Modelling Ambiguous Assignments for Multi-Person Tracking in Crowds

Daniel Stadler[1,2,3] Jürgen Beyerer[2,1,3]
[1]Karlsruhe Institute of Technology [2]Fraunhofer IOSB [3]Fraunhofer Center for Machine Learning
{daniel.stadler,juergen.beyerer}@iosb.fraunhofer.de

Abstract

Multi-person tracking is often solved with a tracking-by-detection approach that matches all tracks and detections simultaneously based on a distance matrix. In crowded scenes, ambiguous situations with similar track-detection distances occur, which leads to wrong assignments. To mitigate this problem, we propose a new association method that separately treats such difficult situations by modelling ambiguous assignments based on the differences in the distance matrix. Depending on the numbers of tracks and detections, for which the assignment task is determined ambiguous, different strategies to resolve these ambiguous situations are proposed. To further enhance the performance of our tracking framework, we introduce a camera motion-aware interpolation technique and make an adaptation to the motion model, which improves identity preservation. The effectiveness of our approach is demonstrated through extensive ablative experiments with different detection models. Moreover, the superiority w.r.t. other trackers is shown on the challenging MOT17 and MOT20 datasets, where state-of-the-art results are obtained.

1. Introduction

Multi-person tracking (MPT) demands the localization and identification of all targets throughout a video sequence and is a basic component for several applications like human activity detection or surveillance related tasks.

The predominant methodology to solve the MPT problem is the *tracking-by-detection* paradigm [4, 5, 24, 38, 42, 48, 50, 51]. An object detector is applied in every frame of a sequence and the generated detections are associated to the current tracks based on a distance measure. For example, the Intersection over Union (IoU) of detection and track box is often used. Mostly, the Hungarian method [18] is leveraged for solving the assignment problem.

While the association task is easy if targets are far away from each other, the assignment of detections to tracks can be ambiguous in crowded scenes, where persons have similar spatial positions. We therefore argue that it is promising to treat those ambiguous situations separately and develop an association method which explicitly models ambiguous assignments by looking more closely at the distance matrix of all possible track-detection matches.

We determine ambiguous situations by introducing a similarity constraint, which indicates for two possible track-detection matches whether they are similar, and scan the distance matrix for all similar assignments. Depending on the numbers of detections and tracks that lead to similar assignments, *i.e.*, whether there are more tracks or more detections, we propose different strategies to resolve the ambiguous situations. For example, we find that one simple yet effective method of handling so-called *ambiguous detections* (there are less detections than tracks) is to simply delete those detections and propagate the involved tracks with the motion model until there are clear matches again. The reason for the success of this deletion strategy lies in the fact that ambiguous detection boxes are often located between two tracks covering parts of both targets due to the inaccuracy of the detectors in crowded scenes. For the other kind of ambiguous situations, *i.e.*, there are more detections than tracks which lead to ambiguous assignments, the best investigated method is an initialization strategy, that suppresses duplicate detections by demanding a target to be detected multiple consecutive frames to start a new track.

After resolving all ambiguous situations, the standard assignment process applying the Hungarian method is performed for the remaining tracks and detections. Note that the proposed association method with the ambiguous assignments modelling needs only a distance matrix and a predefined similarity threshold as input. Therefore, it can be included in any tracking-by-detection based approach, independent from how the distance matrix is calculated.

Besides the approach of modelling and resolving ambiguous assignments, we find that trackers from literature either include an interpolation mechanism for occluded track boxes and neglect the influence of moving cameras [28], or apply a camera motion compensation (CMC) model [2]. To take advantage of both modules, we develop a camera motion-aware interpolation technique that transforms start and end box of the interpolation into a common frame using the transformation matrices coming from the CMC model before the interpolation is performed.

We observe another problem caused by inaccurate detection boxes right before track inactivation due to occlusion: The change of box size, in particular of box height, is often overestimated in the motion model which makes propagated inactive track boxes shrink or grow too fast. As a consequence, re-identification after occlusion fails and identity conservation is harmed. To counteract this, we make an adaptation in the motion model that preserves the height of inactive track boxes during propagation.

The main contributions of our work are summarized in the following:

- We propose a novel association method for tracking-by-detection based approaches, that models ambiguous assignments by searching for possible track-detection matches with similar distances.
- Depending on the numbers of tracks and detections leading to ambiguous assignments, we investigate different strategies to resolve the ambiguous situations.
- A camera motion-aware interpolation technique is introduced and an adaptation to the motion model is made to further improve the tracking performance.

2. Related Work

Tracking-by-detection. Most of the multi-person tracking (MPT) approaches from the literature follow the tracking-by-detection paradigm [4, 5, 24, 38, 42, 48, 50, 51], which splits the MPT task into two sub-problems: detection and association. With many detection models publicly available, most of the MPT research aims at improving the association task, for example, by designing advanced distance measures [11, 36, 40, 51]. One of the most used distance measures for assigning detections to tracks is based on IoU as in the SORT framework [4]. The further development DeepSORT [48] additionally uses visual information of objects extracting appearance features with a separate convolutional neural network, which is also done in many other works [24, 42, 51]. For motion prediction of targets, often a Kalman filter is used in MPT. When additionally the camera is moving, another important component next to the targets motion model is the compensation of camera motion [2], which makes the estimated position of propagated track boxes more accurate. Besides position, motion, and appearance information, human poses can be leveraged in MPT [42, 50]. Moreover, instead of only extracting information of single objects, some tracking methods also consider the relations between the targets [24, 39, 51].

While the design of advanced distance measures, which combine information of several object cues [11, 36, 40, 51], lead to an improved association performance, there still remain ambiguous situations, in that the distance measures give no clear picture about which of the possible track-detection matches are correct. Some approaches follow a hierarchical association scheme [1, 37, 44], first matching detection boxes to short tracklets and afterwards matching on the tracklets level, to mitigate this issue. However, those methods cannot be processed in an online manner, which makes them unsuitable for real-time applications. In contrast, we propose a new online association method, that models and resolves ambiguous assignments needing only the current distance matrix of track-detection pairs as input.

Handling ambiguities. A common strategy to handle ambiguous situations in the association process is to follow a multiple hypothesis tracking (MHT) approach [17, 19, 53], where multiple association hypotheses are maintained for several time steps to find the optimal solution. However, MHT often comes with a high computational complexity which increases exponentially with the number of considered time steps. As already mentioned, another idea to overcome ambiguities is to pursue a hierarchical association scheme [1, 37, 44], in which first short high confidence tracklets are generated before they are merged to longer trajectories. In [29], split-merge conditions are introduced to deal with missing detections of occluded tracks. The special feature of our ambiguous assignments modelling is that ambiguities are determined with the distance matrix of possible track-detection matches and therefore can be included in any tracking-by-detection approach.

Track interpolation. Many methods apply a simple linear interpolation of track boxes to close the gaps of recovered tracks after occlusion [15, 28, 29, 30]. While this works well in scenes with limited motion dynamics, the interpolated boxes are inaccurate when severe camera motion occurs. In [6], single-object trackers like KCF [14] or Medianflow [16] are used for handling fragmented tracks. However, this brings a computational overhead and the single-object trackers also suffer from occlusion so that the interpolation might fail. The cyclic pseudo-observation trajectory filling strategy from [12] incorporates camera motion in the interpolation process, however, future frames are needed for motion prediction. Thus, the interpolation can only be performed as post-processing. In contrast, our interpolation method needs no future information and is performed immediately when occluded tracks are recovered.

3. Proposed Method

We first describe our approach to find ambiguous assignments in Section 3.1. Then, different methods for resolving ambiguous assignments are proposed in Section 3.2. Furthermore, we introduce a camera motion-aware interpolation module and an adapted motion model in Section 3.3.

3.1. Modelling Ambiguous Assignments

In each time step t of a tracking-by-detection based method, the generated detections $\mathcal{D}^t = [\mathrm{D}_1, \ldots, \mathrm{D}_N]$ are assigned to the tracks from the previous iteration $\mathcal{T}^{t-1} = [\mathrm{T}_1, \ldots, \mathrm{T}_M]$ based on a distance matrix $\mathbf{D} \in \mathbb{R}^{M \times N}$. When targets are far away from each other and clearly visible, the assignment task is easy. However, in crowded scenes, the association of detections to tracks can become *ambiguous*, *e.g.*, because of missing detections, and thus, the risk for tracking errors is high. Instead of treating all possible matches equally by applying the Hungarian method [18] on the full distance matrix $\mathbf{D}$, we propose to handle ambiguous assignments separately, and after that, apply the Hungarian method only on a reduced distance matrix $\mathbf{D}_{\text{clear}}$ with remaining *clear assignments*.

To find ambiguous situations, we first search for *similar assignments* by comparing the distances of possible track-detection matches and introduce a similarity threshold Δ. For example, if the distances of the two best matching tracks T_i and T_j w.r.t. a detection D_k differ by less than Δ (and both distances are below a maximum distance $d_{\max}$ – here ignored for clarity), those possible matches belong to the set of similar assignments $\mathcal{A}^{\text{s}}$:

$$|\mathbf{D}[i,k] - \mathbf{D}[j,k]| < \Delta \iff (\{i,j\},\{k\}) \in \mathcal{A}^{\text{s}} \tag{1}$$

D_k is termed an *ambiguous detection* as it is not clear to which track the detection should be assigned. Similarly, there can be an *ambiguous track* T_l if for the best matching detections D_m and D_n w.r.t. T_l the following holds:

$$|\mathbf{D}[l,m] - \mathbf{D}[l,n]| < \Delta \iff (\{l\},\{m,n\}) \in \mathcal{A}^{\text{s}} \tag{2}$$

While these two examples illustrate the idea of similar assignments, note that in crowded scenes and depending on the choice of Δ, similar assignments can include both multiple detections and tracks. In this case, the rows and columns of the distance matrix $\mathbf{D}$ have to be scanned multiple times in order to get the complete set of similar assignments $\mathcal{A}^{\text{s}}$.

Finally, the set of *ambiguous assignments* $\mathcal{A}^{\text{a}}$ is the subset of $\mathcal{A}^{\text{s}}$, where the numbers of detections and tracks differ. We do not keep similar assignments with equal numbers of detections and tracks as ambiguous assignments, since in such cases all detections and tracks can be matched. Thus, the relation of $\mathcal{A}^{\text{s}}$ and $\mathcal{A}^{\text{a}}$ can be expressed as:

$$A \in \mathcal{A}^{\text{s}} \implies A \in \mathcal{A}^{\text{a}} \iff |A[0]| \neq |A[1]| \tag{3}$$

The complete process, how ambiguous assignments are determined with the distance matrix $\mathbf{D}$, the similarity threshold Δ, and a maximum allowed distance $d_{\max}$, can be found in Algorithm 1. Additionally, we provide a toy example distance matrix with highlighted similar, ambiguous, and clear assignments in Figure 1. The similar assignments with equal numbers of tracks and detections (orange) are not treated separately but resolved together with the clear assignments (teal) with the Hungarian method. Unassigned tracks are not terminated immediately but turn *inactive* for at most $i_{\max}$ time steps and unassigned detections start new tracks. To resolve the ambiguous assignments, different

Algorithm 1: Modelling Ambiguous Assignments

Input: Distance matrix $\mathbf{D} \in \mathbb{R}^{M \times N}$ of M tracks and N detections
Similarity threshold Δ, maximum distance $d_{\max}$
Output: Set of ambiguous assignments $\mathcal{A}^{\text{a}}$

$\mathcal{A}^{\text{s}} \leftarrow \emptyset$; $\mathcal{A}^{\text{a}} \leftarrow \emptyset$ `// sim. / amb. assignm.`
for $i = 1 \ldots N$ **do** `// iterate over all dets`
  dets $\leftarrow \emptyset$; tracks $\leftarrow \emptyset$
  `// start with best match`
  dist $\leftarrow$ *min*($\mathbf{D}[:, i]$) `// for comparison`
  sim_dets $\leftarrow \{(i, \text{dist})\}$ `// (idx, distance)`
  do `// find similar assignments`
    dets $\leftarrow$ dets $\cup$ sim_dets
    sim_tracks $\leftarrow \emptyset$
    `// find similar tracks`
    for n, dist $\in$ sim_dets **do**
      t_idx $\leftarrow$ *where*($\mathbf{D}[:, n]$ − dist $< \Delta$ $\wedge\, \mathbf{D}[:, n] < d_{\max}$)
      t_dists $\leftarrow \mathbf{D}$[t_idx, n]
      for idx, d $\in$ *zip*(t_idx, t_dists) **do**
        sim_tracks $\leftarrow$ sim_tracks $\cup$ {(idx, d)}
    tracks $\leftarrow$ tracks $\cup$ sim_tracks
    `// find similar detections`
    sim_dets $\leftarrow \emptyset$
    for m, dist $\in$ sim_tracks **do**
      d_idx $\leftarrow$ *where*($\mathbf{D}[m, :]$ − dist $< \Delta$ $\wedge\, \mathbf{D}[m, :] < d_{\max}$)
      d_dists $\leftarrow \mathbf{D}[m$, d_idx]
      for idx, d $\in$ *zip*(d_idx, d_dists) **do**
        sim_dets $\leftarrow$ sim_dets $\cup$ {(idx, d)}
  while sim_dets $\setminus$ dets $\neq \emptyset$
  `// keep idx of sim. dets / tracks`
  $\mathcal{A}^{\text{s}} \leftarrow \mathcal{A}^{\text{s}} \cup \{(\{t[0] \mid t \in \text{tracks}\}, \{d[0] \mid d \in \text{dets}\})\}$
`// merge sets of sim. dets / tracks`
for $A_k \in \mathcal{A}^{\text{s}}$ **do**
  for $A_l \in \mathcal{A}^{\text{s}} \setminus \{A_k\}$ **do**
    if $(A_k[0] \cap A_l[0]) \vee (A_k[1] \cap A_l[1])$ **then**
      $A_k \leftarrow (A_k[0] \cup A_l[0], A_k[1] \cup A_l[1])$
      $\mathcal{A}^{\text{s}} \leftarrow \mathcal{A}^{\text{s}} \setminus \{A_l\}$
`// save ambiguous assignments`
for A in $\mathcal{A}^{\text{s}}$ **do**
  $n_{\text{T}} = |A[0]|$; $n_{\text{D}} = |A[1]|$
  if $n_{\text{T}} \neq n_{\text{D}}$ **then**
    $\mathcal{A}^{\text{a}} \leftarrow \mathcal{A}^{\text{a}} \cup \{A\}$

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
T_1	1	1	1	0.38	1	1	1	1
T_2	1	1	1	1	1	1	1	0.87
T_3	1	0.34	1	1	1	0.31	1	1
T_4	1	1	1	0.31	1	1	1	1
T_5	0.92	1	1	1	0.06	1	1	1
T_6	1	1	0.11	1	1	1	0.19	1
T_7	1	0.29	1	1	1	0.37	1	1

$$\mathcal{A}^{\mathrm{s}} = \{(\{1,4\},\{4\}), (\{3,7\},\{2,6\}), (\{6\},\{3,7\})\}$$
$$\mathcal{A}^{\mathrm{a}} = \{(\{1,4\},\{4\}), (\{6\},\{3,7\})\}$$

Figure 1: Illustration of a toy example distance matrix $\mathbf{D}$ and resulting similar and ambiguous assignments $\mathcal{A}^{\mathrm{s}}$ and $\mathcal{A}^{\mathrm{a}}$, respectively, after the proposed ambiguous assignments modelling with $\Delta = 0.1$ and $d_{\max} = 0.8$. T_6 is an ambiguous track (cyan) and D_4 is an ambiguous detection (purple). The orange colored similar assignments are not ambiguous, as the numbers of detections and tracks is equal. A clear match $(\mathrm{T}_5, \mathrm{D}_5)$ is highlighted in teal color. Note that $\mathbf{D}[2,8] = 0.87$ exceeds the maximum distance $d_{\max}$. Thus, T_2 turns *inactive* and D_8 starts a new track.

methods are developed which are described in Section 3.2.

Our technique of modelling ambiguous assignments can be applied on any distance matrix $\mathbf{D}$ and therefore incorporated in any tracking-by-detection based method. For simplicity, we use the Intersection over Union (IoU) as matching criterion yielding $\mathbf{D} = \mathbb{I} - \mathrm{IoU}(\mathcal{T}^{t-1}, \mathcal{D}^t)$.

3.2. Resolving Ambiguous Assignments

Each element of the ambiguous assignments $A = (A[0], A[1]) \in \mathcal{A}^{\mathrm{a}}$, where $A[0]$ and $A[1]$ are the sets of track indices and detection indices of A, respectively, can be divided into two subsets. Either the number of tracks $n_{\mathrm{T}} = |A[0]|$ is larger than the number of detections $n_{\mathrm{D}} = |A[1]|$ (ambiguous detections – purple example in Figure 1) or vice versa (ambiguous tracks – cyan example in Figure 1). For each of the two cases, we propose three different strategies to resolve the ambiguous assignments.

Resolving ambiguous detections ($\mathbf{n_D < n_T}$). Ambiguous detections mostly emerge in crowded scenes, where the detector cannot recognize all objects due to heavy occlusion. (1) The first strategy is to *delete* these ambiguous detections arguing that the quality of such detections might be poor. Consequently, the involved tracks become inactive until there are clear matches again. (2) Another possibility is to perform a multiple hypothesis tracking (*mht*) approach for the detections and tracks of the ambiguous assignments. The (ten best) hypotheses are updated in each time step until they can be resolved by clear matches. (3) The third strategy is to allow *multiple* assignments of detections so that two tracks competing for a detection can be both updated with it at the same time. In the *standard* association, detections are assigned to the best matching tracks with the Hungarian method and unassigned tracks turn inactive.

Resolving ambiguous tracks ($\mathbf{n_T < n_D}$). Ambiguous tracks occur when multiple detections fit well to a track. This is the case for duplicate false positive detections, but also when new partly-occluded targets occur for the first time. To resolve ambiguous tracks, we investigate the following strategies. (1) *Deleting* the detections and inactivating the involved tracks. (2) Following a *mht* approach (ten best hypotheses maintained) until hypotheses can be resolved by clear matches. (3) Applying an *initialization* strategy in that unmatched detections start *tentative* tracks, which have to be confirmed in n_{active} consecutive frames until activation – otherwise they are deleted. The *standard* procedure assigns to each track a detection with the Hungarian method and unassigned detections start new tracks.

3.3. Improved Interpolation and Motion Model

Many trackers apply a simple linear interpolation of track boxes when an occluded track is recovered and potential camera motion is neglected. We find that this simplification leads to bad interpolation results when the camera is moving and therefore propose a camera motion-aware interpolation technique, that makes use of a CMC model. Instead of copying the last detection box into the frame, in which the target is recovered, and generating the interpolated boxes without considering camera motion (standard linear interpolation), the following steps are performed:

1. The last detection box from frame t_k is transformed with the transformation matrices $\mathbf{W}_{k+1}, \dots, \mathbf{W}_m$ coming from a CMC model to the middle frame t_m and the recovered box from frame t_l is transformed with the inverse transformation matrices $\mathbf{W}_l^{-1}, \dots, \mathbf{W}_{m+1}^{-1}$ to the middle frame t_m with $m = \lceil (l-k)/2 \rceil$.
2. Linear interpolation of the transformed boxes is done.
3. The interpolated boxes are transformed back to their respective frames.

Note that a transformation matrix $\mathbf{W}_n$ describes the motion from frame $n-1$ to frame n. A visual illustration of our camera motion-aware interpolation technique and the standard linear interpolation is depicted in Figure 2.

As linear motion model for track propagation, we follow [48] using a Kalman filter with track state $\mathrm{T.state} = (x, y, a, h, \dot{x}, \dot{y}, \dot{a}, \dot{h})$, where x, y, a, h are the track box vertical position, horizontal position, aspect ratio, and height,

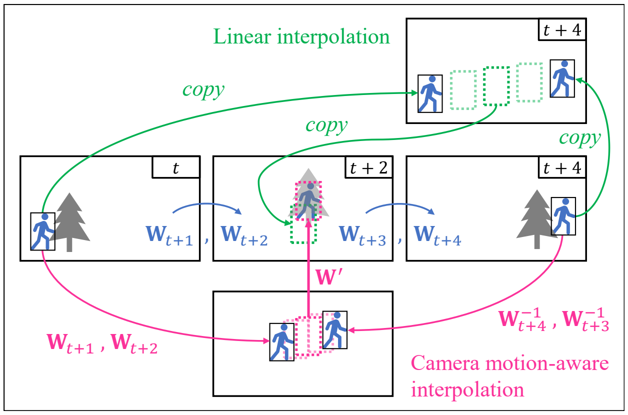

Figure 2: Standard linear interpolation (green) and the proposed camera motion-aware interpolation (magenta) that considers the transformation matrices $\{\mathbf{W}_n\}$. The reverse transformations of the interpolated boxes are summarized with $\mathbf{W}'$ for clarity. For example, the interpolated box of frame $t+1$ is transformed with $\mathbf{W}' = \mathbf{W}_{t+2}^{-1}$ and for the box of frame $t+2 = m$, $\mathbf{W}'$ is the identity matrix. Note that for gaps with more than three frames, multiple transformation matrices have to be applied one after another in the reverse transformation of interpolated boxes.

respectively. While this model works well with precise detection boxes, we find it vulnerable w.r.t. low quality detections in crowded scenes, which is depicted in Figure 3. The last associated detection box, before a track turns inactive due to severe occlusion, might only contain body parts leading to an inaccurate bounding box which negatively affects the motion estimation in the Kalman filter. In our observations, a wrong height of the bounding box causes the most problems. Therefore, we propose to preserve the height of a track whenever it turns inactive by setting $\dot{h}$ to zero.

The full pipeline of our tracker is listed in Algorithm 2. Contributions are highlighted with colors, which are in correspondence with Figures 1, 2, and 3. Note that the shown algorithm is the variant, where ambiguous tracks are resolved by applying the initialization strategy with tentative tracks as described in Section 3.2.

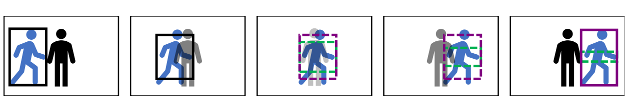

Figure 3: Motion prediction with (violet) and without (green) height preservation. Active track boxes are depicted in solid lines, propagated inactive track boxes in dashed lines. With height preservation ($\dot{h} = 0$), the track can be continued after occlusion. Otherwise, the low quality box in the second frame makes the propagated track box shrink in each frame ($\dot{h} < 0$), so that a re-activation fails.

Algorithm 2: Proposed tracking pipeline at time t

Input: Set of previous tracks $\mathcal{T}^{t-1} = \{\mathrm{T}_1, \ldots, \mathrm{T}_M\}$
Set of current detections $\mathcal{D}^t = \{\mathrm{D}_1, \ldots, \mathrm{D}_N\}$
Camera trafo matrices $\mathrm{W} = [\mathbf{W}_1, \ldots, \mathbf{W}_F]$
Kalman filter (KF) with noise cov. $\mathbf{Q}$ and $\mathbf{R}$
Similarity threshold Δ, maximum distance $d_{\max}$
Activation count n_{active}, inactive patience $i_{\max}$
Output: Updated set of tracks $\mathcal{T}^t$

for $\mathrm{T} \in \mathcal{T}^{t-1}$ **do** `// apply motion models`
  $\mathrm{T} \leftarrow$ camera_motion_compensation$(\mathrm{T}, \mathbf{W}_t)$
  T.mean, T.cov $\leftarrow$ KF.predict(T.mean, T.cov, $\mathbf{Q}$)
$\mathbf{D} = \mathbb{I} - \mathrm{IoU}(\mathcal{T}^{t-1}, \mathcal{D}^t)$ `// distance matrix`
`// find and resolve amb. assignm.`
ass_t_idx $\leftarrow$ [ ]; ass_d_idx $\leftarrow$ [ ]
$\mathcal{A}^{\mathrm{a}} \leftarrow$ find_amb_assignm$(\mathbf{D}, \Delta, d_{\max})$ `// Alg. 1`
for $A \in \mathcal{A}^{\mathrm{a}}$ **do** `// strategies from Sec. 3.2`
  t_idx $\leftarrow A[0]$; d_idx $\leftarrow A[1]$
  if |d_idx| < |t_idx| **then** `// missing dets`
    t, d $\leftarrow$ res_amb_dets$(\mathcal{T}^{t-1}, \mathcal{D}^t$, d_idx, t_idx)
  else if |d_idx| > |t_idx| **then** `// miss. tracks`
    t, d $\leftarrow$ res_amb_tracks$(\mathcal{T}^{t-1}, \mathcal{D}^t$, d_idx, t_idx)
  ass_t_idx $\leftarrow$ ass_t_idx + t
  ass_d_idx $\leftarrow$ ass_d_idx + d
`// assign clear matches`
unass_t_idx = $[x|x \in \mathrm{range}(M) : x \notin$ ass_t_idx]
unass_d_idx = $[y|y \in \mathrm{range}(N) : y \notin$ ass_d_idx]
$\mathbf{D}_{\text{clear}} = \mathbf{D}$[unass_t_idx, unass_d_idx]
t, d $\leftarrow$ clear_assignm$(\mathbf{D}_{\text{clear}}, d_{\max})$
ass_t_idx $\leftarrow$ ass_t_idx + t
ass_d_idx $\leftarrow$ ass_d_idx + d
for $i, \mathrm{T} \in \mathcal{T}^{t-1}$ **do** `// update tracks`
  if $i \in$ ass_t_idx **then** `// assigned det`
    $j \leftarrow$ ass_d_idx[ass_t_idx.index(i)]
    T.mean, T.cov $\leftarrow$ KF.update(T, D_j, $\mathbf{R}$)
    if T.state = *inactive* **then** `// Fig. 2`
      cma_inter(T.mean, T.last_p, T.n_inac, W)
      T.state $\leftarrow$ *active*; T.n_inac $\leftarrow 0$
    T.last_p $\leftarrow$ T.mean; T.n_det $\leftarrow$ T.n_det + 1
    if T.n_det = n_{active} **then**
      T.state $\leftarrow$ *active*
  else `// no assigned det`
    T.n_inac $\leftarrow$ T.n_inac + 1
    if T.state = *tentative* $\vee$ T.n_inac > $i_{\max}$ **then**
      $\mathcal{T}^{t-1} \leftarrow \mathcal{T}^{t-1} \setminus \{\mathrm{T}\}$ `// remove`
    if T.state = *active* **then**
      T.state $\leftarrow$ *inactive*
      T.mean[8] $\leftarrow 0$ `// ḣ = 0, Fig. 3`
$\mathcal{T}^t \leftarrow \mathcal{T}^{t-1}$ `// save updated tracks`
for $j, \mathrm{D} \in \mathcal{D}^t$ **do** `// start new tracks`
  if $j \notin$ ass_d_idx **then**
    $\mathrm{T}^{\text{new}} \leftarrow$ KF.initiate(D)
    $\mathcal{T}^t \leftarrow \mathcal{T}^t \cup \{\mathrm{T}^{\text{new}}\}$

4. Experiments

4.1. Datasets

MOT17. The MOT17 dataset [27] comprises 14 diverse sequences for multi-person tracking, 7 for training and testing each, including videos with both static as well as moving cameras. As the annotations for the test split are not publicly available, we follow [33, 34, 45, 49, 57] and divide the train split into two halves for ablative experiments.

MOT20. A more recent version of the MOTChallenge (*https://motchallenge.net/*) is MOT20 [9], which focuses on tracking in very crowded scenes. It consists of a train and a test split with 4 sequences each. For ablative experiments, the train split is also divided into two halves.

CrowdHuman. As one of the largest datasets for person detection, the CrowdHuman dataset [35] is frequently used for pre-training. It is divided into three splits – train (15000 images), validation (4370 images), and test (5000 images).

4.2. Evaluation Metrics

For evaluating tracking performance, we use MOTA [3], which incorporates numbers of false positives (FP), false negatives (FN), and identity switches (IDSW), as well as IDF1 [32], and the recently proposed HOTA [26]. Besides that, numbers of mostly tracked (MT) and mostly lost (ML) targets as well as number of fragmentations (FRAG) are reported. TrackEval [25] is used for calculating all metrics.

4.3. Implementation Details

Tracker. The parameters of our tracker are empirically set as follows: The threshold for the distances of two possible track-detection matches to be considered as similar is $\Delta = 0.1$. The minimum required IoU $o_{\min}$ between a track and a detection box for matching is 0.2. Thus, the maximum allowed distance gets $d_{\max} = 1 - o_{\min} = 0.8$. The number of consecutive detections for a tentative track to become active, and the number of frames, a track is kept as inactive without assigned detection before termination, are set to $n_{\text{active}} = 4$ and $i_{\max} = 40$, respectively. As motion model, a Kalman filter with the implementation of [48] is applied. For CMC on MOT17, the Enhanced Correlation Coefficient Maximization from [10] is leveraged. On MOT20, neither CMC nor the proposed camera motion-aware interpolation are performed, since there is hardly any camera motion in the sequences of MOT20.

Detector. Unless otherwise specified, a Faster R-CNN detector [31] with FPN [22] as neck and ResNet-50 [13] as backbone is used in ablative experiments. The model is pre-trained on CrowdHuman train split for 30 epochs with a batch size of 16 and a learning rate of 0.01, which is reduced by factor 10 after epochs 24 and 27. After that, fine-tuning on the first half of MOT17 train is conducted with an initial learning rate of 0.001 for another 30 epochs with the same schedule. When testing our tracker on the second half of MOT20 train, the first half of MOT20 train is taken for fine-tuning instead. In the comparison with the state-of-the-art on MOT17 / 20, the respective full train splits are taken for fine-tuning. The only used data augmentation is horizontal flipping. We also run experiments with RetinaNet [23] and the crowd-specific detector CrowdDet [8] applying the same neck, backbone, and training schedules. Faster RCNN and CrowdDet are trained with the implementation of [8] and MMDetection [7] is used for training RetinaNet. The non-maximum suppression threshold is set to 0.5, the minimum score threshold for detections kept in tracking is 0.9 for Faster RCNN and CrowdDet and 0.6 for RetinaNet.

4.4. Ablation Studies

Modelling ambiguous assignments. We first run experiments with the various strategies to resolve ambiguous detections and tracks proposed in Section 3.2. The tracking results are summarized in Table 1. The first line corresponds to the standard association, where no ambiguous assignments are modelled and all tracks and detections are assigned at once using the Hungarian method.

For resolving *ambiguous detections*, the strategy of *deletion* and waiting for clear matches yields the by far best results boosting MOTA, IDF1, and HOTA by 0.6, 1.3, and 0.8 points, respectively. The *mht* approach improves identity preservation, but MOTA is not enhanced. The strategy of allowing *multiple* associations does not work, as too many duplicate detections are introduced (FP increased by 70%). A qualitative example, in that the deletion of an ambiguous detection prevents an identity switch, is given in Figure 4.

As can be seen in Table 1, for resolving *ambiguous tracks*, the *deletion* strategy does only enhance identity preservation measured in IDF1, but MOTA is reduced due to an increased number of missing detections. In contrast,

Table 1: Tracking results with different strategies for resolving ambiguous detections and tracks.

Amb. dets	Amb. tracks	MOTA	IDF1	HOTA
standard	standard	74.0	76.9	62.9
delete	standard	74.6	78.2	63.7
mht	standard	73.2	77.3	63.2
multi	standard	66.3	74.6	60.8
standard	delete	73.1	77.2	62.8
standard	mht	74.4	76.6	62.8
standard	init	75.7	78.0	63.6
delete	init	**76.5**	**79.4**	**64.5**

(a) Standard association.

(b) With deletion of ambiguous detections.

Figure 4: (b) With the deletion of an ambiguous detection (dotted box) in the middle frame and waiting for clear assignments, all targets are correctly tracked. (a) In contrast, an identity switch occurs in the standard association because the assignment of the inaccurate ambiguous detection box distorts the motion estimation of the assigned track. Note that inactive tracks are marked with dashed lines.

the *mht* approach increases MOTA but lowers IDF1. The *initialization* technique, however, significantly improves all tracking measures: +1.7 MOTA, +1.1 IDF1, +0.7 HOTA. This is because duplicate detections often occur only in single frames, which do not initialize wrong tracks due to the tentative track state. At the same time, hardly any correct detections are removed with the initialization strategy.

The last line of Table 1 shows, that the deletion strategy for resolving ambiguous detections and the initialization strategy for resolving ambiguous tracks bring complementary gains, as their combination leads to further great improvements: The proposed modelling of ambiguous assignments and separate treatment of ambiguous detections and tracks raises both MOTA and IDF1 by 2.5 points and HOTA by 1.6 points w.r.t. the standard association.

Improved interpolation and motion model. We ablate the influence of the proposed camera motion-aware interpolation as well as the height preservation in the motion model in Table 2. The first two lines show the importance of interpolation for the final tracking performance, as even a linear interpolation significantly improves HOTA,

Table 2: Impact of standard linear interpolation (LI) and proposed camera motion-aware interpolation (CMAI) as well as height preservation (HP) in the motion model.

LI	CMAI	HP	MOTA	IDF1	HOTA	FN	FP
✗	✗	✗	72.2	75.9	61.1	37116	**7131**
✓	✗	✗	73.2	76.1	62.6	24798	18036
✗	✓	✗	73.5	76.3	62.7	24570	17766
✗	✓	✓	**74.0**	**76.9**	**62.9**	**23694**	17859

IDF1, and MOTA by greatly reducing the number of FN. In comparison to the standard approach, our camera motion-aware interpolation achieves both lower values of FN and FP showing that the interpolated boxes are more accurate. This holds especially for sequences with severe camera motion, *e.g.*, MOT17-13, in which the camera motion-aware interpolation improves MOTA by 2.1 points over the standard interpolation. Note that the used transformation matrices do not have to be calculated separately, but come from the CMC model and thus, the computational overhead w.r.t. the linear interpolation is negligible. The height preservation of inactive tracks enhances the accuracy of propagated track boxes which leads to a further improvement in identity preservation (+0.6 points), which is also beneficial for MOTA (+0.5 points) and HOTA (+0.2 points).

Results with various detectors and datasets. To demonstrate the generalization abilities of our tracker w.r.t. different detectors and datasets, we run several experiments applying detections from Faster RCNN, RetinaNet, and CrowdDet on MOT17 and MOT20. The results of these experiments are listed in Table 3. For each combination,

Table 3: Improvements with various detectors and datasets. The base tracker assigns all track-detection pairs with the Hungarian method and uses a standard linear interpolation.

Tracker	Dataset	Detector	IDF1	HOTA	MOTA
Base	MOT17	RetinaNet	66.3	54.5	59.3
Ours	MOT17	RetinaNet	69.6	56.8	62.9
Base	MOT17	FRCNN	76.1	62.6	73.2
Ours	MOT17	FRCNN	79.4	64.5	76.5
Base	MOT17	CrowdDet	78.2	64.3	73.5
Ours	MOT17	CrowdDet	80.6	65.6	76.7
Base	MOT20	RetinaNet	56.3	48.2	71.1
Ours	MOT20	RetinaNet	59.4	49.8	74.2
Base	MOT20	FRCNN	74.6	60.7	84.5
Ours	MOT20	FRCNN	75.2	61.0	84.4
Base	MOT20	CrowdDet	78.6	63.7	85.7
Ours	MOT20	CrowdDet	80.0	64.5	85.8

Table 4: State-of-the-art methods on MOT17 / 20 test set using private detections. Entries are sorted with ascending MOTA.

MOT17

Method	MOTA ↑	IDF1 ↑	HOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDSW ↓	FRAG ↓
Semi-TCL [20]	73.3	73.2	59.8	972	441	**22944**	124980	2790	8010
FairMOT [56]	73.7	72.3	59.3	1017	408	27507	117477	3303	8073
PermaTrack [43]	73.8	68.9	55.5	1032	405	28998	115104	3699	6132
RelationTrack [54]	73.8	74.7	61.0	981	546	27999	118623	**1374**	2166
CSTrack [21]	74.9	72.6	59.3	978	411	23847	114303	3567	7668
GRTU [46]	74.9	75.0	62.0	1170	444	32007	107616	1812	**1824**
TransTrack [41]	75.2	63.5	54.1	1302	**240**	50157	86442	3603	4872
TPAGT [34]	76.2	68.0	57.9	1203	321	32796	98475	3237	5658
CorrTracker [45]	76.5	73.6	60.7	1122	300	29808	99510	3369	6063
MAATrack (*ours*)	**79.4**	**75.9**	**62.0**	**1356**	282	37320	**77661**	1452	2202

MOT20

Method	MOTA ↑	IDF1 ↑	HOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDSW ↓	FRAG ↓
MLT [55]	48.9	54.6	43.2	384	274	45660	216803	2187	3067
TransCenter [52]	58.5	49.6	43.5	603	185	64217	146019	4695	9581
FairMOT [56]	61.8	67.3	54.6	**855**	**94**	103440	**88901**	5243	7874
TransTrack [41]	65.0	59.4	48.9	622	167	27191	150197	3608	11352
Semi-TCL [20]	65.2	70.1	55.3	761	131	61209	114709	4139	8508
LCC [58]	66.0	67.0	53.2	699	165	43938	129584	2237	4154
CSTrack [21]	66.6	68.6	54.0	626	192	25404	144358	3196	7632
GSDT [47]	67.1	67.5	53.6	660	164	31913	135409	3131	9875
RelationTrack [54]	67.2	70.5	56.5	773	111	61134	104597	4243	8236
MAATrack (*ours*)	**73.9**	**71.2**	**57.3**	741	153	**24942**	108744	**1331**	**1450**

we also apply a baseline tracker, that performs a standard association with the Hungarian method and uses a linear interpolation. One can see, that there are significant improvements w.r.t. the baseline among all detectors and on both datasets. The largest improvements are observed using detections from RetinaNet with gains of 3.3 (3.1), 2.3 (1.6), and 3.6 (3.1) points in terms of IDF1, HOTA, and MOTA, respectively, on MOT17 (MOT20). As expected, the tracking performance improves consistently by applying better detection models, whereby the overall best tracking results are achieved with the crowd-specific model CrowdDet. Note that, for example on MOT17, the detection results of the three models measured in average precision at an IoU threshold of 0.5 (AP_{50}) are 78.3 for RetinaNet, 86.5 for Faster RCNN, and 87.8 for CrowdDet.

4.5. Comparison with the State-of-the-Art

As the best results have been achieved in combination with CrowdDet, we keep it as detection model when applying our method on the test sets of MOT17 and MOT20. The results of our tracker termed *MAATrack* (Modelling Ambiguous Assignments), generated by the official evaluation server, are compared against the state-of-the-art in Table 4. MAATrack achieves the overall best tracking performance among all methods with large gains compared to the second best entry CorrTracker [45] / RelationTrack [54] in terms of MOTA (+2.9 / +6.7), IDF1 (+2.3 / +0.7), and HOTA (+1.3 / +0.8) on MOT17 / 20. Furthermore, the best values of MT and FN are obtained on MOT17. On MOT20, MAATrack has the least numbers of FP, IDSW, and FRAG. The superior results show that a separate treatment of ambiguous assignments is beneficial for multi-target tracking, especially in crowds, as most tracking errors occur in such situations.

5. Conclusion

We develop a new association method for multi-target tracking in crowded scenes, which explicitly models ambiguous assignments of detections and tracks and treats those separately from clear track-detection matches. To resolve these ambiguous situations, different strategies are investigated. Moreover, we introduce two additional modules, a camera motion-aware interpolation technique and an adapted motion model, to further improve tracking performance. The effectiveness of our approach is shown with ablative experiments and state-of-the-art results are obtained on two popular benchmarks for multi-person tracking.

References

[1] N. M. Al-Shakarji, F. Bunyak, G. Seetharaman, and K. Palaniappan. Multi-object tracking cascade with multi-step data association and occlusion handling. In *IEEE Int. Conf. Adv. Video Sign. Surv.*, 2018.

[2] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. Tracking without bells and whistles. In *Int. Conf. Comput. Vis.*, pages 941–951, 2019.

[3] K. Bernardin, A. Elbs, and R. Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment. In *Eur. Conf. Comput. Vis. Worksh.*, 2006.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *IEEE Int. Conf. Image Process.*, pages 3464–3468, 2016.

[5] E. Bochinski, V. Eiselein, and T. Sikora. High-speed tracking-by-detection without using image information. In *IEEE Int. Conf. Adv. Video Sign. Surv.*, 2017.

[6] E. Bochinski, T. Senst, and T. Sikora. Extending iou based multi-object tracking by visual information. In *IEEE Int. Conf. Adv. Video Sign. Surv.*, 2018.

[7] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. MMDetection open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

[8] X. Chu, A. Zheng, X. Zhang, and J. Sun. Detection in crowded scenes: One proposal, multiple predictions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12211–12220, 2020.

[9] 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.

[10] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1858–1865, 2008.

[11] W. Feng, Z. Hu, W. Wu, J. Yan, and W. Ouyang. Multi-object tracking with multiple cues and switcher-aware classification. *arXiv:1901.06129*, 2019.

[12] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, X. Pan, and J. Zhao. Mat: Motion-aware multi-object tracking. *arXiv:2009.04794*, 2020.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015.

[15] M. Hofmann, M. Haag, and G. Rigoll. Unified hierarchical multi-object tracking using global data association. In *IEEE Int. Worksh. Perform. Eval. Track. Surv.*, pages 22–28, 2013.

[16] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Int. Conf. Pattern Recog.*, pages 2756–2759, 2010.

[17] C. Kim, F. Li, and J. M. Rehg. Multi-object tracking with neural gating using bilinear lstm. In *Eur. Conf. Comput. Vis.*, pages 200–215, 2018.

[18] H. W. Kuhn and B. Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[19] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Int. Conf. Comput. Vis.*, 2007.

[20] W. Li, Y. Xiong, S. Yang, M. Xu, Y. Wang, and W. Xia. Semi-tcl: Semi-supervised track contrastive representation learning. *arXiv:2107.02396*, 2021.

[21] C. Liang, Z. Zhang, Y. Lu, X. Zhou, B. Li, X. Ye, and J. Zou. Rethinking the competition between detection and reid in multi-object tracking. *arXiv:2010.12138*, 2020.

[22] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 936–944, 2017.

[23] Tsung-Yi Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Int. Conf. Comput. Vis.*, pages 2999–3007, 2017.

[24] Q. Liu, Q. Chu, B. Liu, and N. Yu. Gsm: Graph similarity model for multi-object tracking. In *IJCAI*, pages 530–536, 2020.

[25] J. Luiten and A. Hoffhues. Trackeval. https://github.com/JonathonLuiten/TrackEval, 2020.

[26] J. Luiten, A. Osep, P. Dendorfer, P. H. S. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129(2):548–578, 2021.

[27] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016.

[28] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6308–6318, 2020.

[29] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 666–673, 2006.

[30] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1306–1313, 2014.

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[32] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Eur. Conf. Comput. Vis. Worksh.*, pages 17–35, 2016.

[33] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould. Probabilistic tracklet scoring and inpainting for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14329–14339, 2021.

[34] C. Shan, C. Wei, B. Deng, J. Huang, X.-S. Hua, X. Cheng, and K. Liang. Tracklets predicting based adaptive graph tracking. *arXiv:2010.09015*, 2020.

[35] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv:1805.00123*, 2018.

[36] S. Sharma, J. A. Ansari, J. Krishna Murthy, and K. Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *IEEE Int. Conf. on Rob. Auto.*, pages 3508–3515, 2018.

[37] H. Shen, L. Huang, C. Huang, and W. Xu. Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking. *arXiv:1808.01562*, 2018.

[38] A. Specker, D. Stadler, L. Florin, and J. Beyerer. An occlusion-aware multi-target multi-camera tracking system. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 4173–4182, 2021.

[39] D. Stadler and J. Beyerer. Improving multiple pedestrian tracking by track management and occlusion handling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10958–10967, 2021.

[40] D. Stadler, L. W. Sommer, and J. Beyerer. Pas tracker: Position-, appearance- and size-aware multi-object tracking in drone videos. In *Eur. Conf. Comput. Vis. Worksh.*, pages 604–620, 2020.

[41] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo. Transtrack: Multiple object tracking with transformer. *arXiv:2012.15460*, 2021.

[42] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person reidentification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3701–3710, 2017.

[43] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon. Learning to track with object permanence. In *Int. Conf. Comput. Vis.*, pages 10860–10869, 2021.

[44] G. Wang, Y. Wang, H. Zhang, R. Gu, and J. Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *ACM Int. Conf. Multimedia*, pages 482–490, 2019.

[45] Q. Wang, Y. Zheng, P. Pan, and Y. Xu. Multiple object tracking with correlation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3876–3886, 2021.

[46] S. Wang, H. Sheng, Y. Zhang, Y. Wu, and Z. Xiong. A general recurrent tracking framework without real data. In *Int. Conf. Comput. Vis.*, pages 13219–13228, 2021.

[47] Y. Wang, K. Kitani, and X. Weng. Joint Object Detection and Multi-Object Tracking with Graph Neural Networks. *arXiv:2006.13164*, 2020.

[48] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE Int. Conf. Image Process.*, pages 3645–3649, 2017.

[49] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan. Track to detect and segment: An online multi-object tracker. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12352–12361, 2021.

[50] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *Eur. Conf. Comput. Vis.*, pages 472–487, 2018.

[51] J. Xu, Y. Cao, Z. Zhang, and H. Hu. Spatial-temporal relation networks for multi-object tracking. In *Int. Conf. Comput. Vis.*, pages 3987–3997, 2019.

[52] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv:2103.15145*, 2021.

[53] L. Ying, T. Zhang, and C. Xu. Multi-object tracking via mht with multiple information fusion in surveillance video. *Multim. Sys.*, 21(3):313–326, 2015.

[54] E. Yu, Z. Li, S. Han, and H. Wang. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *arXiv:2105.04322*, 2021.

[55] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, and Z. Xiong. Multiplex labeling graph for near-online tracking in crowded scenes. *IEEE Internet Things J.*, 7(9):7892–7902, 2020.

[56] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129:3069–3087, 2021.

[57] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking objects as points. In *Eur. Conf. Comput. Vis.*, pages 474–490, 2020.

[58] Z. Zou, J. Huang, and P. Luo. Compensation tracker: Reprocessing for lost object. *arXiv:2008.12052*, 2021.