})$
  $D_{infer} = \texttt{GMP}(\{\mathcal{T}_{re\_matched}, \mathcal{T}_{lost}\}, \mathcal{T}_{neibors})$
  /* re-updates in $\mathcal{T}_{re\_matched}, \mathcal{T}_{lost}$ */
  $\mathcal{T}_{re\_matched}, \mathcal{T}_{lost} \leftarrow$ KF_Re-update($\{\mathrm{T}_{re\_matched}, \mathcal{T}_{lost}; \mathrm{D}_{infer}\}$)
  /* add new tracklets */
  for d *in* $\mathcal{D}_{unmatched}$ **do**
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$
  end
  /* update tracking results */
  $\mathcal{T} \leftarrow \{\mathcal{T}_{matched}, \mathcal{T}_{re\_matched}, \mathcal{T}_{lost}\}$;
  Note that NTS is Eq. (4)
  /* update neighboring tracklets */
  $\mathcal{T}_{neibors} = \texttt{NTS}(\mathcal{T}_{matched})$
end
Return: $\mathcal{T}$

Track termination is not shown in the pseudo-code for simplicity, and the key steps of GroupTrack are in green.

IV. Experiments

A. Experimental Setting

Datasets: We evaluate our GroupTrack on the MOT17 [51] and MOT20 [52] datasets, and submit the test results to the official MOT Challenge evaluation server for comparison with the state-of-the-art trackers. Following the commonly used setting, we also utilize the publicly accessible YOLOX detector [12], trained by ByteTrack [43] on the MOT17 and MOT20 datasets to obtain the object detections. For ablation studies, we employ the initial half of each video from the MOT17 training set for training and the latter half for evaluation. [41].

Evaluation Metrics: We adopt the CLEAR metrics [53] (MOTA, IDs, FP, FN, *etc.*), IDF1 [54], and HOTA [55] to evaluate the tracking performance. In particular, MOTA is a comprehensive metric that jointly measures IDs, FP and FN, and prefers to reflect the detection performance. IDF1 focuses on the performance of data association, and HOTA is also an encompassing metric to evaluate the combined efficiency of detection and association.

Implementation Details: For fair comparison, we directly take the YOLOX [12] detector pertained by [43] to generate the object detections. For the data association, we set different similarity threshold τ for different datasets according to the crowd level and the camera perspective. For MOT17, we set the default τ to 0.5 and keep the maximum length of lost tracks for 30 frames. For MOT20, we set τ to 0.3 and keep

TABLE I: Comparison of the state-of-the-art methods under the private detection on the **MOT17** and **MOT20** test set. The best results are marked in **bold**, and the second are underlined.

Method	MOTA↑	IDF1↑	HOTA↑	FP↓	FN↓	IDs↓
MOT17 private detection						
ReMOT [56] (IVC'21)	77.0	72.0	59.7	33204	93612	2853
GSDT [57] (ICRA'21)	73.2	66.5	55.2	26397	120666	3891
QuasiDense [37] (CVPR'21)	68.7	66.3	53.9	26589	146643	3378
SOTMOT [58] (CVPR'21)	71.0	71.9	-	39537	118983	5184
SiamMOT [59] (CVPR'21)	76.3	72.3	-	-	-	-
CorrTracker [60] (CVPR'21)	76.5	73.6	60.7	29808	99510	3369
PermaTrackPr [61] (ICCV'21)	73.8	68.9	55.5	28998	115104	3699
FairMOT [39] (IJCV'21)	73.7	72.3	59.3	27507	117477	3303
CSTrack [35] (TIP'22)	74.9	72.6	59.3	23847	114303	3567
RelationTrack [62] (TMM'22)	73.8	74.7	61.0	27999	118623	1374
TrackFormer [63] (CVPR'22)	74.1	68.0	57.3	34602	108777	2829
MeMOT [17] (CVPR'22)	72.5	69.0	56.9	37221	115248	2724
MTrack [16] (CVPR'22)	72.1	73.5	-	53361	101844	2028
MOTR [64] (ECCV'22)	73.4	68.6	57.8	-	-	2439
ByteTrack [43] (ECCV'22)	80.3	77.3	63.1	25491	83721	2196
P3AFormer [65] (ECCV'22)	81.2	78.1	-	**17281**	86861	1893
MOTRv2 [66] (CVPR'23)	78.6	75.0	62.0	23409	94797	2619
GHOST [19] (CVPR'23)	78.7	77.1	62.8	-	-	2325
MotionTrack [15] (CVPR'23)	81.1	80.1	65.1	23802	81660	**1140**
UCMCTrack [67] (AAAI'24)	80.6	**81.0**	**65.7**	36213	**71454**	1689
Hybrid-SORT [68] (AAAI'24)	79.3	78.4	63.6	-	-	-
SMILEtrack [69] (AAAI'24)	80.7	80.1	65.0	23187	81792	1251
GroupTrack (ours)	**81.3**	80.3	65.2	22278	81993	1161
MOT20 private detection						
FairMOT [39] (IJCV'21)	61.8	67.3	54.6	103440	88901	5243
GSDT [57] (ICRA'21)	67.1	67.5	53.6	31507	135395	3230
CorrTracker [60] (CVPR'21)	65.2	69.1	-	79429	95855	5183
SiamMOT [59] (CVPR'21)	67.1	69.1	-	-	-	-
SOTMOT [58] (CVPR'21)	68.6	71.4	57.4	57064	101154	4209
CSTrack [35] (TIP'22)	66.6	68.6	54.0	25404	144358	3196
RelationTrack [62] (TMM'22)	67.2	70.5	56.5	61134	104597	4243
MeMOT [17] (CVPR'22)	63.7	66.1	54.1	47882	137982	1938
MTrack [16] (CVPR'22)	63.5	69.2	-	96123	86964	6031
ByteTrack [43] (ECCV'22)	77.8	75.2	61.3	26249	87594	1223
P3AFormer [65] (ECCV'22)	78.1	76.4	-	25413	86510	1332
MOTRv2 [66] (CVPR'23)	76.2	72.2	60.3	-	-	-
GHOST [19] (CVPR'23)	73.7	75.2	61.2	-	-	1264
MotionTrack [15] (CVPR'23)	78.0	76.5	62.8	28629	**84152**	1165
UCMCTrack [67] (AAAI'24)	75.6	77.4	62.8	28678	96199	1335
Hybrid-SORT [68] (AAAI'24)	76.4	76.2	62.5	-	-	-
SMILEtrack [69] (AAAI'24)	78.0	76.3	63.0	**23246**	86112	1208
GroupTrack (ours)	**78.2**	**77.5**	**63.5**	25154	86676	**1106**

the maximum length of lost tracks for 60 frames. Besides, the high-score detection threshold is set to 0.6. Note that all the tracking results in this work are obtained by only using IoU as association metric.

B. Evaluation on Different Datasets

We compare our GroupTrack with the state-of-the-art trackers on the test sets of MOT17 and MOT20 under the private detection protocol, and the tracking results are summarized in Table I. GroupTrack is denoted as "GD1" in the MOT Challenge website.

MOT17 Dataset. Compared to the baseline ByteTrack, GroupTrack adopt group motion patterns to perform the data association, and can achieve gains of +1.0 MOTA, +3.0 IDF1, +2.1 HOTA and IDs has almost halved. Our GroupTrack focuses on solving the challenge of learning robust motion model in dense crowds and extreme occlusion.

TABLE II: The effect of integrating group motion patterns on MOT17 validation set

Methods	w/ GMP	MOTA↑	IDF1↑	FP↓	FN↓	IDs↓
GroupTrack	✗	76.4	79.3	3354	9189	174
GroupTrack	✓	**77.2**	**81.0**	**3124**	**8992**	**141**

TABLE III: The effects of Distance (SEC) and Velocity Cosine Distance (VCD) on MOT17 validation set.

Methods	MOTA↑	IDF1↑	FP↓	FN↓	IDs↓
-	76.4	79.3	3354	9189	174
VCD	76.5	79.6	3318	9203	166
SEC	77.0	80.7	3201	9023	152
VCD + SEC	**77.2**	**81.0**	**3124**	**8992**	**141**

It makes full use of group motion patterns to infer the occluded or missed detections, and thus can maintain a continuous trajectory in a long term. Thus, in terms of MOTA, our GroupTrack achieves the best results. Using deep learning technique, MotionTrack [15] introduces an Interaction Module and Refined module to learn discriminative motion patterns, but it considers the interaction with all the trajectories which can't correct the low-score detections. Our GroupTrack is only based on KF, and achieves higher MOTA than that of MotionTrack (MOTA of 81.3 vs 81.1). Also, it has relatively less FP (FP of 22278 vs 23802).

MOT20 Dataset. Compared with MOT17, MOT20 is more challenging, and it has many highly crowded scenarios, which contain trajectory interactions and mutual occlusions. For the previous trackers, they model target motion individually, which can easily drift to other distracted trajectory due to the close distance. In contrast, our GroupTrack models the target motion using prior from group motion patterns, and the dense crowds can provide more reliable neighboring tracklets. As observed in Table I, GroupTrack achieves the best results in the key metrics: MOTA, IDF1, HOTA and IDs. Compared to the baseline ByteTrack, GroupTrack achieves gains of +0.4 MOTA, +2.3 IDF1, +2.2 HOTA and -117 IDs. For the recent state-of-the-art tracker MotionTrack, our GroupTrack outperforms it in metrics both reflecting the detection ability and association ability (*i.e.*, +0.2 MOTA, +1.0 IDF1, +0.7 HOTA and -59 IDs). It is worth mentioning that our tracker only utilizes simple KF motion model with group motion patterns, which is more effective than other trackers that model target motion or learn ReID feature using complex network architectures.

C. Ablation Studies of GroupTrack

The Effect of Group Motion Patterns. To evaluate the effectiveness of group motion patterns (GMP), we create a variant of GroupTrack without using GMP for motion modelling. The results on the MOT17 validation set are reported in Table II. As observed, GroupTrack achieves gains of +0.8 MOTA, +1.7 IDF1, -230 FP, -197 FN and -33 IDs when equipped with GMP. The results also indicate that the GMP can reduce false matching because it helps to correct the occluded or missed detections on the low-quality

TABLE IV: The effect of different strategies of aggregating position cues on MOT17 validation set.

Strategies	MOTA↑	IDF1↑	FP↓	FN↓	IDs↓
Max	76.7	80.3	3301	9117	165
Average	77.1	80.8	3147	8999	144
Weighted Average	**77.2**	**81.0**	**3124**	**8992**	**141**

TABLE V: The effect of similarity threshold τ on MOT17 validation set.

τ	MOTA↑	IDF1↑	FP↓	FN↓	IDs↓
0.3	76.6	79.8	3384	9168	170
0.4	77.0	80.5	3245	9011	152
0.5	**77.2**	**81.0**	3124	**8992**	**141**
0.6	76.9	80.7	**3089**	9124	159
0.7	76.8	80.7	3105	9084	162

tracklets, which is crucial for tracking task in dense crowds.

The Effect of Neighboring Tracklets Selection. For GroupTrack, we also explore the effect of using different similarity metrics to select neighboring tracklets on the MOT17 validation set. The tracking results are reported in Table III. When we only utilize VCD, the MOTA and IDF1 slightly increases, and the number of FP, FN and IDs decreases. The reason is that the neighboring tracklets with similar motion patterns can provide reliable clues to correct object detection, and thus can effectively improve the robustness of motion model. However, VCD does not consider the relative locations of the neighboring tracklets, inevitably introducing distant erroneous tracklets that degrade tracking performance. When only using SEC, the tracking performance can be significantly improved. This suggests that the closer trajectories usually share the same motion patterns. Upon jointly combining VCD and SEC, the performance can be further advanced with the well-matched neighboring tracklets.

The Effect of Aggregating Location Cues. Besides, for GroupTrack, we also investigate the effects of the strategies of aggregating location cues in Eq. (9), and the results are reported in Table IV. By taking a simple average of the inferred locations from neighboring tracklets (denoted as "Average"), it can achieve superior tracking performance compared to solely utilizing the inferred location from the best-matched neighboring tracklet, denoted as "Max" in table IV (*e.g.*, +0.4 MOTA, +0.5 IDF1 and -21 IDs). The reason is that when the group motion patterns heavily rely on a single neighboring tracklet, an abrupt motion change of that tracklet can lead to erroneous location inference, resulting in tracking failure. When we adopt a weighted average based on the tracklet similarities, the best tracking performance can be achieved. This suggests that the weighted average strategy can effectively aggregate location cues from the neighboring tracklets.

The Effect of Similarity Threshold τ. For GroupTrack, we also explore the effect of using different similarity thresholds τ on the MOT17 validation set. The tracking results are reported in Table V. When τ increases from 0.3 to 0.5, the tracking performance can be significantly improved. This indicates that a higher similarity threshold can effectively filter out erroneous neighboring tracklets, thereby enhancing the prior from group motion patterns. However, when τ continues to increase, the tracking performance gradually decreases. The reason is that an overly high threshold can lead to a reduction in the number of neighboring tracklets, thus providing insufficient clues for establishing group motion patterns, and damaging the robustness of the motion model.

V. CONCLUSIONS

We propose a simple yet efficient MOT tracker, *i.e.*, GroupTrack, to learn robust motion state for each target using group motion patterns, which demonstrates its effectiveness in dense crowds and occluded scenarios. The group motion patterns can help to correct occluded or missed detections, and thus can solve the problem of false matching. We have achieved state-of-the-art performance on two public datasets MOT17 and MOT20. Currently, we don't consider the target motion with sudden change of velocity, in which situation we can't aggregate enough prior from group motion patterns. In the future, we will also apply our method to 3D object tracking in various scenarios.

REFERENCES

[1] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *TPAMI*, vol. 44, no. 7, pp. 3614–3633, 2021.

[2] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *ICCV*, 2015, pp. 2722–2730.

[3] A. Buyval, A. Gabdullin, R. Mustafin, and I. Shimchik, "Realtime vehicle and pedestrian tracking for didi udacity self-driving car challenge," in *ICRA*, 2018, pp. 2064–2069.

[4] L. Naik, T. M. Iversen, A. Kramberger, J. Wilm, and N. Krüger, "Multi-view object pose distribution tracking for pre-grasp planning on mobile robots," in *ICRA*, 2022, pp. 1554–1561.

[5] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011, pp. 3153–3160.

[6] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *CVPR*, 2018, pp. 6154–6162.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NeurIPS*, vol. 28, 2015.

[10] P. Sun, Y. Jiang, E. Xie, W. Shao, Z. Yuan, C. Wang, and P. Luo, "What makes for end-to-end object detection?" in *ICML*, 2021, pp. 9934–9944.

[11] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *CVPR*, 2021, pp. 14 454–14 463.

[12] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv:2107.08430*, 2021.

[13] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *arXiv:2206.14651*, 2022.

[14] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *CVPR*, 2023, pp. 9686–9696.

[15] Z. Qin, S. Zhou, L. Wang, J. Duan, G. Hua, and W. Tang, "Motiontrack: Learning robust short-term and long-term motions for multi-object tracking," in *CVPR*, 2023, pp. 17 939–17 948.

[16] E. Yu, Z. Li, and S. Han, "Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking," in *CVPR*, 2022, pp. 8834–8843.

[17] J. Cai, M. Xu, W. Li, Y. Xiong, W. Xia, Z. Tu, and S. Soatto, "Memot: Multi-object tracking with memory," in *CVPR*, 2022, pp. 8090–8100.

[18] H. Ren, S. Han, H. Ding, Z. Zhang, H. Wang, and F. Wang, "Focus on details: Online multi-object tracking with diverse fine-grained representation," in *CVPR*, 2023, pp. 11 289–11 298.

[19] J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi, and L. Leal-Taixé, "Simple cues lead to a strong multi-object tracker," in *CVPR*, 2023, pp. 13 813–13 823.

[20] Z. Liu, X. Wang, C. Wang, W. Liu, and X. Bai, "Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth," *arXiv:2306.05238*, 2023.

[21] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[22] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, "Deep learning in multi-object detection and tracking: state of the art," *Applied Intelligence*, vol. 51, pp. 6400–6429, 2021.

[23] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[24] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *CVPR*, 2008, pp. 1–8.

[25] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *ICCV*, 2019, pp. 6569–6578.

[26] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *ICIP*, 2016, pp. 3464–3468.

[27] H. W. Kuhn, "The hungarian method for the assignment problem," *NAV RES LOG*, vol. 2, no. 1-2, pp. 83–97, 1955.

[28] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *ICIP*, 2017, pp. 3645–3649.

[29] Z. Wang, H. Zhao, Y.-L. Li, S. Wang, P. Torr, and L. Bertinetto, "Do different tracking tasks require different appearance models?" *NeurIPS*, vol. 34, pp. 726–738, 2021.

[30] T. Fischer, T. E. Huang, J. Pang, L. Qiu, H. Chen, T. Darrell, and F. Yu, "Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking," *TPAMI*, 2023.

[31] S. You, H. Yao, B.-K. Bao, and C. Xu, "Utm: A unified multiple object tracking model with identity-aware feature enhancement," in *CVPR*, 2023, pp. 21 876–21 886.

[32] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "Fastreid: A pytorch toolbox for general instance re-identification," *arXiv:2006.02631*, 2020.

[33] K. Zhou and T. Xiang, "Torchreid: A library for deep learning person re-identification in pytorch," *arXiv:1910.10093*, 2019.

[34] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *TPAMI*, vol. 44, no. 9, pp. 5056–5069, 2021.

[35] C. Liang, Z. Zhang, X. Zhou, B. Li, S. Zhu, and W. Hu, "Rethinking the competition between detection and reid in multiobject tracking," *TIP*, vol. 31, pp. 3182–3196, 2022.

[36] Z. Lu, V. Rathod, R. Votel, and J. Huang, "Retinatrack: Online single stage joint detection and tracking," in *CVPR*, 2020, pp. 14 668–14 678.

[37] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *CVPR*, 2021, pp. 164–173.

[38] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *ECCV*, 2020, pp. 107–122.

[39] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *IJCV*, vol. 129, pp. 3069–3087, 2021.

[40] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *ICCV*, 2015, pp. 3029–3037.

[41] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *ECCV*, 2020, pp. 474–490.

[42] F. Yang, S. Odashima, S. Masui, and S. Jiang, "Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space," in *WACV*, 2023, pp. 4799–4808.

[43] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *ECCV*, 2022, pp. 1–21.

[44] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, and X. Pan, "Mat: Motion-aware multi-object tracking," *Neurocomputing*, vol. 476, pp. 75–86, 2022.

[45] T. Khurana, A. Dave, and D. Ramanan, "Detecting invisible people," in *ICCV*, 2021, pp. 3174–3184.

[46] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *TMM*, 2023.

[47] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021," in *ICCV*, 2021, pp. 2809–2819.

[48] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *ICCV*, 2019, pp. 941–951.

[49] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould, "Probabilistic tracklet scoring and inpainting for multiple object tracking," in *CVPR*, 2021, pp. 14 329–14 339.

[50] W. Ren, X. Wang, J. Tian, Y. Tang, and A. B. Chan, "Tracking-by-counting: Using network flows on crowd density maps for tracking multiple targets," *TIP*, vol. 30, pp. 1439–1452, 2020.

[51] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv:1603.00831*, 2016.

[52] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," *arXiv:2003.09003*, 2020.

[53] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *JIVP*, vol. 2008, pp. 1–10, 2008.

[54] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV*, 2016, pp. 17–35.

[55] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *IJCV*, vol. 129, pp. 548–578, 2021.

[56] F. Yang, X. Chang, S. Sakti, Y. Wu, and S. Nakamura, "Remot: A model-agnostic refinement for multiple object tracking," *Image Vis Comput*, vol. 106, p. 104091, 2021.

[57] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *ICRA*, 2021, pp. 13 708–13 715.

[58] L. Zheng, M. Tang, Y. Chen, G. Zhu, J. Wang, and H. Lu, "Improving multiple object tracking with single object tracking," in *CVPR*, 2021, pp. 2453–2462.

[59] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," in *CVPR*, 2021, pp. 12 372–12 382.

[60] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, "Multiple object tracking with correlation learning," in *CVPR*, 2021, pp. 3876–3886.

[61] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon, "Learning to track with object permanence," in *ICCV*, 2021, pp. 10 860–10 869.

[62] E. Yu, Z. Li, S. Han, and H. Wang, "Relationtrack: Relation-aware multiple object tracking with decoupled representation," *TMM*, pp. 1—-1, 2022.

[63] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *CVPR*, 2022, pp. 8844–8854.

[64] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking with transformer," in *ECCV*, 2022, pp. 659–675.

[65] Z. Zhao, Z. Wu, Y. Zhuang, B. Li, and J. Jia, "Tracking objects as pixel-wise distributions," in *ECCV*, 2022, pp. 76–94.

[66] Y. Zhang, T. Wang, and X. Zhang, "Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors," in *CVPR*, 2023, pp. 22 056–22 065.

[67] K. Yi, K. Luo, X. Luo, J. Huang, H. Wu, R. Hu, and W. Hao, "Ucmctrack: Multi-object tracking with uniform camera motion compensation," *arXiv:2312.08952*, 2023.

[68] M. Yang, G. Han, B. Yan, W. Zhang, J. Qi, H. Lu, and D. Wang, "Hybrid-sort: Weak cues matter for online multi-object tracking," *arXiv:2308.00783*, 2023.

[69] Y.-H. Wang, J.-W. Hsieh, P.-Y. Chen, M.-C. Chang, H. H. So, and X. Li, "Smiletrack: Similarity learning for occlusion-aware multiple object tracking," *arXiv:2211.08824*, 2023.