



BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking

Vukasin D. Stanojevic¹ · Branimir T. Todorovic¹

Received: 21 December 2023 / Revised: 11 March 2024 / Accepted: 11 March 2024
© The Author(s) 2024

Abstract

Handling unreliable detections and avoiding identity switches are crucial for the success of multiple object tracking (MOT). Ideally, MOT algorithm should use true positive detections only, work in real-time and produce no identity switches. To approach the described ideal solution, we present the BoostTrack, a simple yet effective tracing-by-detection MOT method that utilizes several lightweight plug and play additions to improve MOT performance. We design a detection-tracklet confidence score and use it to scale the similarity measure and implicitly favour high detection confidence and high tracklet confidence pairs in one-stage association. To reduce the ambiguity arising from using intersection over union (IoU), we propose a novel Mahalanobis distance and shape similarity additions to boost the overall similarity measure. To utilize low-detection score bounding boxes in one-stage association, we propose to boost the confidence scores of two groups of detections: the detections we assume to correspond to the existing tracked object, and the detections we assume to correspond to a previously undetected object. The proposed additions are orthogonal to the existing approaches, and we combine them with interpolation and camera motion compensation to achieve results comparable to the standard benchmark solutions while retaining real-time execution speed. When combined with appearance similarity, our method outperforms all standard benchmark solutions on MOT17 and MOT20 datasets. It ranks first among online methods in HOTA metric in the MOT Challenge on MOT17 and MOT20 test sets. We make our code available at <https://github.com/vukasin-stanojevic/BoostTrack>.

Keywords Multi-object tracking · Data association · Similarity measure · Detection confidence · Tracking-by-detection

1 Introduction

Multiple object tracking (MOT) is one of the most important problem in computer vision and has applications in areas of autonomous robotics [20, 50], autonomous driving [12, 24, 43, 52] and smart cities [8, 44, 52, 72]. The problem consists of determining the position and identity of each object of interest (e.g. pedestrian) for every frame of the video. This is usually done in a tracking-by-detection paradigm, by applying detection and tracking steps for every input frame. Given a set of detections, the goal of the tracking step is to assign detections to tracked objects. Due to occlusions,

some objects cannot be detected even though they are still in the scene. When an object reappears, it should not be recognized as a new object but rather matched to an existing one. Kalman filter [27] is usually used as a tracking algorithm to overcome missing detections and occlusions and provide estimates of the object state. The assignment problem can be formulated as a bipartite matching task between the detections and tracklets and solved using the Hungarian algorithm [28]. Intersection over union (IoU) is an effective measure of similarity between the detections and existing tracklets and can be used to create a cost matrix needed for the Hungarian algorithm. To better deal with occlusions and crowded scenes, appearance similarity is usually used in addition to IoU or other motion cues. Computing appearance similarity requires the extraction of visual features. However, using high-quality feature extractors (e.g. FastReID [25]) increases execution time and limits real-time application [1].

To reduce the number of false positives and ghost tracks, low-confidence detections are usually filtered out, and only a subset of detections is used for association. However, not all

✉ Vukasin D. Stanojevic
vukasin.stanojevic@pmf.edu.rs
Branimir T. Todorovic
branimir.todorovic@pmf.edu.rs

¹ Department of Computer Science, Faculty of Science and Mathematics, University of Nis, Visegradska 33, Nis 18000, Serbia

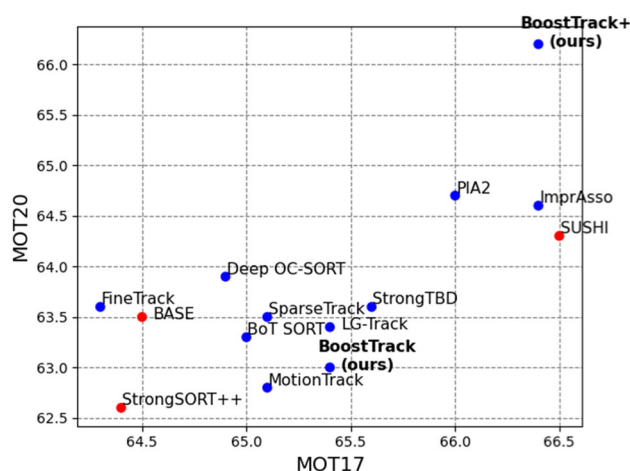


Fig. 1 Value of HOTA metric for various tracking methods on MOT17 and MOT20 test sets under private detection protocol. Blue circles represent online trackers

low-confidence detections are false positives. ByteTrack [69] uses a two-stage assignment where high-confidence detections are used in the first stage while remaining detections and unassociated tracklets are used in the second. Following a ByteTrack, several works have used low-confidence detections in the second assignment stage [1, 21, 30, 37, 58, 62]. On the other hand, using low-confidence detections in the second stage to match with remaining tracklets can result in identity switches (IDSWs). Suppose two objects bounding boxes are highly overlapped and only one has a high detection confidence score. In that case, we may assign that detection to the wrong object (forcing the incorrect association of the other detection in the second stage), but if we used both detections in the first stage, we could match them correctly [55].

Several papers [11, 13, 60] have adopted multiple-stage association where more recently updated tracklets (or the more *confident* ones) are associated first. Note that any multiple-stage association can introduce identity switches.

Ideally, MOT algorithm should be online, operating in real-time, the similarity measure should be discriminative enough to enable correct match of all tracklet-detection pairs in one-stage assignment, and all true positive and none of the false positive detections should be used.

To approach the described ideal solution, in this paper, we present the BoostTrack, a simple tracking-by-detection system built on top of SORT [6] that uses several lightweight plug-and-play additions that can significantly improve performance.

To avoid two-stage assignment and still utilize low-confidence detections, we propose to increase (boost) the confidence of two groups of low-confidence detection bounding boxes:

1. the bounding boxes where we predict an object *should* be,
2. counterintuitively, the bounding boxes where currently tracked objects *should not* be.

When an object is partially occluded, its detected bounding box confidence can be low, but the IoU between the predicted position and the bounding box can be high. We propose to increase the detection confidence of such detections. On the other hand, low-confidence detections positioned where we do not predict an object should be could be a noise, but can often be a new object that is only partially visible (e.g. entering the scene or standing on the edge). We use Mahalanobis distance measure [38] to discover these outliers and find that increasing the confidence of these detections also improves the performance.

To utilize the benefits of multiple-stage assignment and avoid its drawbacks, we introduce detection-tracklet confidence, which can be used to scale any similarity measure and implicitly favour high-confidence tracklet, high-confidence detection pairs in a one-stage assignment.

IoU alone can lead to many identity switches in crowded scenes, and recent algorithms use appearance features in addition to IoU and other motion features. However, using an additional visual embedding module increases time complexity, reducing FPS and the possibility of real-time application. We propose three lightweight plug and play additions that can improve association performance:

1. We use detection-tracklet confidence scores to scale IoU and increase the similarity of high confidence detection-tracklet pairs. High variance prediction giving high IoU (or any other similarity measure) with relatively low confidence detection should not have the same weight as the low variance prediction, high confidence detection overlap.
2. Mahalanobis distance [38] can be used as a similarity measure to account for estimated tracklet variance. Admissible values depend upon the dimensionality of the tracklet and the chosen confidence interval, and any change requires a different scaling parameter. We introduce a more robust way of using Mahalanobis distance as a similarity measure.
3. To reduce the possibility of identity switches in crowded scenes, we introduce shape similarity motivated by the fact that a mismatch can happen due to the high IoU overlap of moving objects. Still, the shape of the objects (i.e. width and height) should remain relatively constant in a short time frame.

In the rest of the paper, we refer to our methods for improving the estimation of the detection confidence and improving the

calculation of the similarity matrix as detection confidence boosting and similarity boosting, respectively.

We demonstrate the effectiveness of proposed additions on MOT17 [41] and MOT20 [14] datasets. It has become a standard practice to apply camera motion compensation (CMC) [1, 4, 16, 17, 37, 54] and interpolation of fragmented tracks [1, 17, 67, 69] to MOT. By integrating CMC and gradient boosting interpolation from [67], we achieve comparable results with state of the art methods, without using time costly visual features and running at the speed of 65.45 FPS on MOT17 and 32.79 FPS on MOT20, on a desktop with one NVIDIA GeForce RTX 3090 GPU and AMD Ryzen 9 5950X 16-Core CPU. Furthermore, by adding visual embedding to our system, which we refer to as BoostTrack+, at the expense of longer run-time (15.35 FPS on MOT17 and 3.05 FPS on MOT20), we outperform all standard benchmark solutions. BoostTrack+ ranks first among online methods in HOTA score on the MOT17 test set and first among all methods in HOTA score on the MOT20 test set (see figure 1 for visual comparison).

In summary, we make the following contributions:

- We introduce two confidence detection boosting techniques to utilize low confidence detections in one-stage assignment,
- We define detection-tracklet confidence and use it to give more weight to high detection confidence - high tracklet confidence association pairs, avoiding the need for multiple-stage association used in some previous works,
- We propose a novel way of incorporating Mahalanobis distance and shape similarity to the similarity matrix,
- We perform a detailed ablation study on MOT17 and MOT20 validation sets to show the effectiveness of the proposed methods. Our appearance-free BoostTrack method outperforms standard benchmark solutions and achieves comparable performance with the most recent methods on MOT17 and MOT20 test sets. Our BoostTrack+ method ranks first among online methods in HOTA score on both MOT17 and MOT20 test sets under private detection protocol.

We give overview of our method in figure 2. The rest of the paper is structured as follows: in section 2, we review the related work focusing on various multiple-stage association procedures, tracklet confidence, and different similarity measures used in previous works. Section 3 introduces detection-tracklet confidence and three proposed similarity matrix boosting techniques. In section 4, we discuss our two detection confidence boosting strategies - namely, increasing the detection confidence of *likely* objects based on IoU and increasing the detection confidence of *unlikely* objects based on Mahalanobis distance. We discuss experiments, show the results of the ablation study and compare our results with

benchmark methods in section 5. We conclude our work in section 6.

2 Related work

2.1 Sort

Solving the MOT online using the Kalman filter for tracking, IoU as a similarity matrix, and the Hungarian algorithm for the assignment was first introduced in SORT [6]. SORT uses a linear constant velocity model, and in every step, the Kalman filter is used to predict the state of the tracklet:

$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T, \quad (1)$$

where u, v, s, r represent the coordinates of the bounding box center, area and aspect ratio, respectively, and $\dot{u}, \dot{v}, \dot{s}$ corresponding velocities (authors assume aspect ratio to be constant).

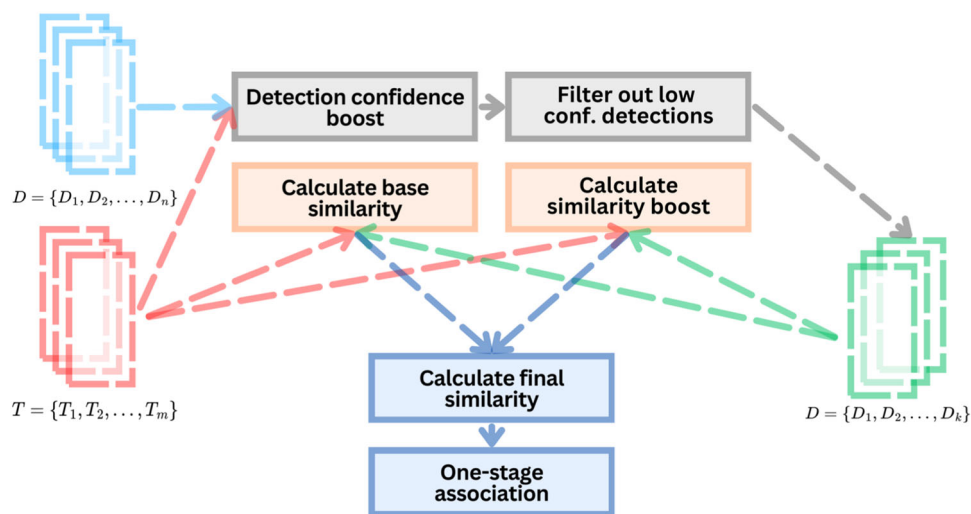
Assignment cost is calculated as $-1 \cdot \text{IoU}(D, T)$, and only assignments with IoU greater than a specified threshold, τ_{IoU} , are considered admissible.

2.2 Working with unreliable detections

Various strategies have been used to deal with unreliable detections, i.e. to identify and discard false positive detections [46]. In [61], an SVM [7] model was trained to classify detections into tracked or inactive class. In [42], all detections are used for the association, but the association is done in a two-stage manner, associating first detections and tracklets with greater similarity measure and associating the rest in the second stage. Filtering out unreliable detections based on detection confidence, i.e. thresholding, is the most common practice [17, 60, 65]. However, some low-confidence detections can correspond to partially occluded objects, and using these detections can increase performance.

In [51], high-confidence detections are used for trajectory initialization and tracking, while low-confidence detections are used for tracking only. ByteTrack [69] uses all detection boxes in a two-stage assignment where high-confidence boxes are used in the first stage and the remaining boxes and tracklets in the second. Following the ByteTrack, the same two-stage assignment was adopted in several works (e.g. [1, 30, 37, 58, 62]). In [39] authors proposed an offline tracking algorithm that uses all detection boxes. LG-Track [40] uses localization and classification confidence scores from the detectors and divides detections into four groups based on thresholds for the two scores. The association is performed in four stages using different cost matrices, which are differently scaled by detection confidence scores in different association stages. ImprAsso [55] splits detections into high

Fig. 2 Overview of our BoostTrack method. We use existing tracklets and detected bounding boxes to increase detection confidence scores before filtering out low confidence detections. We use the remaining bounding boxes to calculate the base (main) similarity measure and to improve it by adding our lightweight similarity boost



and low confidence sets and calculates associating distance for both sets, which are combined into a single matrix and used in a single association step.

2.3 Tracklet confidence

Being a general term, tracklet confidence has been differently defined and used for different purposes in several prior works [2, 3, 11, 13, 42, 63]. In [2], tracklet confidence is defined as an intuitive measure of similarity between constructed and real object trajectory and used to split trackers into low and high confidence groups, which are associated differently based on the group they belong to. In [11], the scene is split into $k \times k$ grid and both detections and predicted bounding boxes are used as candidates for the association. Tracklet confidence is defined and used to calculate the probability of the object being in a given area of the image and to filter out unreliable predictions. Tracklet score is calculated based on detection scores in [3] and used as tracklet termination criteria. In [42], tracklet confidence is designed and used to detect occlusions. Since low confidence detection, in the case of a true positive, usually means the object is partially visible or occluded, in [63], tracklet confidence is defined as a measure of object visibility and predicted as a part of object state. The difference between detection box confidence and predicted tracklet confidence is used as an additional similarity measure [63].

2.4 Similarity measures

Various improvements and additions to the IoU have been proposed to improve matching performance. In [30] Generalized IoU (GIoU) [48] is used. Normalized IoU is proposed in [42] to include differences between bounding box size and center. To account for object motion, a momentum term was

added to IoU in [9]. Width and height information are used in several previous works, e.g. [32, 35, 65], and in [63], Height Modulated IoU is introduced to incorporate height similarity into IoU matrix explicitly.

Since DeepSORT [60], using appearance features to associate detections with the tracklets has become a popular approach in tracking-by-detection MOT [1, 17, 57, 59]. Specifically, the cosine distance between visual embedding vectors is used to construct the association cost matrix. Mahalanobis distance is used as a gating mechanism to discard inadmissible associations. Since uncertainty increases when the tracklet is not updated (due to occlusion or missing detection), the assignment is done in cascade, in increasing order of the number of steps since the last update [60]. Following a DeepSORT, a weighted sum of Mahalanobis distance and cosine distance is used in several other works (e.g. [1, 17, 68]). In [33], Mahalanobis distance is smoothed by adding $\alpha \cdot I$ to the covariance matrix when calculating the distance.

2.5 Our approach

To the best of our knowledge, no prior work used detection-tracklet confidence to implicitly prioritize high-confidence detection or high-confidence tracklets in a single-stage association. In [35, 60], recently updated tracklets (i.e. the more confident ones) are explicitly favoured in cascade matching, and works [1, 30, 37, 42, 58, 62, 69] use two-stage matching prioritizing high-confidence detections.

We adopted and modified shape similarity from [32]. In [32] (and [65]), shape similarity (similarity between height and width) is used in conjunction with Euclidian distance and visual similarity to construct an association cost matrix and it cannot be used as a standalone metric or addition to the association cost. In our work, shape similarity is used together with detection-tracklet confidence to create a stan-

alone addition to the similarity matrix designed to reduce possible ambiguity arising from the IoU measure.

Prior works used Mahalanobis distance directly to define the similarity matrix [17, 68] or to discard inadmissible associations [60]. In our work, we convert Mahalanobis distance into probabilities, creating a more intuitive, robust, and universal metric.

In [63], Height Modulated IoU, velocity and confidence cost are added to the appearance cost to create a final cost matrix used for association. In our terminology, this can be seen as boosting the appearance similarity by adding additional lightweight measures. In our work, we propose and use different similarity measures.

In [55], one-stage association is performed using all detection boxes and normalization parameter β is used to control the association cost of low-confidence detections. Our detection confidence boosting strategy does not use all detection boxes in the association step or require any change in creating an assignment cost matrix but rather increases the confidence of some boxes before filtering out low-confidence detections. No prior work, to our knowledge, explicitly targeted and used detections that *seem to be* outliers. Our detection confidence boosting method attempts to use more true positive detection bounding boxes without the need for sequence-specific hyperparameter tuning used in some previous works (e.g. [1, 55, 69]).

3 Similarity matrix boost techniques

In this section, we introduce our similarity matrix boost techniques. Proposed improvements are orthogonal to existing approaches and can be added to any similarity matrix S_{base} (e.g. S_{base} can be calculated as IoU between detected bounding boxes and existing tracklets) to improve assignment performance.

Note, to use the similarity matrix as an assignment cost matrix needed for the Hungarian algorithm [28], we need to “reverse” the values, i.e. the greater the similarity between a given detection-tracklet pair, the lower the corresponding assignment cost. As in SORT [6], we obtain the assignment cost matrix by multiplying the similarity matrix with -1.

3.1 Detection-tracklet confidence similarity boost

To benefit from hierarchical assignments that favour high-confident detections [1, 30, 37, 58, 62, 69] or recently updated tracklets [35, 60] and avoid the drawbacks of such approaches, we design detection-tracklet confidence as a scaling factor to favour high-confidence detection, high-confidence tracklet pairs in one-stage assignment.

Let $D = \{D_1, D_2, \dots, D_n\}$ and $T = \{T_1, T_2, \dots, T_m\}$ be the set of detections and set of tracklets, respectively,

and $c_{d_1}, c_{d_2}, \dots, c_{d_n}$ and $c_{t_1}, c_{t_2}, \dots, c_{t_m}$ their corresponding confidence scores. T_1, T_2, \dots, T_m are obtained as outputs of Kalman prediction step.

Recently updated tracklets (i.e. active tracklets that were recently assigned detection and executed Kalman update step) should have more reliable state prediction and higher confidence. Due to initial noisy predictions, new tracklets should have less confidence. Let $age(T_j)$ and $last_update(T_j)$ be the number of steps since creation and the number of steps since the last update of tracklet T_j , respectively. We define tracklet confidence c_{t_j} as:

$$c_{t_j} = \begin{cases} \beta^{s_{init} - age(T_j)}, & \text{if } age(T_i) < s_{init} \\ \beta^{last_update(T_j) - 1}, & \text{otherwise,} \end{cases} \quad (2)$$

for $j \in \{1, 2, \dots, m\}$. $\beta \in (0, 1)$, is the tracklet confidence decay hyperparameter, and s_{init} is the number of steps we consider the tracklet as “new”, i.e. having initial unreliable predictions.

Note that detection confidence scores $c_{d_1}, c_{d_2}, \dots, c_{d_n}$ are available as the output of the detector.

We define detection-tracklet confidence of detection D_i and tracklet T_j , c_{d_i, t_j} , as a product $c_{d_i} \cdot c_{t_j}$. To encourage admissible associations only (e.g. $IoU(D_i, T_j) \geq \tau_{IoU}$) we set c_{d_i, t_j} to 0 for inadmissible associations. In summary, we define c_{d_i, t_j} as

$$c_{d_i, t_j} = \begin{cases} c_{d_i} \cdot c_{t_j}, & \text{if } (D_i, T_j) \text{ is admissible} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

We define confidence matrix C as $C = [c_{d_i, t_j}]_{n \times m}$.

Using the detection-tracklet confidence scores, we can boost similarity matrix S by adding confidence scaled IoU(D, T):

$$S_{boost} = S_{base} + \lambda_{IoU} \cdot C \odot IoU(D, T), \quad (4)$$

where by \odot we denote element-wise matrix product, and λ_{IoU} is a hyperparameter. Note that any similarity score can be used in place of the IoU.

3.2 Mahalanobis distance similarity boost

Mahalanobis distance [38] is used as a similarity measure in some previous works (e.g. [1, 17, 68]).

Note that the Kalman filter, here used as a state estimator, provides rigorous and optimal performance guarantees that do not rely on any assumptions on process or observation noise other than the mean and the covariance are known, and the Kalman filter is an optimal minimum mean square error estimator [27]. However, if the process and observation

noise are assumed to be Gaussian, then the estimated mean and state covariance parameterize Gaussian distributions.

In that case, Mahalanobis distance values are chi-squared distributed, and relevant values depend on the degrees of freedom and chosen confidence interval boundary. Usually, a 95% confidence interval and 4 degrees of freedom are used, giving a range of admissible values of (0, 9.4877) and requiring a relatively low weight factor $\lambda = 0.02$ [1, 17, 68]. In 3D MOT, detections are given as a tuple of 7 parameters, and in both 2D and 3D MOT we may choose to use only the box center for calculating Mahalanobis distance. However, changing the confidence interval or the degrees of freedom gives a different range of values and would require a different λ value.

On the other hand, only a relative difference between Mahalanobis distance values is relevant for the assignment task. From the perspective of a given tracklet, we can think of Mahalanobis distances between the detections as unnormalized probabilities. Motivated by this, we apply the softmax function to normalize Mahalanobis distance. First, we clip distances to a max limit value (e.g. 9.4877). Then, we subtract each value from the limit value. Finally, we apply softmax, and to avoid giving weight to inadmissible associations, we set the similarity measure (i.e. “probability”) to 0 for detections beyond the limit value. Note that we apply softmax for each column of the similarity matrix. A pseudocode of our procedure is illustrated in algorithm 1 in Appendix A.

After obtaining the S^{MhD} similarity matrix in a described way, we can boost the initial similarity measure S_{base} by:

$$S_{boost} = S_{base} + \lambda_{MhD} \cdot S^{MhD}(D, T), \quad (5)$$

where by λ_{MhD} we denote the weight of Mahalanobis distance similarity boost.

Normalizing Mahalanobis distances provides greater robustness to dimensionality changes (no need for changing the weight, but only the clip threshold) and enables direct comparison with other similarity measures. In case when few detection bounding boxes have a similar Mahalanobis distance to a given tracklet, softmax can effectively reduce the impact of using Mahalanobis distance similarity and make S_{base} similarity more decisive (however, in case of S_{base} ambiguity, S^{MhD} may provide new information and enable correct assignment). Furthermore, adjusting softmax temperature gives us more control in handling ambiguous assignments. Note that we keep the temperature parameter equal to 1 for simplicity.

3.3 Shape similarity boost

To avoid possible ambiguity from the other similarity metrics (e.g. IoU), we reintroduce the shape similarity metric. Consider a scenario where two objects are highly overlapped (e.g.

pedestrians passing by one another). Corresponding tracklets can have greater IoU with wrong detection boxes leading to identity switch. However, in a short time frame, the objects’ shape (width and height) should remain relatively constant, and using shape information could reduce possible ambiguity.

We should not rely too much on the tracklet shape information from a tracklet that was not recently updated. For example, a person may move hands and can appear wider, but only temporarily. Even the height can change due to the object moving closer or farther away from the camera. On the other hand, we should also consider detection confidence because comparing shapes with unreliable detections could reduce the reliability of the shape similarity. To account for this, we scale shape similarity by detection-tracklet confidence scores introduced in subsection 3.1. Let $ds_{i,j}$ be the shape difference between the detection D_i and the tracklet T_j defined as

$$ds_{i,j} = \frac{|D_i^w - T_j^w|}{\max(D_i^w, T_j^w)} + \frac{|D_i^h - T_j^h|}{\max(D_i^h, T_j^h)}, \quad (6)$$

where by w and h in superscript we denote width and height, respectively. We define our shape similarity measure between detection D_i and tracklet T_j as

$$S_{d_i,t_j}^{shape} = c_{d_i,t_j} \cdot \exp(-ds_{i,j}). \quad (7)$$

We can boost similarity measure S_{base} by:

$$S_{boost} = S_{base} + \lambda_{shape} \cdot S^{shape}(D, T), \quad (8)$$

where λ_{shape} is a hyperparameter used as the weight of the shape similarity.

Combining the three proposed similarity boost techniques we get:

$$S_{boost} = S_{base} + \lambda_{IoU} \cdot C \odot \text{IoU}(D, T) + \lambda_{MhD} \cdot S^{MhD}(D, T) + \lambda_{shape} \cdot S^{shape}(D, T). \quad (9)$$

4 Detection confidence boosting techniques

Not all low-confidence detections are false positives. In this section, we describe our proposed methods of utilizing two groups of low-confidence detections by boosting their confidence score.

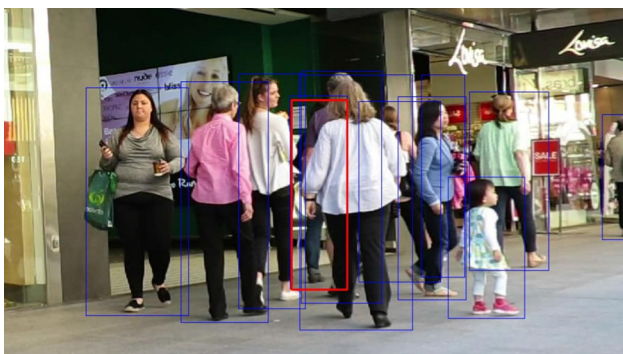


Fig. 3 Detecting likely objects based on IoU value. Blue bounding boxes represent tracklets. The red bounding box is the detection with the original confidence score of 0.1748, which is increased based on high IoU with the predicted bounding box

4.1 Detecting likely objects

When an object is partially occluded, sometimes it can still be detected. Detection confidence of such partially visible objects can be low, and their detected bounding boxes can be discarded for having a confidence score below some threshold τ_D . However, having a tracking module, i.e. Kalman filter, enables us to predict where the object should be. If the IoU score between the tracklet and the detection box is high, we propose to increase the confidence of that detection box, enabling it to be used for later association. For each detection box D_i we can calculate IoU between the detection box and all tracklets and increase the confidence of detection D_i , and obtain boosted confidence \hat{c}_{d_i} using the maximum value of calculated IoUs:

$$\hat{c}_{d_i} = \max(c_{d_i}, \beta_c \cdot \max_j(\text{IoU}(D_i, T_j))). \tag{10}$$

Hyperparameters β_c and τ_D implicitly define the IoU threshold for the detections to be used for the association, even if the original detection confidence c_{d_i} is low. This way, we also increase the confidence of some detections where $c_{d_i} > \tau_D$, which we found to slightly increase the performance combined with our detection-tracklet confidence from subsection 3.1.

Figure 3 shows an example of a highly occluded person with a low detection confidence bounding box (in red), $c_d = 0.1748$, that has a high IoU (0.949) with the predicted bounding box. Our method increases the confidence of this detection and uses it for association.

4.2 Detecting “unlikely” objects

Previous confidence boosting strategy aimed to increase the detection confidence for detections where a tracked object is

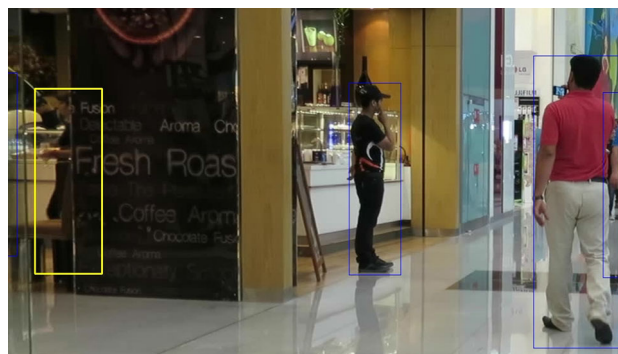


Fig. 4 Detecting “unlikely” objects based on Mahalanobis distance. Blue bounding boxes represent tracklets. The yellow bounding box is the detection with the original confidence score of 0.2252, which is increased based on the high Mahalanobis distance between all existing tracklets

likely to be. However, some objects could never be detected in the first place. They can be partially occluded during the entire video or positioned on the edge of the scene and only partially visible. We propose a method to detect some of these objects. In particular, we note that false positive low-confidence detections typically occur near the tracked objects (with the exception discussed in the previous subsection). If we detect an object *far* from where any tracked object is supposed to be, we can assume that this detection is not result of a motion and noise produced by currently tracked object. Such outliers can actually be previously undetected objects.

Mahalanobis distance can be used to detect outliers [23]. As previously noted, Mahalanobis distance values are chi-square distributed and values greater than a certain threshold τ_{MhD} are considered outliers. We set the threshold τ_{MhD} to 13.2767, corresponding to a 99% confidence interval bound for 4 degrees of freedom chi-square distribution.

For a given detection D_i , we compute the distance between D_i and every tracklet T_j and consider D_i outlier if the distance between D_i and the closest tracklet is greater than τ_{MhD} , i.e. if

$$\min_j(MhD(D_i, T_j)) > \tau_{MhD}. \tag{11}$$

If a given detection D_i is an outlier with respect to state distributions of all currently tracked objects, we consider it to correspond to the previously undetected object. However, some of these outliers can still be false positives. To all detections where inequality (11) holds, we apply non-maximum suppression with a threshold of $\tau_{NMS} = 0.3$ to remove overlapping detections and reduce the number of used false positive detections. We provide details on the influence of τ_{NMS} in Appendix B. We set the detection confidence of the remaining detections to τ_D , allowing them to be used in the association step.

Figure 4 shows an example of a low detection confidence bounding box (in yellow) with original detection confidence $c_d = 0.2252$. The closest Mahalanobis distance between existing tracklets is 2178.23, and we assume it is unlikely to be a false positive generated by currently tracked objects. We increase the detection confidence of such detections. Note that our method can increase the confidence of false positive detections. Still, in the ablation study in subsection 5.3, we show that the number of used IDs does not increase significantly and that our method improves MOT performance.

5 Experiments

5.1 Datasets and metrics

Datasets. We use standard MOT benchmark datasets, MOT17 [41] and MOT20 [14], to conduct experiments. As in several popular benchmarks [1, 17, 37, 69], we replace original dataset detections with detections from YOLOX-X [22] and conduct experiments under private detection protocol.

MOT17 contains static and moving camera videos of pedestrians. Videos are filmed with different FPS settings (ranging from 14 to 30 FPS) and split into training and test sets. Following previous works [1, 17, 37, 69, 71], we construct a validation set using the second half of each training sequence and use it for the ablation study (note that using a validation set is important because the detector and feature extractor are trained on the first half of the training data).

MOT20 contains 8 sequences (4 for training and the remaining 4 as the test set) of crowded scenes filmed with a static camera. Similarly, we use the second half of each sequence as the validation set.

Metrics. We evaluate tracking performance using widely accepted metrics CLEAR metrics (we focus primarily on MOTA, IDs, IDSWs) [5], IDF1 [49] and HOTA [36]. MOTA (Multi-Object Tracking Accuracy) is defined using the number of false positives (FP), false negatives (FN), identity switches (IDSW) and the total number of ground truth detections (gtDet) as:

$$\text{MOTA} = 1 - \frac{|\text{FP}| + |\text{FN}| + |\text{IDSW}|}{|\text{gtDet}|}. \quad (12)$$

Note that FP and FN are ID agnostic. As such, MOTA does not penalize wrong associations heavily (only the IDSW accounts for associations mismatch, but is insignificant compared to $|\text{FP}| + |\text{FN}|$) and is mainly used as a metric for detection performance. IDF1 metric computes the matching on the id level and can be used to measure the association performance. HOTA combines detection, association and localization accuracy and attempts to assess the whole tracking performance. It is calculated as a geo-

metric mean between the detection accuracy (DetA) and the association accuracy (AssA), integrated over different localisation thresholds (approximated as a finite sum for $\alpha \in \{0.05, 0.1, \dots, 0.95\}$) [36].

Since our detection confidence boosting techniques can introduce possible false positive detections and new identities, we explicitly monitor IDs and IDSWs in addition to MOTA, IDF1 and HOTA when discussing the impact of our confidence boosting methods.

5.2 Implementation details

Kalman filter. As in [17, 60, 68, 69], we define the state as eight-dimensional vector $\mathbf{x} = [u, v, h, r, \dot{u}, \dot{v}, \dot{h}, \dot{r}]^T$, where by (u, v) we denote coordinates of the bounding box center, height and aspect ratio of the bounding box, respectively, while $\dot{u}, \dot{v}, \dot{h}, \dot{r}$ represent their corresponding velocities. We retain constant process and measurement noise from [6].¹

MOT specific settings. Same as [6], we report tracklet state only in the case of 3 consecutive matches (i.e. the Kalman updates) and use $\tau_{IoU} = 0.3$ as criteria to discard inadmissible associations. As in [9, 37], we set the detection confidence threshold τ_D to 0.6 for MOT17 and 0.4 for MOT20. In previous works (e.g. [1, 60, 69]), unassociated tracklets are kept for $A_{\max} = 30$ frames. Since our detection confidence boost techniques rely on the tracklet predicted position, we keep tracklets alive for a longer period. Since different sequences can have different frame rates, a fixed value of 30 steps corresponds to different clock times (e.g. 2.14 seconds for the MOT17-05 sequence and 1 second for the MOT17-09 sequence). We use sequence specific value $A_{\max} = \max(30, 2 * \text{FPS})$, i.e. we keep unassociated tracklets alive for at least 2 seconds. As in previous works (e.g. [19, 31, 37, 70]), we resize images from MOT17 and MOT20 to 1440×800 and 1600×896 , respectively.

BoostTrack specific settings. As the base similarity measure, we use IoU, i.e. $S_{base} = \text{IoU}$ in equation (9).

We run a grid search to find values of $\lambda_{IoU}, \lambda_{MhD}$ and λ_{shape} . For each λ we tested values from the set $\{0, 0.25, 0.5, 0.75, 1\}$. As the best trade-off between different metrics on MOT17 and MOT20, we choose values $\lambda_{IoU} = 0.5, \lambda_{MhD} = 0.25, \lambda_{shape} = 0.25$. We observed that any setting where λ_{MhD} is not dominant improves the performance compared to the baseline.

We tested various (β, s_{init}) settings for $(\beta, s_{init}) \in [0.7, 0.95] \times \{0, 1, \dots, 10\}$ on the MOT17 validation set. We found that any setting improves the performance of the baseline and set $\beta = 0.9, s_{init} = 7$ for the best trade-off between

¹ They used a 7-tuple as a state, and we adapted it to the 8-tuple scenario. We lower the variance of aspect ratio measurement noise to 0.01 as in [60] and increase the processing noise variance corresponding to \dot{r} to 0.01.

different metrics. We provide more details on the influence of β and s_{init} in Appendix C.

We set the limit value for Mahalanobis distance for similarity boost and outliers detection to 13.2767, corresponding to a 99% confidence interval boundary for 4 degrees of freedom chi-square distribution. For boosting the detection confidence of likely objects, we set $\beta_c = 0.65$ for MOT17 and $\beta_c = 0.5$ for the MOT20 dataset, corresponding to the required IoU of 0.923 and 0.8 for low-confidence detection to surpass the threshold τ_D .

We evaluate results using TrackEval [26].

Additional modules. For the YOLOX-X [22], we use the weights from [69]. We use Enhanced correlation coefficient maximization from [18] for CMC and rely on implementation from [17]. We keep the same settings as in [17], but resize the images to 350 pixels in width (and proportional height). We use linear interpolation implementation from [69] and do not set a maximum interval for interpolation. Tracks with less than 25 detections are not interpolated (default value from [69]), and we further improve interpolation results by applying gradient boosting interpolation from [67]. As in [1, 37, 69], we use FastReID [25] as a visual embedding model, and apply Dynamic Appearance embedding update from [37]. When using appearance similarity, we add $\lambda_{app} \cdot S_{app}(D_i, T_j)$ to the overall similarity measure proposed in equation (9), where $S_{app}(D_i, T_j)$ represents cosine similarity between visual embedding vectors. To account for a total weight of 2 for non-appearance similarity, we set $\lambda_{app} = 3$ to give more weight to the appearance similarity. In addition to admissibility condition $\text{IoU}(D_i, T_j) \geq \tau_{IoU}$, we allow assignments between detection box D_i and tracklet T_j if

$$\text{IoU}(D_i, T_j) \geq \frac{\tau_{IoU}}{2} \wedge S_{app}(D_i, T_j) \geq \frac{3}{4}. \quad (13)$$

Hardware. We run all the experiments on the desktop with AMD Ryzen 9 5950X 16-Core CPU and NVIDIA GeForce RTX 3090 GPU.

Software. Our implementation is developed on top of publicly available codes [6, 17, 37, 67, 69].

5.3 Ablation study

Similarity boost To test the impact of each component of the proposed similarity boost technique, we conduct a detailed ablation study on MOT17 and MOT20 validation sets. Table 1 shows results for every combination of proposed components: detection-tracklet confidence (DTC) boost, Mahalanobis distance (MhD) boost and shape similarity boost. The first row corresponds to the case where $S = \text{IoU}(D, T)$ and represents the baseline. We set $\lambda_{IoU} = 0.5$, $\lambda_{MhD} = 0.25$, $\lambda_{shape} = 0.25$ (see BoostTrack specific

settings from subsection 5.2). DTC and Shape similarity boost improve performance both as a standalone addition and combined. Adding the MhD similarity boost alone slightly decreases the overall performance, but combined with DTC and Shape, it results in significant performance gain on MOT17 and MOT20. We set λ values based on the grid search and found that any setting where λ_{MhD} is not dominant improves the performance. Since similarity boost should improve the association performance, we consider IDF1 the most important metric to show the advantage of the proposed methods. We achieved +1.346 IDF1 on MOT17 and +0.953 IDF1 on MOT20 compared to the baseline. Note that we also achieve improvement in other metrics.

Detection confidence boost. We test the effectiveness of our detection confidence boost (DCB) techniques and show results in table 2. We tested our detecting likely objects (DLO) and detecting “unlikely” objects (DUO) strategies combined with our similarity boost (SB) components (when SB is used, we assume all three components, i.e. the last row of table 1). In addition to previous metrics, we display the total number of used IDs because boosting detection confidence can introduce new identities. Note that MOT17 and MOT20 validation sets contain 339 and 1418 ground truth identities, respectively.

Our study shows that both DCB techniques improve the MOT performance, both as standalone additions and combined with our SB techniques. Since DCB introduces new detections and SB aims to improve association, we discuss improvements in HOTA [36] metric to summarize both detection and association performance. Without SB, we get +0.584 HOTA on MOT17 and +1.925 HOTA on MOT20. If we include SB, we get an overall performance increase of +1.546 HOTA on MOT17 and +2.327 HOTA on MOT20. This shows that proposed boosting techniques complement each other and can be used jointly. We also get a significant increase in IDF1 and MOTA metrics. IDSW cannot be trivially compared because the number of used IDs can be significantly different.

Additional modules. As the standard practice, BoostTrack uses camera motion compensation (CMC) and interpolation to connect fragmented tracks. We use gradient boosting interpolation (GBI) from [67], but we also show results obtained using linear interpolation (LI). Finally, we add appearance similarity (AS) and show the effect of added components on MOT17 and MOT20 datasets in table 3. We did not include run-time speed in tables 1 and 2 because all the experiments run at approximately the same FPS. Adding CMC or AS affects execution speed, and we show FPS in addition to previously used metrics. To make comparison easier, we divide the table 3 into three parts. The first part shows the results of adding CMC, GBI and AS to the baseline. To distinguish between the fast appearance-free method and the slower method that uses AS, we label the latter as Boost-

Table 1 Ablation study on the MOT17 and MOT20 validation sets for different similarity boost settings (best in bold)

Setting DTC				MOT17				MOT20			
	MhD	Shape		HOTA	MOTA	IDF1	IDSW	HOTA	MOTA	IDF1	IDSW
DTC	MhD	Shape		HOTA	MOTA	IDF1	IDSW	HOTA	MOTA	IDF1	IDSW
✗	✗	✗		66.132	74.736	77.298	227	56.717	69.913	73.715	1120
✗	✗	✓		66.768	75.003	78.575	182	57.202	69.953	74.471	1074
✗	✓	✗		65.953	74.491	77.289	250	56.041	69.639	72.766	1488
✗	✓	✓		66.345	74.839	77.901	209	57.015	69.831	74.175	1248
✓	✗	✗		66.27	74.89	78.111	188	57.076	69.959	74.157	1095
✓	✗	✓		66.526	74.988	78.553	181	57.116	69.964	74.227	1101
✓	✓	✗		66.419	74.916	78.022	199	57.31	69.928	74.612	1134
✓	✓	✓		66.831	74.951	78.644	175	57.414	69.962	74.668	1114

Table 2 Ablation study on the MOT17 and MOT20 validation sets for different detection confidence boost settings (best in bold)

Setting SB			MOT17					MOT20				
	DLO	DUO	HOTA	MOTA	IDF1	IDs	IDSW	HOTA	MOTA	IDF1	IDs	IDSW
✗	✗	✗	66.132	74.736	77.298	440	227	56.717	69.913	73.715	1894	1120
✗	✗	✓	66.343	74.817	77.575	443	223	56.723	69.915	73.731	1902	1125
✗	✓	✗	66.361	74.897	77.367	445	222	58.628	73.299	75.211	2039	1243
✗	✓	✓	66.716	74.938	77.716	457	219	58.642	73.271	75.225	2063	1246
✓	✗	✗	66.831	74.951	78.644	424	175	57.414	69.962	74.668	1868	1114
✓	✗	✓	67.117	75.029	79.104	428	171	57.429	69.966	74.698	1870	1116
✓	✓	✗	67.134	75.125	78.987	428	170	59.054	73.339	75.969	2024	1230
✓	✓	✓	67.678	75.014	79.854	442	167	59.044	73.319	75.921	2041	1230

Track+ and show the results of BoostTrack and BoostTrack+ in the second and third parts of the table, respectively.

Adding gradient boosting interpolation greatly improves results on MOT17, and we achieve 70.647 HOTA (+2.969), 79.8 MOTA (+4.786) and 82.323 IDF1 (+2.469). By using CMC combined with GBI, we achieve 71.63 HOTA, 80.692 MOTA and 83.959 IDF1 on the MOT17 validation set, retaining real-time execution speed of 65.45 FPS. Adding appearance similarity further improves the performance, and we get +0.781 HOTA and +1.426 IDF1 with a slight decrease in MOTA (−0.022).

On MOT20, GBI results in +2.553 HOTA, +4.155 MOTA and +1.476 IDF1 improvement. Since videos in MOT20 are filmed with a static camera, using CMC has no significant impact on performance. Adding appearance similarity further improves the performance: +0.834 HOTA, +0.289 MOTA, +1.504 IDF1, at the expense of increased computation time (3.05 FPS).

In the case of the MOT17, our proposed additions effectively replace the need for AS, while adding AS increases the performance further. AS has a more significant impact on association performance in crowded scenes from MOT20, and adding our techniques slightly reduces HOTA and IDF1 values (−0.008 HOTA and −0.349 IDF1) but increases MOTA value (+0.504 MOTA). For a fair comparison, we

keep the same ratio of non-AS and AS when adding AS to the baseline (no SB+DCB) and set $\tau_{AS} = 1.5$.

5.4 Comparison with benchmark methods

We show the evaluation results on the MOT17 and the MOT20 test sets under private detection protocol in tables 4 and 5, respectively.

On the MOT17 and the MOT20 test sets our fast non-appearance BoostTrack method shows comparable performance. On the MOT17, BoostTrack ranks fourth among online methods in HOTA metric, while on the MOT20 it shows comparable results (note that it still outperforms standard benchmark solutions such as StorgSORT [17] or ByteTrack [69]).

Our BoostTrack+ method effectively outperforms standard benchmark solutions on both datasets. Among online trackers, BoostTrack+ ranks first in HOTA and second in IDF1 metric on the MOT17 test set. BoostTrack+ ranks first in HOTA metric among all methods and first in IDF1 metric among online methods. Our method achieves comparable results in MOTA metric on both datasets.

Table 3 Ablation study on the MOT17 and MOT20 validation sets for different additional modules

Setting SB+DCB	CMC	LI	GBI	AS	MOT17					MOT20				
					HOTA	MOTA	IDF1	IDSW	FPS	HOTA	MOTA	IDF1	IDSW	FPS
X	✓	X	X	X	67.881	75.266	80.628	133	67.09	57.053	69.936	74.199	1093	40.40
X	✓	X	✓	X	70.788	80.182	83.299	102	67.09	61.229	77.018	77.183	804	40.40
X	✓	X	X	✓	68.75	75.296	82.165	111	15.68	58.225	70.008	76.225	961	3.28
X	✓	X	✓	✓	71.79	80.258	84.802	83	15.68	62.688	77.227	79.437	718	3.28
✓	X	X	X	X	67.678	75.014	79.854	167	340.00	59.044	73.319	75.921	1230	49.79
✓	X	✓	X	X	70.348	79.803	82.303	154	340.00	61.016	77.208	77.247	923	49.79
✓	X	X	✓	X	70.647	79.8	82.323	148	340.00	61.597	77.474	77.397	911	49.79
✓	✓	X	X	X	68.526	75.454	81.396	116	65.45	59.231	73.363	76.108	1225	32.79
✓	✓	✓	X	X	71.364	80.635	83.927	97	65.45	61.261	77.184	77.437	891	32.79
✓	✓	X	✓	X	71.63	80.692	83.959	97	65.45	61.846	77.442	77.584	882	32.79
✓	✓	X	X	✓	69.352	75.624	82.971	112	15.35	59.91	73.427	77.357	1110	3.05
✓	✓	✓	X	✓	72.15	80.566	85.342	87	15.35	62.075	77.46	78.934	805	3.05
✓	✓	X	✓	✓	72.411	80.67	85.385	81	15.35	62.68	77.731	79.088	781	3.05

Table 4 Comparison with other MOT methods on the MOT17 test set (best in bold). We mark offline methods with '*'

Method	HOTA	MOTA	IDF1	IDSW
FairMOT [68]	59.3	73.7	72.3	3303
MOTR [66]	62.0	78.6	75.0	2619
ByteTrack [69]	63.1	80.3	77.3	2196
QuoVadis [15]	63.1	80.3	77.7	2103
FineTrack [47]	64.3	80.0	79.5	1272
StrongSORT++* [17]	64.4	79.6	79.5	1194
BASE* [29]	64.5	81.9	78.6	1281
Deep OC-SORT [37]	64.9	79.4	80.6	1023
BoT_SORT [1]	65.0	80.5	80.2	1212
SparseTrack [34]	65.1	81.0	80.1	1170
MotionTrack [45]	65.1	81.1	80.1	1140
LG-Track [40]	65.4	81.4	80.4	1125
StrongTBD [53]	65.6	81.6	80.8	954
PIA2 [56]	66.0	82.2	81.1	1026
ImprAsso [55]	66.4	82.2	82.1	924
SUSHI* [10]	66.5	81.1	83.1	1149
BoostTrack (ours)	65.4	80.5	80.2	1104
BoostTrack+ (ours)	66.4	80.6	81.8	1086

Table 5 Comparison with other MOT methods on the MOT20 test set (best in bold). We mark offline methods with '*'

Method	HOTA	MOTA	IDF1	IDSW
FairMOT [68]	54.6	61.8	67.3	5243
ByteTrack [69]	61.3	77.8	75.2	1223
QuoVadis [15]	61.5	77.8	75.7	1187
SuppTrack* [70]	61.9	78.2	75.5	1325
UTM [64]	62.5	78.2	76.9	1228
StrongSORT++* [17]	62.6	73.8	77.0	770
MotionTrack [45]	62.8	78.0	76.5	1165
BoT_SORT [1]	63.3	77.8	77.5	1313
LG-Track [40]	63.4	77.8	77.4	1161
SparseTrack [34]	63.5	78.1	77.6	1120
BASE* [29]	63.5	78.2	77.6	984
FineTrack [47]	63.6	77.9	79.0	980
StrongTBD [53]	63.6	78.0	77.0	1101
Deep OC-SORT [37]	63.9	75.6	79.2	779
SUSHI* [10]	64.3	74.3	79.8	706
ImprAsso [55]	64.6	78.6	78.8	992
PIA2 [56]	64.7	78.5	79.0	1023
BoostTrack (ours)	63	76.4	76.5	992
BoostTrack+ (ours)	66.2	77.2	81.5	899

6 Conclusions

In this paper, we presented three techniques for improving the similarity measure between detections and tracklets and two for increasing the confidence score of low-score detection bounding boxes. Our method uses simple one-stage association and, combined with camera motion compensation and gradient boosting interpolation, achieves comparable performance with state-of-the-art methods on MOT17 and MOT20

datasets while operating in real-time. Adding appearance similarity further increases the performance of our method, and our BoostTrack+ ranks the best online method in HOTA score for MOT17 and MOT20 datasets.

Author Contributions Vukasin Stanojevic: theoretical development, experiment implementation, paper writing, approving the final version of the article publication. Branimir Todorovic: guidance, theoretical development, approving the final version of the article for publication.

Funding No funding was received to assist with the preparation of this manuscript

Data availability We used MOT17 and MOT20 datasets publicly available at <https://motchallenge.net/>

Code Availability We make our code available at <https://github.com/vukasin-stanojevic/BoostTrack>

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A Calculating Mahalanobis distance similarity

In subsection 3.2, we described our method of calculating Mahalanobis distance similarity. The pseudocode of the described procedure is given in algorithm 1.

Algorithm 1 Calculating Mahalanobis distance similarity matrix.

```

1: procedure MHDIST( $D, T, d_{max}$ )  $\triangleright D = \{D_1, D_2, \dots, D_n\}$ ,
    $T = \{T_1, T_2, \dots, T_m\}$ 
2:    $MhD_{raw} = \text{compute\_mahalanobis\_dist}(D, T)$ 
3:    $mask = \mathbf{0}_{n \times m}$ 
4:   for each (i, j) do
5:     if  $MhD_{raw}[i, j] > d_{max}$  then
6:        $mask[i, j] = 1$ 
7:     end if
8:   end for
9:    $MhD_{raw}[mask] = d_{max}$ 
10:   $MhD_{raw} = d_{max} - MhD_{raw}$   $\triangleright$  Resulting values are in
   range (0,  $d_{max}$ ).
11:   $MhD = \mathbf{0}_{n \times m}$ 
12:  for j=1 to m do
13:     $MhD[:, j] = \text{softmax}(MhD_{raw}[:, j])$ 
14:  end for
15:   $MhD[mask] = 0$   $\triangleright$  For pairs ( $D_i, T_j$ ) with original
    $MhD_{raw}[i, j] > d_{max}$ .
16:  return MhD
17: end procedure

```

Appendix B NMS threshold

As noted in subsection 4.2, we apply non-maximum suppression to all detection boxes where equation (11) holds. We tested various τ_{NMS} settings on the MOT17 validation set and show the results in table 6. The first row of the table represents a baseline while the rest show results of applying our technique for detecting “unlikely” objects for specified τ_{NMS} value. We observe that $\tau_{NMS} = 0.3$ gives the best trade-off between different metrics and used IDs and use $\tau_{NMS} = 0.3$ in our experiments in section 5. Note that any τ_{NMS} setting improves the performance compared to the baseline in terms of MOTA metric. This indicates that more true positive detection boxes are being used when we apply our technique for boosting the detection confidence of “unlikely” objects.

Table 6 Effect of τ_{NMS} on the performance on MOT17 validation set

β	HOTA	MOTA	IDF1	IDs	IDSW
/	66.132	74.736	77.298	440	227
0.0	66.03	74.797	77.004	441	226
0.1	66.03	74.795	77.003	441	226
0.2	66.335	74.791	77.509	443	225
0.3	66.343	74.817	77.575	443	223
0.4	66.375	74.814	77.581	448	225
0.5	66.377	74.825	77.58	448	225
0.6	66.337	74.806	77.491	453	229
0.7	66.064	74.778	77.099	449	232
0.8	66.064	74.778	77.099	449	232
0.9	66.064	74.778	77.099	449	232
1.0	66.064	74.778	77.099	449	232

Appendix C Influence of detection-tracklet confidence hyperparameters

We choose tracklet confidence decay β and s_{init} values based on the results on the MOT17 validation set. We tested various settings for $\beta \in [0.7, 0.95]$ and $s_{init} \in \{0, 1, \dots, 10\}$. Table 7 shows values of HOTA, MOTA and IDF1 metrics for different tracklet confidence decay β and $s_{init} \in \{0, 7\}$. We show results without detection confidence boosting, camera motion compensation and interpolation for $\lambda_{IoU} = 0.5$, $\lambda_{MhD} = 0$ and $\lambda_{shape} = 0.25$. We set $\lambda_{MhD} = 0$ because Mahalanobis distance boost is not affected by β and s_{init} . The first line of the table corresponds to the baseline.

Table 7 Effect of β and s_{init} on the performance on MOT17 validation set

β	$s_{init} = 0$			$s_{init} = 7$		
	HOTA	MOTA	IDF1	HOTA	MOTA	IDF1
0	65.958	74.75	77.048	65.958	74.75	77.048
0.7	66.298	75.019	78.396	66.375	75.062	78.574
0.75	66.413	75.031	78.471	66.385	75.064	78.529
0.8	66.467	75.025	78.477	66.442	75.047	78.557
0.85	66.441	75.031	78.288	66.52	75.062	78.5
0.9	66.38	74.96	78.161	66.528	74.99	78.558
0.95	66.483	74.901	78.239	66.606	74.923	78.562

References

- Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-sort: robust associations multi-pedestrian tracking. arXiv preprint abs/2206.14651 (2022). <https://doi.org/10.48550/arXiv.2206.14651>
- Bae, S.H., Yoon, K.J.: Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1218–1225 (2014). <https://doi.org/10.1109/CVPR.2014.159>
- Benbarka, N., Schröder, J., Zell, A.: Score refinement for confidence-based 3d multi-object tracking. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8083–8090 (2021). <https://doi.org/10.1109/IROS51168.2021.9636032>
- Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 941–951 (2019). <https://doi.org/10.1109/ICCV.2019.00103>
- Bernardin, K., Stiefelghagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J. Image Video Process. **2008**, 1–10 (2008)
- Bewley, A., Ge, Z., Ott, L., et al.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP) pp. 3464–3468 (2016). <https://doi.org/10.1109/ICIP.2016.7533003>
- Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152 (1992). <https://doi.org/10.1145/130385.130401>
- Bumanis, N., Vitols, G., Arhipova, I., et al.: Multi-object tracking for urban and multilane traffic: building blocks for real-world application. In: ICEIS (1), pp. 729–736 (2021). <https://doi.org/10.5220/0010467807290736>
- Cao, J., Pang, J., Weng, X., et al.: Observation-centric sort: rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9686–9696 (2023). <https://doi.org/10.1109/CVPR52729.2023.00934>
- Cetintas, O., Brasó, G., Leal-Taixé, L.: Unifying short and long-term tracking with graph hierarchies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22877–22887 (2023). <https://doi.org/10.1109/CVPR52729.2023.02191>
- Chen, L., Ai, H., Zhuang, Z., et al.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multi-media and Expo (ICME) pp. 1–6 (2018). <https://doi.org/10.48550/arXiv.1809.04427>
- Kuang Chiu, H., Prioletti, A., Li, J., et al.: Probabilistic 3d multi-object tracking for autonomous driving. arXiv preprint (2020). <https://doi.org/10.48550/arXiv.2001.05673>
- Dao, M.Q., Frémont, V.: A two-stage data association approach for 3d multi-object tracking. Sensors **21**(9), 2894 (2021). <https://doi.org/10.3390/s21092894>
- Dendorfer, P., Rezatofighi, H., Milan, A., et al.: Mot20: A benchmark for multi object tracking in crowded scenes. (2020). <https://doi.org/10.48550/arXiv.2003.09003>
- Dendorfer, P., Yugay, V., Osep, A., et al.: Quo vadis: is trajectory forecasting the key towards long-term multi-object tracking? Adv. Neural Inf. Process. Syst. **35**, 15657–15671 (2022). <https://doi.org/10.48550/arXiv.2210.07681>
- Du, Y., Wan, J., Zhao, Y., et al.: Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021. In: Proceedings of the IEEE/CVF International conference on computer vision, pp. 2809–2819 (2021). <https://doi.org/10.1109/ICCVW54120.2021.00315>
- Du, Y., Zhao, Z., Song, Y., et al.: Strongsort: make deepsort great again. IEEE Trans. Multimedia (2023). <https://doi.org/10.1109/TMM.2023.3240881>
- Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE Trans. Pattern Anal. Mach. Intell. **30**(10), 1858–1865 (2008). <https://doi.org/10.1109/TPAMI.2008.113>
- Fischer, T., Huang, T.E., Pang, J., et al.: Qdtrack: quasi-dense similarity learning for appearance-only multiple object tracking. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 15380–15393 (2023). <https://doi.org/10.1109/TPAMI.2023.3301975>
- Gad, A., Basmaji, T., Yaghi, M., et al.: Multiple object tracking in robotic applications: trends and challenges. Appl. Sci. (2022). <https://doi.org/10.3390/app12199408>
- Gao, J., Wang, Y., Yap, K.H., et al.: Occlutrack: rethinking awareness of occlusion for enhancing multiple pedestrian tracking. arXiv preprint (2023). <https://doi.org/10.48550/arXiv.2309.10360>
- Ge, Z., Liu, S., Wang, F., et al.: Yolox: Exceeding yolo series in 2021. arXiv preprint (2021). DOIurl-<https://doi.org/10.48550/arXiv.2107.08430>
- Ghorbani, H.: Mahalanobis distance and its application for detecting multivariate outliers. Facta Universitatis, Series: Mathematics and Informatics, pp. 583–595. (2019). <https://doi.org/10.22190/FUMI1903583G>
- Guo, S., Wang, S., Yang, Z., et al.: A review of deep learning-based visual multi-object tracking algorithms for autonomous driving. Appl. Sci. (2022). <https://doi.org/10.3390/app122110741>
- He, L., Liao, X., Liu, W., et al.: Fastreid: a pytorch toolbox for general instance re-identification. arXiv preprint (2020). <https://doi.org/10.48550/arXiv.2006.02631>
- Jonathon Luiten, A.H.: Trackeval. <https://github.com/JonathonLuiten/TrackEval> (2020)
- Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960). <https://doi.org/10.1115/1.3662552>
- Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955). <https://doi.org/10.1002/nav.3800020109>
- Larsen, M.V., Rolfsjord, S., Gusland, D., et al.: Base: probably a better approach to multi-object tracking. arXiv preprint (2023). <https://doi.org/10.48550/arXiv.2309.12035>
- Li, J., Ding, Y., Wei, H.L.: Simpletrack: rethinking and improving the jde approach for multi-object tracking. Sensors (2022). <https://doi.org/10.3390/s22155863>

31. Liu, K., Jin, S., Fu, Z., et al.: Uncertainty-aware unsupervised multi-object tracking. arXiv preprint (2023). <https://doi.org/10.48550/arXiv.2307.15409>
32. Liu, M., Jin, C.B., Yang, B., et al.: Online multiple object tracking using confidence score-based appearance model learning and hierarchical data association. *IET Comput Vis* **13**(3), 312–318 (2019). <https://doi.org/10.1049/iet-cvi.2018.5499>
33. Liu, Z., Zhang, W., Gao, X., et al.: Robust movement-specific vehicle counting at crowded intersections. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2617–2625 (2020). <https://doi.org/10.1109/CVPRW50498.2020.00315>
34. Liu, Z., Wang, X., Wang, C., et al.: Sparsetrack: multi-object tracking by performing scene decomposition based on pseudo-depth. arXiv preprint (2023b). <https://doi.org/10.48550/arXiv.2306.05238>
35. Lu, J., Xia, M., Gao, X., et al.: Robust and online vehicle counting at crowded intersections. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 3997–4003 (2021). <https://doi.org/10.1109/CVPRW53098.2021.00451>
36. Luiten, J., Osep, A., Dendorfer, P., et al.: Hota: a higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.* **129**, 548–578 (2021). <https://doi.org/10.1007/s11263-020-01375-2>
37. Maggolino, G., Ahmad, A., Cao, J., et al.: Deep oc-sort: multi-pedestrian tracking by adaptive re-identification. arXiv preprint (2023). <https://doi.org/10.48550/arXiv.2302.11813>
38. Mahalanobis, P.C.: On the generalized distance in statistics. *Proc. Natl. Inst. Sci.* **2**, 49–55 (1936)
39. Mandel, T., Jimenez, M., Risley, E., et al.: Detection confidence driven multi-object tracking to recover reliable tracks from unreliable detections. *Pattern Recognit.* **135**, 109107 (2023). <https://doi.org/10.1016/j.patcog.2022.109107>
40. Meng, T., Fu, C., Huang, M., et al.: Localization-guided track: a deep association multi-object tracking framework based on localization confidence of detections. arXiv preprint (2023). <https://doi.org/10.48550/arXiv.2309.09765>
41. Milan, A., Leal-Taixé, L., Reid, I.D., et al.: Mot16: a benchmark for multi-object tracking. arXiv preprint (2016). <https://doi.org/10.48550/arXiv.1603.00831>
42. Nasser, M.H., Babae, M., Moradi, H., et al.: Online relational tracking with camera motion suppression. *J. Vis. Commun. Image Represent.* **90**, 103750 (2023). <https://doi.org/10.1016/j.jvcir.2022.103750>
43. Pang, Z., Li, Z., Wang, N.: Simpletrack: understanding and rethinking 3d multi-object tracking. In: Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I, pp. 680–696 (2023). https://doi.org/10.1007/978-3-031-25056-9_43
44. Park, J., Hong, J., Shim, W., et al.: Multi-object tracking on swir images for city surveillance in an edge-computing environment. *Sensors* (2023). <https://doi.org/10.3390/s23146373>
45. Qin, Z., Zhou, S., Wang, L., et al.: Motiontrack: learning robust short-term and long-term motions for multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17939–17948 (2023). <https://doi.org/10.48550/arXiv.2303.10404>
46. Rakai, L., Song, H., Sun, S., et al.: Data association in multiple object tracking: a survey of recent techniques. *Expert Syst. Appl.* **192**, 116300 (2022). <https://doi.org/10.1016/j.eswa.2021.116300>
47. Ren, H., Han, S., Ding, H., et al.: Focus on details: online multi-object tracking with diverse fine-grained representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11289–11298 (2023). <https://doi.org/10.1109/CVPR52729.2023.01086>
48. Rezatofighi, S.H., Tsoi, N., Gwak, J., et al.: Generalized intersection over union: a metric and a loss for bounding box regression. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 658–666 (2019). <https://doi.org/10.1109/CVPR.2019.00075>
49. Ristani, E., Solera, F., Zou, R., et al.: Performance measures and a data set for multi-target, multi-camera tracking. In: European conference on computer vision, pp. 17–35 (2016). https://doi.org/10.1007/978-3-319-48881-3_2
50. Said T, Ghoniemy, S., Karam, O.: Real-time multi-object detection and tracking for autonomous robots in uncontrolled environments. In: 2012 Seventh International Conference on Computer Engineering & Systems (ICCES), pp. 67–72 (2012). <https://doi.org/10.1109/ICCES.2012.6408485>
51. Sánchez-Matilla, R., Poiesi, F., Cavallaro, A.: Online multi-target tracking with strong and weak detections. In: Computer Vision – ECCV 2016 Workshops, pp. 84–99 (2016). https://doi.org/10.1007/978-3-319-48881-3_7
52. Singh, D., Kumar, A., Singh, R.: Multiple Object Tracking of Autonomous Vehicles for Sustainable and Smart Cities, Springer Nature Singapore, pp. 201–219 (2023). https://doi.org/10.1007/978-981-99-3288-7_9
53. Stadler, D.: A detailed study of the association task in tracking-by-detection-based multi-person tracking. In: Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. Ed.: J. Beyerer, pp. 59–85 (2023)
54. Stadler, D., Beyerer, J.: Modelling ambiguous assignments for multi-person tracking in crowds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 133–142 (2022). <https://doi.org/10.1109/wacv54805.2022.00019>
55. Stadler, D., Beyerer, J.: An improved association pipeline for multi-person tracking. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 3170–3179 (2023a). <https://doi.org/10.1109/CVPRW59228.2023.00319>
56. Stadler, D., Beyerer, J.: Past information aggregation for multi-person tracking. In: 2023 IEEE International Conference on Image Processing (ICIP), pp. 321–325 (2023b). <https://doi.org/10.1109/icip49359.2023.10223159>
57. Wang, G., Song, M., Hwang, J.N.: Recent advances in embedding methods for multi-object tracking: a survey. arXiv preprint [arXiv:2205.10766](https://arxiv.org/abs/2205.10766) (2022a). <https://doi.org/10.48550/arXiv.2205.10766>
58. Wang, Y., Hsieh, J.W., Chen, P.Y., et al.: Smiletrack: similarity learning for multiple object tracking. arXiv preprint [abs/2211.08824](https://arxiv.org/abs/2211.08824) (2022b). <https://doi.org/10.48550/arXiv.2211.08824>
59. Wang, Z., Zheng, L., Liu, Y., et al.: Towards real-time multi-object tracking. In: European Conference on Computer Vision, pp. 107–122 (2020). <https://doi.org/10.48550/arXiv.1909.12605>
60. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649 (2017). <https://doi.org/10.1109/ICIP.2017.8296962>
61. Xiang, Y., Alahi, A., Savarese, S.: Learning to track: online multi-object tracking by decision making. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4705–4713 (2015). <https://doi.org/10.1109/ICCV.2015.534>
62. Yang, F., Odashima, S., Masui, S., et al.: Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 4788–4797 (2022). <https://doi.org/10.1109/wacv56688.2023.00478>
63. Yang, M.H., Han, G., Yan, B., et al.: Hybrid-sort: weak cues matter for online multi-object tracking. arXiv preprint [abs/2308.00783](https://arxiv.org/abs/2308.00783) (2023). <https://doi.org/10.48550/arXiv.2308.00783>

64. You, S., Yao, H., Bao, B.K., et al.: Utm: a unified multiple object tracking model with identity-aware feature enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21876–21886 (2023). <https://doi.org/10.1109/cvpr52729.2023.02095>
65. Yu, F., Li, W., Li, Q., et al.: Poi: multiple object tracking with high performance detection and appearance feature. In: Computer Vision – ECCV 2016 Workshops, pp. 36–42 (2016). https://doi.org/10.1007/978-3-319-48881-3_3
66. Zeng, F., Dong, B., Zhang, Y., et al.: Motr: end-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision, pp. 659–675 (2022). https://doi.org/10.1007/978-3-031-19812-0_38
67. Zeng, K., You, Y., Shen, T., et al.: Nct: noise-control multi-object tracking. *Complex Intell. Syst.* **9**(4), 4331–4347 (2023). <https://doi.org/10.1007/s40747-022-00946-9>
68. Zhang, Y., Wang, C., Wang, X., et al.: Fairmot: on the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **129**, 3069–3087 (2020). <https://doi.org/10.1007/s11263-021-01513-4>
69. Zhang, Y., Sun, P., Jiang, Y., et al.: Bytetrack: multi-object tracking by associating every detection box. In: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII, pp. 1–21 (2022). https://doi.org/10.1007/978-3-031-20047-2_1
70. Zhang, Y., Chen, H., Bao, W., et al.: Handling heavy occlusion in dense crowd tracking by focusing on the heads. *arXiv preprint* (2023). <https://doi.org/10.48550/arXiv.2304.07705>
71. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European conference on computer vision, pp. 474–490 (2020). <https://doi.org/10.48550/arXiv.2004.01177>
72. Zhou, X., Jia, Y., Bai, C., et al.: Multi-object tracking based on attention networks for smart city system. *Sustain. Energy Technol. Assess.* **52**, 102216 (2022). <https://doi.org/10.1016/j.seta.2022.102216>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.