A Detailed Study of the Association Task in Tracking-by-Detection-based Multi-Person Tracking

Daniel Stadler

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
daniel.stadler@kit.edu

Abstract

Many multi-person trackers follow the tracking-by-detection paradigm applying a person detector in each frame and linking detections of the same target to form tracks in the association task. While the basic concept is the same among these methods, various motion models, distance metrics to measure the similarity of targets, and matching strategies are used. This makes it difficult to compare different methods and also to assess the influence of single tracking components on the final performance. For these reasons, all parts of the association task are thoroughly investigated in this study. Starting with a simple baseline which is consequently improved with the help of experimental results, a strong tracking-by-detection-based framework is developed that achieves state-of-the-art performance on two multi-person tracking benchmarks.

1 Introduction

The objective of multi-person tracking (MPT) is to detect and identify all persons in every frame of a given video. Applications range from crowd monitoring to autonomous driving and surveillance related tasks.

To solve the MPT problem, most methods pursue the tracking-by-detection (TBD) paradigm. A detector is applied on each image independently and the obtained detection sets are matched such that detections of the same target form a track with a unique ID. This problem of assigning the correct detections to the corresponding tracks is called the association task. While some approaches try to integrate detection and association more tightly [2, 9, 31, 33, 43], the strict separation of the two sub-tasks in TBD can still achieve state-of-the-art results. Currently, the top performing entries of the standard MPT benchmarks MOT17 [22] and MOT20 [6] follow the TBD paradigm [1, 7, 23, 28, 46] leveraging an of-the-shelf detection model and focusing on the association task.

Different strategies to improve the association can be observed in the literature. Motion models based on Kalman filter [16] are used to make the estimated target positions more accurate [4, 5, 10, 40]. In addition, camera motion compensation techniques are integrated to deal with motion of non-static cameras [2, 10, 15, 28, 29]. The core of the association is the distance measure which determines how likely a detection belongs to a so-far tracked target. On the one hand, motion-based metrics such as Intersection over Union (IoU) are utilized and on the other hand, the appearance of targets is leveraged. For example, in DeepSORT [40] and its further development StrongSORT [10], a person re-identification model is applied to extract appearance features from the image patches of the detections and cosine distance between the high-dimensional features is taken as association metric. While DeepSORT uses motion distance only for gating, i.e., prohibiting unlikely assignments, StrongSORT combines it with appearance distance as also done in [1, 7]. Besides the distance metric, the association strategy has a large influence on the performance. While most methods make all assignments at once with the Hungarian algorithm [17], DeepSORT proposes a matching cascade that prefers previously observed targets and ByteTrack [46] performs a second matching step in which low-confident detections are utilized.

In this study, all aforementioned components of the association task in MPT are analyzed in detail. Starting with a baseline TBD approach with strong motion models, a large number of experiments with different distance measures, both motion- and appearance-based, and their combinations are conducted. In addition, matching strategies with multiple stages are investigated. With the help of the experimental results, a strong TBD method is developed which achieves

state-of-the-art results on the two MPT datsets MOT17 [22] and MOT20 [6]. Furthermore, ablative experiments of the proposed framework are performed showing the influence of the single tracking components as well as the sensitivity of the tracking parameters on the final performance.

2 Baseline and Motion Models

A baseline tracker using only IoU as matching metric is built before more advanced matching measures and strategies are investigated. In addition, various motion models for target and camera motion are compared in this section.

Let $\mathcal{T}^{t-1} = \{T_1^{t-1}, \ldots, T_k^{t-1}\}$ be the tracks found until frame I^{t-1} and $\mathcal{D}^t = \{D_1^t, \ldots, T_l^t\}$ the detections generated on the frame I^t of a video $V = [I^1, \ldots, I^n]$ of length n. The association task is to assign the detections $\mathcal{D}^t$ to its corresponding targets $\mathcal{T}^{t-1}$. For this, distances between all confident detections and tracks are calculated and used as cost values. Afterwards, the overall costs of assignments are minimized, e.g., with the Hungarian algorithm [17]. More precisely, given a detection $D = (B_D, s) \in \mathcal{D}_t$ with box B_D and confidence s and the box B_T of a track $T \in \mathcal{T}^{t-1}$, a distance measure d can be calculated using the IoU between detection box B_D and track box B_T:

$$d_{\mathrm{IoU}} = 1 - \mathrm{IoU}(B_{\mathrm{D}}, B_{\mathrm{T}}) \tag{2.1}$$

Before calculation of the distance matrix of detections and tracks, the detections are filtered w.r.t. confidence, i.e., detections with a score s smaller than the threshold s_{track} are removed and not used in the association. In addition, a maximum distance $d_{\max}$ is enforced to prohibit unlikely assignments. Unmatched tracks that are not assigned a detection become *inactive* and are kept for $i_{\max}$ frames in the set of tracks before deletion. Thus, they can be re-activated for a short time period to bridge occlusions, for instance. Unmatched detections with high confidence $s \geq s_{\mathrm{init}}$ start new tracks. Note that some trackers [10, 28, 40, 46] follow an initialization strategy, in which detections first start tentative tracks that have to be confirmed in subsequent frames in order to become active. While this strategy suppresses frame-wise false positive detections, it introduces false negatives since the tentative tracks do not contribute to the tracking output.

If the quality of detections is high and a large threshold s_{init} is set, such an initialization technique can reduce the overall performance, so it is not used in this study unless otherwise stated.

Most MPT approaches have in common, that a Kalman filter (KF) [16] is used to model the motion of targets. However, various formulations of the state vector $\mathbf{x}$ and different implementation details can be found in the MPT literature. The most used variants are originally from the SORT [4] and DeepSORT [40] frameworks. The state vectors of the two KF types are as follows:

$$\mathbf{x}_{\text{SORT}} = (u, v, a, r, \dot{u}, \dot{v}, \dot{a})^{\mathsf{T}} \tag{2.2}$$

$$\mathbf{x}_{\text{DeepSORT}} = (u, v, r, h, \dot{u}, \dot{v}, \dot{r}, \dot{h})^{\mathsf{T}} \tag{2.3}$$

The box center position is (u, v) and the aspect ratio is $r = w/h$ with w and h denoting box width and height, respectively. A derivative of a variable x with respect to time is indicated by $\dot{x}$. Whereas SORT explicitly models the box area $a = w \cdot h$ and its derivative $\dot{a}$ but keeps the aspect ratio r fixed, DeepSORT instead models the box height h and its derivative $\dot{h}$. Thus, the process and measurement noise covariance matrices also differ next to other implementation details, which can be found in the papers [4, 40] or the public source code.

Recently, further developments have been proposed for the DeepSORT variant – the Noise Scale Adaptive (NSA) KF [8] and the height preservation (HP) adaptation [30]. In the update step of the NSA KF, the measurement noise covariance matrix $\mathbf{R}$ is weighted with the confidence of the measurement, i.e., the detection confidence score s, as follows:

$$\mathbf{R}_{\text{NSA}} = (1 - s) \cdot \mathbf{R} \tag{2.4}$$

The higher the detection confidence, the smaller the adapted measurement noise covariance $\mathbf{R}_{\text{NSA}}$ and the more influence has the detection on the track state update. The other adaptation is related to the state vector $\mathbf{x}$. It is empirically found in [30], that predicting inactive tracks for multiple frames without state update, the track box size can change dramatically which hinders re-activation after occlusion. To prevent this, HP can be applied simply setting the derivative $\dot{h}$ to zero before the KF prediciton step, which is also done in [1] and [46].

Besides target motion, modelling camera motion is also important. For camera motion compensation (CMC), again two different methods from literature are

Table 2.1: Motion Model Results.

KF Type	NSA	CMC	HOTA	KF Type	NSA	CMC	HP	HOTA
SORT	✗	✗	67.61	DeepSORT	✗	✗	✗	67.40
SORT	✓	✗	67.67	DeepSORT	✓	✗	✗	67.82
SORT	✗	ECC	67.77	DeepSORT	✗	ECC	✗	68.03
SORT	✗	ORB	**68.36**	DeepSORT	✗	ORB	✗	68.13
SORT	✓	ECC	68.03	DeepSORT	✓	ORB	✗	68.62
SORT	✓	ORB	68.35	DeepSORT	✓	ORB	✓	**68.67**

investigated – the Enhanced Correlation Coefficient (ECC) Maximization [12] and a model from [28] that is based on the ORB [26] feature detector and the RANSAC [13] algorithm. The ORB method is a sparse image registration technique in that foreground objects like moving persons can be neglected, in contrast to the global ECC method. A similar approach is found in [1].

To compare the different motion models, several experiments are run on the validation split (Val) of MOT17, which is created by dividing the train sequences into two halves and using the second ones [35, 46, 48]. As detection model, a publicly available YOLOX [14] model from [46] is utilized, which has been trained on a combined dataset consisting of CrowdHuman [27], CityPersons [45], ETH [11], and the first half of MOT17 train split. Note that this YOLOX model can be regarded as the current standard in MPT on the MOT datasets, since many state-of-the-art methods are using it [1, 10, 5, 28, 23, 37, 36, 46]. If not otherwise stated, the parameters of the tracker are set to $s_{\text{init}} = 0.7$, $s_{\text{track}} = 0.6$, $d_{\max} = 0.8$, $i_{\max} = 30$ and the resolution of the input images is 1440×1080 pixels. To measure the overall tracking accuracy, HOTA [20] is evaluated.

The results with different KF types, KF adaptations and CMC models are summarized in Table 2.1. Without any extensions, the SORT KF performs slightly better than the DeepSORT KF. However, the results of the DeepSORT KF can be largely improved with the NSA adaptation, while NSA in combination with SORT does not enhance the results in all configurations. This is not surprising, as NSA is developed as extension for the DeepSORT KF and the measurement noise covariance matrices $\mathbf{R}$ differ among the KF types. As

expected, ORB outperforms ECC in all experiments. W.r.t. the baselines, ORB improves the overall tracking performance by 0.75 HOTA and 0.73 HOTA for SORT KF and DeepSORT KF, respectively. Additionally adding the height preservation (HP) in the DeepSORT KF variant, a HOTA of 68.67 is achieved which is a gain of 1.27 HOTA in comparison to the DeepSORT KF baseline. Therefore, the DeepSORT KF with NSA and HP extensions is used in all subsequent experiments, together with the CMC model based on ORB features.

3 Distance Measures

As mentioned previously, the distance measure is the core of each TBD algorithm. In the baseline experiments of the last section, the IoU has been leveraged which is the most used motion-based distance metric in MPT. In this section, further distance measures for the association are explored. First, motion-based matching is analyzed in Section 3.1. Then, appearance-based matching is studied in Section 3.2. Both types of infomation are combined in Section 3.3, before further techniques like incorporating the detection confidence and applying gating mechanisms are treated in Sections 3.4 and 3.5, respectively.

3.1 Motion-based Matching

The authors of SimpleTrack [18] experiment with the Generalized IoU (GIoU) [24] as similarity measure in combination with appearance information, which enhances the performance of their tracker. This raises the question, whether other IoU related measures also can improve the matching accuracy. Therefore, different adaptations of the original IoU are investigated in the following. Given two boxes $A = (x_A, y_A, w_A, h_A)$ and $B = (x_B, y_B, w_B, h_B)$, the IoU is the relation of the intersection $A \cap B$ to the union $A \cup B$:

$$\text{IoU} = \frac{A \cap B}{A \cup B} \tag{3.1}$$

The IoU has the drawback that non-overlapping boxes always yield an IoU of 0, independent from how far away the boxes are from each other. To solve this

issue, the GIoU is proposed as

$$\mathrm{GIoU} = \mathrm{IoU} - \frac{C \setminus (A \cup B)}{C} \tag{3.2}$$

where C denotes the smallest enclosing box of A and B. While the spatial distance of the boxes A and B has influence on the box C, it is not modelled explicitly. In contrast, the euclidean distance $d_{\mathrm{L2}}(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$ is directly used in the Distance IoU (DIoU) [47]:

$$\mathrm{DIoU} = \mathrm{IoU} - \frac{d^2_{\mathrm{L2}}(A, B)}{c^2} \tag{3.3}$$

Here, c denotes the diagonal of the smallest enclosing box C. The same paper further introduces the Complete IoU (CIoU) [47], which not only explicitly models spatial distance but also aspect ratio consistency:

$$\mathrm{CIoU} = \mathrm{DIoU} - \alpha v \tag{3.4}$$

$$v = \frac{4}{\pi^2}\left(\arctan\left(\frac{w_A}{h_A}\right) - \arctan\left(\frac{w_B}{h_B}\right)\right)^2 \tag{3.5}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v} \tag{3.6}$$

Note that the IoU and its variants are similarity measures with a maximum similarity of 1. Thus, a distance measure can be created by subtracting the value from 1 as in Equation 2.1.

If a Kalman filter is used as motion model, it is possible to integrate the uncertainty of the motion estimation into the distance measure. For this, DeepSORT [40] and StrongSORT [10] calculate the squared Mahalanobis distance between a detection D and a track T, given the state formulation of the detection box $\mathbf{d} = (u, v, r, h)^\mathsf{T}$ and the projection of the track state (mean and covariance) into measurement space $(\mathbf{y}, \mathbf{S})$:

$$d_{\mathrm{mahal}} = (\mathbf{d} - \mathbf{y})^\mathsf{T} \mathbf{S}^{-1} (\mathbf{d} - \mathbf{y}) \tag{3.7}$$

Whereas DeepSORT uses only the Mahalanobis distance for gating, i.e., preventing unlikely assignments by enforcing a maximum distance, here, d_{mahal} is directly used as matching distance. Additionally, the euclidean distance d_{L2} between detection and track center is considered.

Table 3.1: Motion-based Matching Results.

d	IoU	GIoU	DIoU	CIoU	L2	Mahal
HOTA	68.67	68.47	**68.74**	**68.74**	64.70	62.94

To compare the performance of the aforementioned motion-based distance measures d in the association, several experiments are conducted tuning the maximum distance threshold $d_{\max}$ for each metric separately. The highest achieved HOTA values are reported in Table 3.1. One can see that the IoU-based distance measures work much better than taking the L2 distance or the Mahalanobis distance. While L2 distance does not consider the important information of box dimensions, the Mahalanobis distance is only a rough estimation of the object location if the state uncertainty is high [40]. In the experimental setup, DIoU and CIoU achieve the highest HOTA value of 68.74, closely followed by IoU and GIoU. Note that DIoU and CIoU yield the *exactly* same tracking results. Since the aspect ratio of targets does not vary significantly in MPT, v in Equation 3.4 becomes a very small value, thus $\mathrm{CIoU} \approx \mathrm{DIoU}$ holds. For this reason, the DIoU is used in the rest of this study.

3.2 Appearance-based Matching

Similar to adopting an of-the-shelf detector, many MPT approaches take over a model from the re-identification community for extracting appearance features of targets [1, 10, 19, 34, 40, 41]. Such a network takes a small image patch of a detected person as input and computes a high-dimensional feature vector that represents the appearance of the person. In appearance-based matching, several design choices have to be made when comparing the features of detections and tracks. Which distance measure should be used? How many time steps shall be considered to describe the appearance of a track? What is the best way to combine features from different time steps? In this section, a large amount of experiments is conducted to answer these questions empirically. Given two m-dimensional feature vectors f_D and f_T from a detection and a track, respectively, one can calculate either the cosine distance $d_{\cos}$ or the euclidean

distance d_{L2} to measure their appearance similarity:

$$d_{\cos} = 1 - \frac{f_D \cdot f_T^{\mathsf{T}}}{\|f_D\| \cdot \|f_T\|} \tag{3.8}$$

$$d_{\mathrm{L2}} = \sqrt{(f_{D,1} - f_{T,1})^2 + (f_{D,2} - f_{T,2})^2 + \cdots + (f_{D,m} - f_{T,m})^2} \tag{3.9}$$

Note that $d_{\cos} \in [0, 2]$ and $d_{\mathrm{L2}} \in [0, \infty]$ holds and $\|\cdot\|$ represents the Euclidean norm. Studying the source code of a few appearance-based MPT methods, it is observed that some methods apply a mask to the cosine distance matrix before solving the assignment problem with the Hungarian method. More precisely, all entries above the maximum distance threshold $d_{\max}$ are set to $d_{\max} + \epsilon$ with ϵ being a very small value, e.g., $1\mathrm{e}^{-5}$. This causes unlikely assignments with a distance above the matching threshold $d_{\max}$ to have the same contribution to the overall cost that is minimized by the Hungarian algorithm.

While the detection feature f_D is simply the output of the re-identification model, there are multiple possibilities to build the track feature f_T. In the simplest case, the feature f_D^{t-1} from the last assigned detection D^{t-1} of the track $T^t = [D^{t_{\mathrm{init}}}, \ldots, D^{t-2}, D^{t-1}]$ is used as track feature: $f_T = f_D^{t-1}$. To benefit from temporal information, DeepSORT [40] builds a feature bank $F_T = [f_D^{t-N}, \ldots, f_D^{t-2}, f_D^{t-1}]$ with the features of the past N time steps. The distance to a current detection feature f_D^t is calculated for each feature of the bank. The appearance distance $d(D, T)$ between a detection D and a track T is then chosen to be the minimum of all distances derived from the feature bank:

$$d_{\min}(D, T) = d_{\min}(D, F_T) = \min_{i \in [1,\ldots,N]} d(f_D^t, f_D^{t-i}) \tag{3.10}$$

If the target is clearly visible both in one of the last N frames and the current frame, the extracted features are of high quality and taking the minimum appearance distance is a good choice. However, this is not always the case in MPT, especially when facing severe occlusions. In such situations, the mean distance might be a better choice:

$$d_{\mathrm{mean}}(D, T) = d_{\mathrm{mean}}(D, F_T) = \frac{1}{N} \sum_{i}^{N} d(f_D^t, f_D^{t-i}) \tag{3.11}$$

Moreover, it is possible to average the two measures, which results in a third strategy to calculate the appearance distance between a detection and a track:

$$d_{\text{mean+min}}(D,T) = \frac{1}{2}(d_{\text{mean}}(D,F_T) + d_{\text{min}}(D,F_T)) \tag{3.12}$$

The last investigated strategy for computing the appearance distance is adopted from [38]. Instead of using a feature bank, the track feature f_T is updated in an exponential moving average (EMA) fashion with the newly assigned detection feature f_D^t and a weighting factor α in each time step:

$$f_T^t = \alpha f_T^{t-1} + (1-\alpha) f_D^t \tag{3.13}$$

The re-identification model from [1] is leveraged for feature extraction in the experimental evaluation. It is a BoT (SBS) [21] model with ResNeSt50 [44] as backbone, trained on the first half of MOT17 [22] train split. The performance of the aforementioned appearance-based distance measures and strategies is again compared on the MOT17 Val split, whereby the maximum distance threshold $d_{\max}$ is optimized for each configuration separately. For experiments using the EMA technique, the corresponding parameter α is also tuned.

The resulting HOTA values are reported in Table 3.2. One can see that masking the distance matrix is beneficial for cosine distance but not euclidean (L2) distance. With masking, cosine distance outperforms L2 distance by 0.36 HOTA. Taking $N = 10$ past time steps in a feature bank into account, the results improve significantly by 1.40 to 2.25 points, depending on the strategy of the distance calculation. This shows the importance of temporal information in appearance-based matching. The best results are achieved by averaging the mean and minimum distance of the features (mean+min). Increasing the number of features yields improvements up to $N = 10$, while HOTA values decrease again using 20 or even 100 features. The EMA strategy achieves competitive results but HOTA is 0.25 points worse than the best configuration – the mean+min strategy with $N = 10$ past features and masked cosine distance – which achieves 68.72 HOTA. Note that the overall performance of the appearance-based matching is on par with the motion-based matching from the previous section (Table 3.1). However, on indiviual sequenves of the dataset, differences in HOTA up to 4 points are observed. This motivates the combination of motion- and appearance-based matching which is investigated in the next section.

Table 3.2: Appearance-based Matching Results.

d	Masking	N	Strategy	EMA	HOTA
Cosine	✗	1	✗	✗	66.19
Cosine	✓	1	✗	✗	66.47
L2	✗	1	✗	✗	66.11
L2	✓	1	✗	✗	66.03
Cosine	✓	10	min	✗	67.87
Cosine	✓	10	mean	✗	68.20
Cosine	✓	10	mean+min	✗	**68.72**
Cosine	✓	1	mean+min	✗	66.47
Cosine	✓	2	mean+min	✗	67.25
Cosine	✓	5	mean+min	✗	68.26
Cosine	✓	10	mean+min	✗	**68.72**
Cosine	✓	20	mean+min	✗	68.60
Cosine	✓	100	mean+min	✗	68.05
Cosine	✓	1	✗	✓	68.47

3.3 Combined Matching

Motion- and appearance-based distance measures provide different types of information. Thus, combining both kinds to an advanced distance measure is a promising approach which is also followed in other works [1, 10, 18]. Given two distance measures d_1, d_2 and corresponding weights w_1, w_2, a combined distance d_{comb} can simply be built by a weighted sum:

$$d_{\mathrm{comb}} = w_1 d_1 + w_2 d_2 \tag{3.14}$$

For motion information, the IoU-based distance measures d_{IoU}, d_{GIoU} and d_{DIoU} are considered, while the feature cosine distance $d_{\cos}$ is used for appearance information. Experiments with different configurations are conducted on MOT17 Val. Note that the maximum distance threshold $d_{\max}$ is adjusted when changing distance measures or one of the weights w_1 or w_2. The resulting HOTA values are listed in Table 3.3. The previously achieved results using either motion-

Table 3.3: Combined Matching Results.

d_1	d_2	w_1	w_2	HOTA	d_1	d_2	w_1	w_2	HOTA
IoU	✗	✗	✗	68.67	IoU	Cosine	1	2	69.16
GIoU	✗	✗	✗	68.39	IoU	Cosine	1	3	69.13
DIoU	✗	✗	✗	68.74	IoU	Cosine	1	4	69.22
Cosine	✗	✗	✗	68.72	IoU	Cosine	1	5	69.04
IoU	Cosine	1	1	68.91	GIoU	Cosine	1	4	69.37
IoU	Cosine	2	1	68.62	DIoU	Cosine	1	4	**69.41**

or appearance-based information are also given for reference. The best results are very similar with $\mathrm{HOTA} = 68.74$ for DIoU and $\mathrm{HOTA} = 68.72$ for cosine distance, which justifies the usage of both cues. Combining IoU distance and cosine distance with equal contribution ($w_1 = w_2 = 1$), HOTA improves to 68.91. Giving more weight to the motion-based measure ($w_1 = 2$, $w_2 = 1$), the performance decreases. However, if the appearance information is taken more into account ($w_1 = 1$, $w_2 > 1$), HOTA can be further enhanced up to 69.22 for $w_2 = 4$. The same holds true for combining GIoU or DIoU distance with appearance cosine distance. The largest HOTA value of 69.41 is obtained by combining DIoU distance and cosine distance while setting $w_1 = 1$ and $w_2 = 4$, i.e., giving four times the weight to the appearance information. This is a gain of 0.69 points in HOTA compared to using only one distance measure.

Note that experiments have also been conducted with the Mahalanobis distance d_{mahal} (Equation 3.7) in combination with the appearance cosine distance as it is done in StrongSORT [10]. The highest achieved HOTA in the experimental setup is 69.13. While this is also an improvement w.r.t. using only appearance information, the performance is worse than combining DIoU distance with the appearance cosine distance. Therefore, the combination of DIoU distance and cosine distance is utilized in the remainder of this study.

Table 3.4: Use Detection Confidence Results.

d	d_{score}	HOTA	d	d_{score}	HOTA	d	d_{score}	HOTA
IoU	✗	68.67	DIoU	✗	68.74	DIoU+Cosine	✗	**69.41**
IoU	✓	**68.78**	DIoU	✓	**68.79**	DIoU+Cosine	✓	69.19

3.4 Use of Detection Confidence

Some IoU-based MPT methods incorporate the detection confidence s into the distance calculation by simple multiplication [1, 28, 46]:

$$d_{\text{IoU,score}}(D, T) = 1 - (\text{IoU}(B_D, B_T) \cdot s) \tag{3.15}$$

The motivation behind it is that more confident detections should be favored in the association. Note that this strategy can also be applied together with other IoU-based metrics and its influence is investigated empirically. Because the multiplication of $s \in [s_{\text{track}}, 1]$ changes the scale of the distance measure d, the maximum distance threshold $d_{\max}$ has again been tuned. The results are depicted in Table 3.4. Integrating the detection score into the distance matrix slightly improves HOTA by 0.11 and 0.05 points for IoU and DIoU distance, respectively. However, in combination with the appearance cosine distance, which yields the overall best results, using the detection score degrades the performance. Thus, the detection score is not leveraged in the distance calculation in the remainder of the study.

3.5 Gating

As mentioned before, DeepSORT [40] utilizes the Mahalanobis distance to prevent unlikely assignments which is referred to as gating. The distance measure is only used to prohibit assignments with a distance value above a threshold but is not integrated into the matching distance. In this section, the influence of such a gating mechanism on the tracking performance is analyzed. Besides Mahalanobis distance, IoU, DIoU and appearance cosine distance are tested as gating measures. The combination of DIoU and cosine distance from

Table 3.5: Gating Results.

Gating	✗	IoU	DIoU	Cosine	Mahal
HOTA	69.41	69.45	**69.47**	69.41	69.42

Section 3.3 is taken as distance for matching. Tracking results with additional gating are depicted in Table 3.5. In the experiments, only small HOTA gains up to 0.06 points are achieved, although the gating thresholds have been tuned carefully. For this reason and because a too small gating threshold can degrade the tracking performance, gating is not used in the rest of this work.

4 Multiple Matching Stages

It is the common practice in MPT to solve the assignment problem for all tracks and detections at once as also done in this study so far. However, a few works split the set of tracks or detections into subsets which are processed one after another [1, 28, 30, 40, 46]. Two strategies are revisited – a matching cascade from the famous DeepSORT [40] tracker (Section 4.1) and the BYTE [46] association method which recently lead to notable improvements (Section 4.2).

4.1 DeepSORT Matching Cascade

Given an example track $T^t = [D^{t_{\text{init}}}, \dots, D^{t-k}]$ at time step t, its age a is defined as the time since the track has been observed for the last time. For this example track, $a = k$ holds. Note that in this definition, *active* tracks have an age of 1, whereas *inactive* tracks have an age greater than 1. In DeepSORT [40], tracks with an age of 1 are matched with all available detections. Then, all tracks with an age of 2 are matched with the remaining unmatched detections and so forth. The motivation behind this strategy is to favor tracks that have been observed recently, since the accuracy of propagated track locations decreases over time. However, in StrongSORT [10] – a further development of DeepSORT – it is found that this matching cascade harms the tracking performance when the

Table 4.1: DeepSORT (DS) Matching Cascade Results.

DS Matching Cascade	HOTA	DS Matching Cascade	HOTA
✗	**69.41**	✓	67.86

tracker gets stronger because the additional prior constraints limit the matching accuracy [10]. To investigate the influence of the DeepSORT matching cascade on the so-far best tracker of this study (Section 3.3), it is utilized in an additional experiment. The result is shown in Table 4.1. Integrating the matching cascade significantly decreases HOTA by 1.55 points which confirms the results from [10]. Obviously, this matching cascade is not used in further experiments.

4.2 BYTE Association

Usually, only high-confident detections are used in the association as low-confident ones include many false positives that harm the tracking performance. In contrast, an association technique named BYTE is proposed in [46], which allows to make use of low-confident detections in a second matching stage. Detections with confidence score below s_{track} are not removed but compared to unmatched tracks that have not been assigned a high-confident detection in the first association. Since the low-confident detections are not utilized to start new tracks but only for assignment to already tracked targets, the overall performance can be largely increased. The authors of [46] show this by applying the BYTE association to different trackers which leads to consistent improvements. Among the trackers, the varying distance measures are kept in the first matching stage. However, in the newly introduced second matching stage, only the IoU distance is leveraged as the authors argue that most tracks in this stage suffer from occlusion or motion blur, where appearance features are not reliable [46].

Since the tracking pipeline of this study differs quite a lot from other approaches with the improved motion modelling from Section 3.1 and the combined distance measure from Section 3.3, it is also experimented with appearance-based cosine distance next to other distance measures in the second association stage. Although it is not mentioned in the paper [46], the publicly available source

Table 4.2: Second Matching Results. The best result using only one matching stage is achieved with a combination of DIoU and cosine distance: HOTA = 69.41 (Table 3.3).

Use Inactive	Distance	HOTA	Use Inactive	Distance	HOTA
✗	IoU	69.66	✓	IoU	**70.29**
✗	DIoU	69.70	✓	DIoU	70.22
✗	Cosine	69.68	✓	Cosine	70.22
✗	DIoU+Cosine	69.73	✓	DIoU+Cosine	70.14

code reveals that only *active* tracks are considered in the second matching stage. In this study, it is also tested whether the inclusion of *inactive* tracks in this stage can be beneficial. Resulting HOTA values of the conducted experiments related to the second matching stage can be found in Table 4.2.

In contrast to [46], appearance-based distances like the cosine distance and the combination with DIoU also achieve good results. Compared to the baseline, where only one matching stage is used (HOTA $= 69.41$), gains up to 0.32 HOTA are obtained. Note that the applied distance threshold of the second stage $d_{\max,2}$ influences the performance, so it is tuned carefully for each configuration.

When additionally inactive tracks are used, IoU-based matching results in 70.29 HOTA which is a huge improvement compared to using only active tracks in the second matching stage. It is observed that the optimized distance threshold $d_{\max,2}$ is much lower than in the implementation of [46] (0.19 vs. 0.5). Setting such a low threshold ensures that only inactive tracks with accurately predicted locations can be matched. With the usage of inactive tracks, no prior constraints are applied that could limit the matching accuracy, similar as the matching cascade of DeepSORT (see Section 4.1). Since the IoU-based matching in the second stage yields an improvement of 0.88 HOTA in comparison to the one-stage baseline, it is leveraged in all further experiments.

Table 5.1: Parameter Tuning Results.

	$i_{\max}$	N	Strategy	s_{init}	s_{track}	$d_{\max,1}$	$d_{\max,2}$	HOTA
Before Tuning	30	10	mean+min	0.7	0.6	3.18	0.19	70.29
After Tuning	**28**	**16**	**mean**	0.7	0.6	**3.13**	0.19	**70.77**

5 Parameter Tuning and Sensitivity

Before evaluating different motion models to develop a baseline tracker for this study, some parameters had to be set initially: the number of frames an inactive track is kept ($i_{\max}$), confidence thresholds for detections to be considered in the association and to start new tracks (s_{track} and s_{init}) and the maximum distance threshold to prevent unlikely assignments ($d_{\max,1}$). Extending the tracking framework with additional components, further parameters are introduced. Integrating appearance features (Section 3.2), the number of past time steps in the feature bank (N) and the strategy how to calculate the cosine distance (min, mean, mean+min) have to be chosen. With the utilization of a second matching stage (Section 4.2), another maximum distance threshold has to be set ($d_{\max,2}$). Since the number of parameters has increased during this study, some might not be set optimal anymore. For this reason, an extensive grid search has been performed to find the best parameter configuration of the tracker. The results are summarized in Table 5.1, whereby parameters that have changed are bold. Optimizing the set of parameters gives a notable plus of 0.48 HOTA.

To get a better understanding of the importance and the sensitivity of the tracking parameters, hundreds of experiments have been conducted in that each parameter has been varied within a decent interval around the best value (specified by grid search), while all the other parameters were fixed at their optimum. The resulting HOTA curves are shown in Figure 5.1.

The confidence threshold s_{init} of a detection to initialize a new track obviously has a large influence on the tracking performance. With a too low threshold, many false positives are introduced, whereas with a too large threshold, many targets are missed. The track threshold s_{track} decides, whether a detection is considered

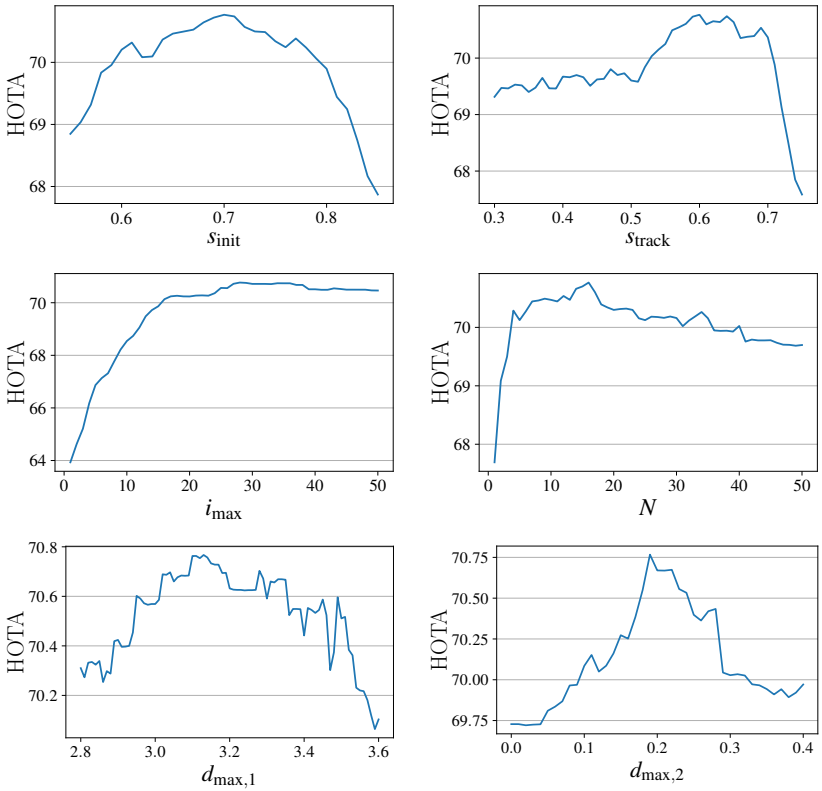

Figure 5.1: Sensitivity of Tracking Parameters.

in the first association stage or the second. Priority is given to detections with confidence above s_{track} and in the second stage, a stricter maximum distance is enforced for the lower-confident detections. In the experiments, $s_{\text{track}} = 0.6$ achieved the best results. This value is 0.1 smaller than s_{init}, which equals the relation in [46].

Another important parameter is i_{max}. The higher the value, the longer the occlusions that can be bridged. If this so-called inactive patience, however, is

too high, wrong assignments to inactive tracks can occur, since the location accuracy decreases over time. For the number of appearance features in the feature bank, the empirically found best value is $N = 16$. If only a few features are considered, the full potential of the temporal information is not leveraged, whereas features from too far in the past might not be representative anymore due to changes in appearance.

The best values for the matching thresholds $d_{\max,1}$ and $d_{\max,2}$ are 3.13 and 0.19, respectively, on MOT17 Val. Too small values prevent correct assignments while too large values allow wrong assignments. The fluctuations in the corresponding HOTA curves are caused by the small depicted HOTA ranges and in addition – like for all parameters – are attributable to the finite dataset size.

6 Post-processing

The so-far developed tracking framework works fully online which means that the tracking results are final after processing each frame of the input video. Some applications without real-time requirements allow to refine the tracking results with post-processing techniques to improve the performance. Besides simple linear interpolation of fragmented tracks, two more sophisticated post-processing methods introduced in StrongSORT [10] are investigated – the Appearance Free Link (AFLink) model and Gaussian Smoothed Interpolation (GSI).

AFLink is a small convolutional neural network that takes the center positions and corresponding frames of two tracks as input and computes a connectivity score solely based on spatio-temporal information. If this connectivity score is higher than a threshold and some spatio-temporal constraints are fulfilled, the two tracks are linked hypothesizing that they belong to the same target. Implementation details can be found in the StrongSORT paper [10].

Since the maximum gap in a fragmented track is $i_{\max} = 28$ (see Table 5.1), which corresponds to roughly one second on MOT17, many of those gaps can be successfully filled with linear interpolation (LI). However, in some cases the linear approximation is not accurate enough. Therefore, GSI employs Gaussian process regression [39] to model non-linear motion of targets. Another

Table 6.1: Post Processing Results.

AFLink	Interpolation	HOTA	AFLink	Interpolation	HOTA
✗	✗	70.77	✓	LI	72.52
✓	✗	70.84	✓	GSI	**72.81**

advantage compared to the linear interpolation is that the noisy trajectories are smoothed. It is referred to [10] for details of the GSI algorithm.

Table 6.1 depicts the post-processing results after application of AFLink as well as linear and Gaussian smoothed interpolation. AFLink slightly improves HOTA by 0.07 points. Since the model does not integrate appearance information, strict spatio-temporal constraints have to be enforced to prevent wrong connections. For potentially larger improvements, more sophisticated approaches like ReMOT [42] could be applied which is left for future work. Based on appearance features enhanced by self-supervised learning, tracks are not only merged in [42], but erroneous tracks consisting of different targets are additionally cut apart. Looking at the results of the two interpolation techniques, it is observed that both significantly improve the overall performance with gains of 1.68 and 1.97 points in HOTA for LI and GSI, respectively. As expected, the non-linear GSI outperforms the simple linear interpolation.

7 Ablation Study

In this work, several components related to the association task in MPT have been investigated and a strong tracking framework based on the TBD paradigm has been developed. Starting from a simple baseline with standard Kalman filter (KF) for track propagation and IoU distance as association metric, extensions of the KF and a camera motion compensation (CMC) module were introduced. Then, motion-based matching was combined with appearance-based matching leading to a sophisticated distance measure. Afterwards, low-confident detections were integrated into the association within a second matching stage. Finally, parameter tuning and post-processing were performed. All these steps lead to consistent

Table 7.1: Ablation Study. Abbreviations: CMC = camera motion compensation, NSA+HP = Noise Scale Adaptive Kalman filter + height preservation, DIoU+Cosine = Distance IoU + cosine distance, PT = parameter tuning, PP = post-processing (AFLink + Gaussian smoothed interpolation).

CMC	NSA+HP	DIoU+Cosine	2^{nd} Matching	PT	PP	HOTA
✗	✗	✗	✗	✗	✗	67.40 (±0.00)
✓	✗	✗	✗	✗	✗	68.13 (+0.73)
✓	✓	✗	✗	✗	✗	68.67 (+0.54)
✓	✓	✓	✗	✗	✗	69.41 (+0.74)
✓	✓	✓	✓	✗	✗	70.29 (+0.88)
✓	✓	✓	✓	✓	✗	70.77 (+0.48)
✓	✓	✓	✓	✓	✓	**72.81** (+2.16)

improvements of the overall tracking performance measured in HOTA that are summarized in Table 7.1. Besides the offline post-processing, the largest gains in the online tracker come from the second matching stage (+0.88 HOTA), the combined distance measure (+0.74 HOTA), and the CMC model (+0.73 HOTA). All components together boost HOTA significantly from 67.40 to 72.81.

8 Comparison with the State-of-the-Art

The final tracker of this study is named *StrongTBD* because of the large improvements w.r.t. the TBD baseline from Section 2. StrongTBD is compared to the state-of-the-art on MOT17 [22] and MOT20 [6] test splits in this section. Before delving into the results, it should be noted that annotations of the test splits are not publicly available and evaluation is done by submitting the tracking results to the official server (*motchallenge.net*). Besides HOTA, other performance measures such as MOTA [3] and IDF1 [25] are also computed. To prevent parameter tuning on the test data, one is restricted to four submissions. However, the tracking performance is highly dependent on the setting of some parameters, especially on the detection thresholds s_{init} and s_{track} (see Section 5). For example, changing s_{init} and s_{track} from 0.7 to 0.4 and 0.6 to 0.3, respectively, MOTA increases by approximately 10 points on the MOT20-08 sequence in the

Table 8.1: State-of-the-Art Methods on MOT17.

Method	MOTA	IDF1	HOTA	FP	FN	IDSW
MAATrack [30]	79.4	75.9	62.0	37320	77661	1452
RTU++ [36]	79.5	79.1	63.9	29508	84618	1302
StrongSORT [10]	79.6	79.5	64.4	27876	86205	1194
SAT [37]	80.0	79.8	64.4	25125	86505	1356
ByteTrack [46]	80.3	77.3	63.1	25491	83721	2196
QuoVadis [7]	80.3	77.7	63.1	25491	83721	2103
FOR_Tracking [23]	80.4	77.7	63.6	28674	79452	2298
BoT-SORT [1]	80.5	80.2	65.0	**22521**	86037	1212
ByteTrackV2 [28]	80.6	78.9	63.6	35208	**73224**	1239
StrongTBD	**81.6**	**80.8**	**65.6**	24171	78759	**954**

Table 8.2: Values of s_{init} on MOT17 and MOT20 test sets.

MOT17	01	03	06	07	08	12	14	MOT20	04	06	07	08
s_{init}	0.8	0.75	0.75	0.7	0.7	0.8	0.65	s_{init}	0.7	0.4	0.7	0.4

submissions of StrongTBD. This and the fact that some works do not report their applied thresholds makes a fair comparison among methods difficult. The trend of using various thresholds for different sequences of the datasets [1, 28, 46] further complicates the comparison.

Nevertheless, Table 8.1 lists the 10 best performing trackers on MOT17 with ascending MOTA values. StrongTBD achieves the highest values in MOTA, IDF1, and HOTA. Furthermore, it has the least number of identity switches (IDSW). Despite the aforementioned comparability issues, the results show that the developed tracker can compete with the state-of-the-art. To make these results reproducible, the s_{init} values of the submission for the sequences of MOT17 are reported in Table 8.2. Note that for the tracking thresholds $s_{\text{track}} = s_{\text{init}} - 0.1$ holds, just as in [1, 28, 46].

Table 8.3: State-of-the-Art Methods on MOT20.

Method	MOTA	IDF1	HOTA	FP	FN	IDSW
SAT [37]	75.0	76.6	62.6	**15549**	113136	**816**
OC-SORT [5]	75.7	76.3	62.4	19067	105894	942
RTU++ [36]	76.5	76.8	62.8	19247	101290	971
FOR_Tracking [23]	76.8	76.4	61.4	27112	91254	1443
ByteTrackV2 [28]	77.3	75.6	61.4	22867	93409	1082
ReMOT [42]	77.4	73.1	61.2	28351	**86659**	1789
ByteTrack [46]	77.8	75.2	61.3	26249	87594	1223
QuoVadis [7]	77.8	75.7	61.5	26249	87594	1187
BoT-SORT [1]	77.8	**77.5**	63.3	24638	88863	1313
StrongTBD	**78.0**	77.0	**63.6**	25473	87330	1101

Table 8.2 also shows the values of s_{init} on the final submission on the MOT20 dataset. The results on this benchmark of the 10 best performing trackers are given in Table 8.3. StrongTBD obtains the highest MOTA and HOTA as well as the second highest IDF1, which confirms the competitiveness of the developed tracking framework. Note that the parameter configuration of StrongTBD has been adapted on the MOT20 dataset in order to be more comparable to the second best entry BoT-SORT [1]. More precisely, the input resolution of the MOT20-04 and MOT20-07 sequences are set to 1600×896 pixels, while a resolution of 1920x736 pixels is used in MOT20-06 and MOT20-08. In addition, an IoU distance threshold of 0.7 is integrated, which helps to prevent IDSW in crowded scenes. Furthermore, the same initialization strategy as in [1, 28, 46] is followed, in that new tracks are tentative until they get confirmed with an assigned detection in the subsequent frame. As already discussed in Section 2, such a strategy is beneficial if the threshold s_{init} is quite low which is the case for MOT20-06 and MOT20-08 (see Table 8.2). The target density on MOT20 with 127 persons per image is much higher than on MOT17 with only 21.1 persons per image [32]. As StrongTBD has been developed on MOT17 Val, some design choices are not optimal for very crowded scenes as in MOT20. In the future, more focus should be put on tracking in such challenging scenarios.

9 Conclusion

In this study, all components of the association task in MPT have been analyzed in detail. Two of the most important findings are that the combination of motion- and appearance-based distance measures outperforms the sole usage of one information type and that leveraging low-confident detections in a second association stage yields significant improvements. The influence of various tracking components from motion models to post-processing techniques has been investigated as well as the sensitivity of the results to the setting of tracking parameters. The empirical results were used to develop a sophisticated tracking-by-detection method that achieves state-of-the-art performance on the two challenging MPT benchmarks MOT17 and MOT20. Further potential lies in enhancing the association accuracy in very crowded scenes as in the MOT20 dataset, which sould be investigated more thoroughly in the future.

References

[1] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky. “BoT-SORT: Robust Associations Multi-Pedestrian Tracking”. In: *CoRR* abs/arXiv:2206.14651 (2022).

[2] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. “Tracking Without Bells and Whistles”. In: *ICCV*. 2019, pp. 941–951.

[3] K. Bernardin, A. Elbs, and R. Stiefelhagen. “Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment”. In: *ECCV Workshops*. 2006.

[4] A. Bewley et al. “Simple online and realtime tracking”. In: *ICIP*. 2016, pp. 3464–3468.

[5] J. Cao et al. “Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking”. In: *CoRR* abs/arXiv:2203.14360 (2022).

[6] P. Dendorfer et al. “MOT20: A benchmark for multi object tracking in crowded scenes”. In: *CoRR* abs/arXiv:2003.09003 (2020).

[7] P. Dendorfer et al. “Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?” In: *CoRR* abs/arXiv:2210.07681 (2022).

[8] Y. Du et al. “GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021”. In: *ICCV Workshops*. 2021, pp. 2809–2819.

[9] Y. Du et al. “Looking Beyond Two Frames: End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers”. In: *CoRR* abs/arXiv:2103.14829 (2021).

[10] Yunhao Du et al. “StrongSORT: Make DeepSORT Great Again”. In: *CoRR* abs/arXiv:2202.13514 (2022).

[11] A. Ess et al. “A mobile vision system for robust multi-person tracking”. In: *CVPR*. 2008.

[12] G. D. Evangelidis and E. Z. Psarakis. “Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.10 (2008), pp. 1858–1865.

[13] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (1981), pp. 381–395.

[14] Z. Ge et al. “YOLOX: Exceeding YOLO Series in 2021”. In: *CoRR* abs/arXiv:2107.08430 (2021).

[15] S. Han et al. “MAT: Motion-aware multi-object tracking”. In: *Neurocomputing* 476 (2022), pp. 75–86.

[16] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *J. basic Eng.* 82.1 (1960), pp. 35–45.

[17] H. W. Kuhn and B. Yaw. “The Hungarian Method for the Assignment Problem”. In: *Naval Research Logist. Quart.* 2.1–2 (1955), pp. 83–97.

[18] J. Li et al. “SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking”. In: *Sensors* 22.15 (2022).

[19] Q. Liu et al. “GSM: Graph Similarity Model for Multi-Object Tracking”. In: *IJCAI*. 2020, pp. 530–536.

[20] J. Luiten et al. “HOTA: A Higher Order Metric for Evaluating Multi-object Tracking”. In: *Int. J. Comput. Vis.* 129.2 (2021), pp. 548–578.

[21] H. Luo et al. “Bag of Tricks and a Strong Baseline for Deep Person Re-Identification”. In: *CVPR Workshops*. 2019, pp. 1487–1495.

[22] A. Milan et al. “MOT16: A Benchmark for Multi-Object Tracking”. In: *CoRR* abs/arXiv:1603.00831 (2016).

[23] M. H. Nasseri et al. “Fast Online and Relational Tracking”. In: *CoRR* abs/arXiv:2208.03659 (2022).

[24] H. Rezatofighi et al. “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression”. In: *CVPR*. 2019, pp. 658–666.

[25] E. Ristani et al. “Performance Measures and a Data Set for Multi-target, Multi-camera Tracking”. In: *ECCV Workshops*. 2016, pp. 17–35.

[26] E. Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. In: *ICCV*. 2011, pp. 2564–2571.

[27] S. Shao et al. “CrowdHuman: A Benchmark for Detecting Human in a Crowd”. In: *CoRR* abs/arXiv:1805.00123 (2018).

[28] D. Stadler and J. Beyerer. “BYTEv2: Associating More Detection Boxes Under Occlusion for Improved Multi-Person Tracking”. In: *ICPR Workshops*. 2022.

[29] D. Stadler and J. Beyerer. “Improving Multiple Pedestrian Tracking by Track Management and Occlusion Handling”. In: *CVPR*. 2021, pp. 10958–10967.

[30] D. Stadler and J. Beyerer. “Modelling Ambiguous Assignments for Multi-Person Tracking in Crowds”. In: *WACV Workshops*. 2022, pp. 133–142.

[31] D. Stadler and J. Beyerer. “Multi-Pedestrian Tracking with Clusters”. In: *AVSS*. 2021.

[32] D. Stadler and J. Beyerer. “On the Performance of Crowd-Specific Detectors in Multi-Pedestrian Tracking”. In: *AVSS*. 2021.

[33] P. Sun et al. “TransTrack: Multiple Object Tracking with Transformer”. In: *CoRR* abs/arXiv:2012.15460 (2021).

[34] S. Tang et al. “Multiple People Tracking by Lifted Multicut and Person Re-identification”. In: *CVPR*. 2017, pp. 3701–3710.

[35] Q. Wang et al. “Multiple Object Tracking With Correlation Learning”. In: *CVPR*. 2021, pp. 3876–3886.

[36] S. Wang et al. “Extendable Multiple Nodes Recurrent Tracking Framework With RTU++”. In: *IEEE Trans. Image Process.* 31 (2022), pp. 5257–5271.

[37] S. Wang et al. “Tracking Game: Self-adaptative Agent based Multi-object Tracking”. In: *ACM Multimedia*. 2022, pp. 1964–1972.

[38] Z. Wang et al. “Towards Real-Time Multi-Object Tracking”. In: *ECCV*. 2020, pp. 107–122.

[39] C. K. I. Williams and C. E. Rasmussen. “Gaussian Processes for Regression”. In: *NIPS*. 1995, pp. 514–520.

[40] N. Wojke, A. Bewley, and D. Paulus. “Simple online and realtime tracking with a deep association metric”. In: *ICIP*. 2017, pp. 3645–3649.

[41] J. Xu et al. “Spatial-Temporal Relation Networks for Multi-Object Tracking”. In: *ICCV*. 2019, pp. 3987–3997.

[42] F. Yang et al. “ReMOT: A model-agnostic refinement for multiple object tracking”. In: *Image Vis. Comp.* 106 (2021).

[43] F. Zeng et al. “MOTR: End-to-End Multiple-Object Tracking with Transformer”. In: *ECCV*. 2022, pp. 659–675.

[44] H. Zhang et al. “ResNeSt: Split-Attention Networks”. In: *CVPR Workshops*. 2022, pp. 2735–2745.

[45] S. Zhang, R. Benenson, and B. Schiele. “CityPersons: A Diverse Dataset for Pedestrian Detection”. In: *CVPR*. 2017, pp. 4457–4465.

[46] Y. Zhang et al. “ByteTrack: Multi-Object Tracking by Associating Every Detection Box”. In: *ECCV*. 2022.

[47] Z. Zheng et al. “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression”. In: *AAAI*. 2020, pp. 12993–13000.

[48] X. Zhou, V. Koltun, and P. Krähenbühl. “Tracking Objects as Points”. In: *ECCV*. 2020, pp. 474–490.