

On the detection-to-track association for online multi-object tracking

Xufeng Lin^a, Chang-Tsun Li^{a,*}, Victor Sanchez^b, Carsten Maple^c

^a School of Information Technology, Deakin University, Waurn Ponds Campus, Geelong, VIC 3216, Australia

^b Department of Computer Science, The University of Warwick, Coventry, CV4 7AL, UK

^c Warwick Manufacturing Group, The University of Warwick, Coventry, CV4 7AL, UK

ARTICLE INFO

Article history:

Received 3 July 2020

Revised 2 March 2021

Accepted 15 March 2021

Available online 22 March 2021

MSC:

41A05

41A10

65D05

65D17

Keywords:

Detection-to-track association

Multi-object tracking

Track association

ABSTRACT

Driven by recent advances in object detection with deep neural networks, the tracking-by-detection paradigm has gained increasing prevalence in the research community of multi-object tracking (MOT). It has long been known that appearance information plays an essential role in the detection-to-track association, which lies at the core of the tracking-by-detection paradigm. While most existing works consider the appearance distances between the detections and the tracks, they ignore the statistical information implied by the historical appearance distance records in the tracks, which can be particularly useful when a detection has similar distances with two or more tracks. In this work, we propose a hybrid track association (HTA) algorithm that models the historical appearance distances of a track with an incremental Gaussian mixture model (IGMM) and incorporates the derived statistical information into the calculation of the detection-to-track association cost. Experimental results on three MOT benchmarks confirm that HTA effectively improves the target identification performance with a small compromise to the tracking speed. Additionally, compared to many state-of-the-art trackers, the DeepSORT tracker equipped with HTA achieves better or comparable performance in terms of the balance of tracking quality and speed.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Multi-object tracking (MOT), which aims to track multiple objects of interest in video sequences, is an essential building block of a wide range of advanced applications such as video surveillance and autonomous driving. Driven by the great success of deep neural networks in object detection, the *tracking-by-detection* paradigm has gained substantial attention in recent years. Such a paradigm first detects targets of interest in single video frames and then builds up tracks of targets by associating the detections. Thus, lying at the heart of tracking-by-detection based MOT is the detection-to-track association. To avoid any confusion, a 'target' in this manuscript refers to an object of interest that appears, moves and disappears in a camera's field of view, and a 'track' refers to a target's trajectory or path, which records all the information, including ID, positions and appearance features, associated with the target. The works proposed to address the data association problem can be categorized into *offline* (a.k.a *batch*) and *online* approaches. Offline approaches [1–3] typically formulate the data association as a global optimization problem by considering the detections over a batch of frames or all frames. They are effective

in overcoming issues like unreliable detections, similar appearance, and frequent occlusions, but their high computational cost and non-causal fashion of tracking make them unsuitable for time-critical applications. By contrast, the online approaches [4,5] sequentially link the detections with tracks on a frame-by-frame basis by only considering the information up to the present frame, thus compared to the offline approaches, they are usually much more efficient but are more prone to association errors.

Like in many other computer vision tasks, the accuracy-efficiency trade-off for MOT is quite profound. Higher accuracy of detection and association usually comes with a higher computational cost. Notably, many existing works tend to place little emphasis on tracking efficiency. Complicated modules and ad-hoc twists are usually devised to improve tracking accuracy. This is most likely because the performance ranking on the MOT benchmarking datasets is largely based on the tracking accuracy. However, for many practical applications, such as natural disasters monitoring and video surveillance systems, the tracking speed is essential for the success of the overall system. Another important aspect of the tracking performance that is sometimes overlooked is the accuracy and consistency of target identification. In practice, reliable identity information about the tracked targets is key for many high-level tasks such as abnormal behavior analysis and trajectory predictions.

* Corresponding author.

E-mail address: changtsun.li@deakin.edu.au (C.-T. Li).

Existing MOT methods typically perform detection-to-track association by exploiting multiple cues, including the motion and appearance information. The appearance information, which is usually represented as feature vectors produced by an appearance model, is important for accurate long-term tracking and identification. However, most existing association methods only consider the appearance feature distances calculated at the current frame and ignore the historical appearance feature distance records, which are also useful for track association. In this paper, we focus on the online track association based on appearance information and propose a hybrid track association (HTA) algorithm that enables more accurate and robust detection-to-track association with a small compromise to the tracking speed. We model the historical appearance distance records between the detections and tracks with an efficient incremental Gaussian mixture model (IGMM) [6]. The statistical information derived from the IGMM is then used as auxiliary information for the association cost calculation based on appearance distance.

2. Related work

Given the vast variety of studies on MOT, we will only review those that are most closely related to our work.

Bewley *et al.* [4] proposed a simple online and real-time tracking (SORT) algorithm, which only uses the Intersection over Union (IoU) distance between the predicted detections of the tracks (with a Kalman filter) and the detections in the current frame for the association. The appearance information and the potential occlusions are ignored in this framework. Therefore, it is fast and can easily reach a processing speed of 60~100 frames per second (FPS). However, due to the lack of appearance information in the association process, SORT shows poor identification performance in scenes with moderate occlusions.

Wojke *et al.* [5] proposed a DeepSORT framework, which extends the SORT framework by introducing appearance information in the association process with a deep convolutional neural network (CNN) trained offline on large-scale image datasets. As the appearance information is typically represented by fixed-length feature vectors produced by an appearance model, e.g. the CNN-based person re-identification model in [5], we will use “appearance information” and “appearance feature” interchangeably hereinafter. Another import contribution of DeepSORT is the *cascade matching strategy* (CMS) [5], which gives higher priorities to tracks that have been seen more recently. The authors argued that CMS is effective in reducing the incorrect association due to temporary and long-term occlusions. In fact, many earlier works [7–10] have pointed out the importance of appearance information in the detection-to-track association. However, few studies have been dedicated to investigating the effects of different appearance-based association methods on tracking performance.

Recently, Wang *et al.* [11] proposed to handle the detection-to-track association based on the distances between the appearance features of the detections and the ‘smoothed’ feature of each track, which is an exponential moving average of the appearance features of temporally adjacent detections in the same track. This strategy considers the temporal relationship of detections and thus is expected to be more accurate than CMS. More recently, Han *et al.* [12] proposed to exploit the complementary information of a pair of synchronized videos captured, respectively, from a top view in a high altitude, e.g. by a drone-mounted camera, and from a horizontal view, e.g. by a helmet-mounted camera, to improve the accuracy of track association for MOT. However, it requires that both the horizontal- and top-view videos are available and synchronized prior to tracking, which may not be applicable in some practical scenarios.

Unlike the aforementioned methods that only consider the distance between appearance features at the current frame, in this work we propose a hybrid track association algorithm that also incorporates the statistical information of the historical appearance distance records. Our work bears some similarity to the association method based on tracklet confidence [9]. In that, the average of the pairwise historical appearance distances of a track is used to measure the track quality (or confidence). Our method is essentially different from [9] in twofold. Firstly, we use an IGMM to efficiently estimate the distribution, rather than the average score in [9], of the historical distances. Secondly, the estimated IGMM is used to provide extra information for the calculation of association cost, while in [9] the average score is used to determine which tracks will be associated preferentially.

3. Hybrid track association

3.1. Motivation

We denote the ℓ -dimensional appearance feature of a detection i appearing at frame t as \mathbf{f}_t^i and the historical appearance features of track j up to frame t as $T_t^j = \{\mathbf{f}_s^j\}$, where s denotes the frame indices where the target corresponding to track j has appeared. To associate the detections at frame t with existing tracks, the appearance feature of each detection is compared with the appearance features of each track. This results in a cost matrix with each element $d_t^{i,j} = \mathcal{F}(\mathbf{f}_t^i, T_t^j)$ being the association cost of assigning detection i to track j , where \mathcal{F} is a function for measuring the appearance distance between a detection and a track. For instance, if we use the minimum cosine distance, then $\mathcal{F}(\mathbf{f}_t^i, T_t^j) = \min_s \left(1 - \frac{\mathbf{f}_t^i \cdot \mathbf{f}_s^j}{\|\mathbf{f}_t^i\|_2 \|\mathbf{f}_s^j\|_2} \right)$. With the cost matrix, the detection-to-track association can be modeled as an assignment problem and solved using the Hungarian algorithm. To reduce the risk of assigning false detections or newly emerged objects to existing tracks, a permissible maximum distance d_{\max} is used to ignore any association with a cost exceeding d_{\max} .

The above distance-based association only considers the appearance distances at the current frame t . However, the historical appearance distance records may also provide useful information for reducing the ambiguities in the detection-to-track association. An example is shown in Fig. 1, where the ground-truth is that detection i belongs to Track 2 at frame t . However, due to the outliers appearing in the tracks or the limited discriminative power of the appearance features, detection i has a smaller distance to Track 1 than to Track 2, i.e. $d_t^{i,1} < d_t^{i,2}$. Consequently, the above distance-based assignment will wrongly associate detection i with Track 1 rather than Track 2. Meanwhile, if we look at the corresponding histograms of the historical distance records of the two tracks, it becomes evident that the likelihood of $d_t^{i,2}$ belonging to the distance distribution of Track 2 is higher than that of $d_t^{i,1}$ belonging to the distance distribution of Track 1, i.e. $p(d_t^{i,1} | \text{Track1}) < p(d_t^{i,2} | \text{Track2})$, which implies that detection i is more likely to belong to Track 2. Such statistical information can be exploited to correct the inaccurate associations when multiple tracks with marginally different distances are competing for the same detection. This motivates us to model the historical distance records and use the derived statistical information to aid the detection-to-track association for MOT. To achieve this purpose, when a detection i is assigned to a track j at a specific frame t , we store not only the appearance feature \mathbf{f}_t^i but also the corresponding appearance feature distance $d_t^{i,j}$. As track j builds up, this will generate the historical appearance distance records $D^j = \{d_t^{i,j}\}$, where t is the frame indices where detection i is found to be matched with track j . For brevity, we will drop the superscripts i and j in the follow-

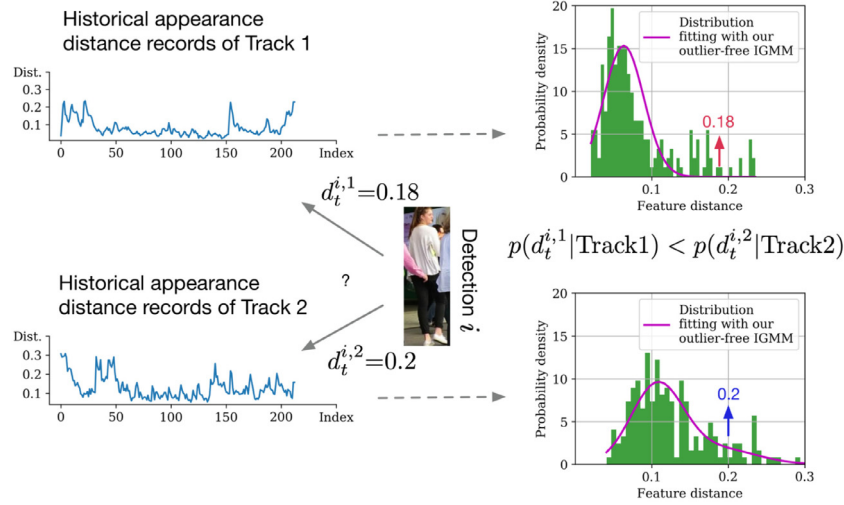


Fig. 1. Exploiting the statistical information of the historical appearance distance records of a track for the detection-to-track association. The two plots on the left side, respectively, show the historical distance records for Track 1 and Track 2. The corresponding histograms are shown on the right. The distance-based assignment will wrongly assign detection i to Track 1 as $d_t^{i,1} < d_t^{i,2}$, while looking at the two histograms, we can see that the likelihood of $d_t^{i,1}$ belonging to the distance distribution of Track 2 is higher than that of $d_t^{i,1}$ belonging to the distance distribution of Track 1, i.e. $p(d_t^{i,1}|\text{Track1}) < p(d_t^{i,1}|\text{Track2})$. Such extra information provides a chance to correct the wrong assignment.

ing analysis and denote the historical appearance distance records of a track by $D = \{d_n\}$, where $n = 1, 2, \dots$ is the internal index of the appearance distance records in D .

3.2. Incremental gaussian mixture model

Before delving into the details of modeling the historical distance records of a track, let us turn our attention to its relationship with foreground objects detection in video surveillance. Detecting foreground objects is usually accomplished by establishing a statistical background model at each pixel location and the pixel values that do not fit the model are considered as ‘foreground’. Similarly, if we consider the historical appearance distances of a track as pixel ‘values’ appearing at a ‘track location’ over time, modeling the historical distances of a track is equivalent to modeling the background pixel values at a pixel location. For this reason, we adopt the Gaussian mixture model, which is widely used for background modeling, to model the historical appearance distances of a track. Considering the time-series nature of the online association and the real-time requirement of tracking systems, we develop our algorithm based on a fast incremental Gaussian mixture model (IGMM) [6].

Following the work in [13], we model the cosine distance or squared Euclidean distance¹ between two ℓ -dimensional unit-norm appearance features as a chi-square distribution with ℓ degrees of freedom (see the histograms in Fig. 1). However, it is unsuitable to fit a non-norm chi-square distribution with an IGMM, so we take the fourth root of the chi-square random variable to approximately transform the chi-square distribution to a normal distribution, as suggested in [14]. As we can see in Fig. 2, transforming to a normal distribution allows for a more accurate fitting with IGMM. Suppose that we have the historical distance records $D = \{d_n\}$ for a track, where n is the internal index of the data points in D and d_n is the fourth root of the distance between the appearance features of a track and a detection assigned to it. To model the distribution of the data points in D , we use the following Gaussian mixture model with K_n components:

$$p(d_n|\Theta_n) = \sum_{k=1}^{K_n} \pi_{k,n} p(d_n|\mu_{k,n}, \sigma_{k,n}^2), \quad (1)$$

where $\pi_{k,n}$, $\sum_{k=1}^{K_n} \pi_{k,n} = 1$ is the mixture weight for the k^{th} component when the n^{th} data point arrives and $p(d_n|\mu_{k,n}, \sigma_{k,n}^2)$ is the k^{th} component defined as a Gaussian distribution with mean $\mu_{k,n}$ and variance $\sigma_{k,n}^2$:

$$p(d_n|\mu_{k,n}, \sigma_{k,n}^2) = \frac{1}{\sqrt{2\pi\sigma_{k,n}^2}} \exp\left(-\frac{(d_n - \mu_{k,n})^2}{2\sigma_{k,n}^2}\right). \quad (2)$$

According to the Bayes’ rule, the posterior probability of d_n belonging to the k^{th} component can be computed as

$$p(z_k = 1|d_n) = \frac{\pi_{k,n} p(d_n|\mu_{k,n}, \sigma_{k,n}^2)}{\sum_{m=1}^{K_n} \pi_{m,n} p(d_n|\mu_{m,n}, \sigma_{m,n}^2)}, \quad (3)$$

where $\mathbf{z} = (z_1, \dots, z_{K_n})$ is a vector of K_n mutually exclusive binary variables with $z_k = 1$ indicating the k^{th} component is responsible for generating the data point.

When a new data point d_n arrives, i.e. a new detection is assigned to a track, the incremental estimation process is carried out by executing either Procedure 1: creating a new component or Procedure 2: updating the existing component(s), followed by an additional Procedure 3) removing spurious components.

Procedure 1: creating a new component

If there are no existing components (i.e. $K_n = 0$ before the first data point arrives) or the update criterion in Procedure 2 is not met, a new component $k = K_n + 1$ is created and initialized:

$$\begin{cases} v_{k,n+1} = 1, N_{k,n+1} = 1, \mu_{k,n+1} = d_n, \\ K_{n+1} \leftarrow K_n + 1, \sigma_{k,n+1}^2 = \sigma_{ini}^2, \pi_{k,n+1} = 1/\sum_{m=1}^{K_{n+1}} N_{m,n+1}, \end{cases} \quad (4)$$

where $N_{k,n}$ is the accumulated posterior probability of the k^{th} component when the n^{th} data point arrives, $v_{k,n}$ is a counter used for determining whether the k^{th} component is spurious or not (see Procedure 3 below), and σ_{ini}^2 is the user-specified initial variance of a component depending on the nature of the appearance feature.

Procedure 2: updating existing component(s)

When the squared Mahalanobis distance between a new data point d_n and any component k ($1 \leq k \leq K_n$) is smaller than $\chi_{1,1-\tau}^2$, i.e. the $1 - \tau$ percentile of a chi-square distribution with 1 degree of freedom, we update the existing components with d_n rather

¹ It is easy to show that they are equivalent in the case of unit-norm vectors.

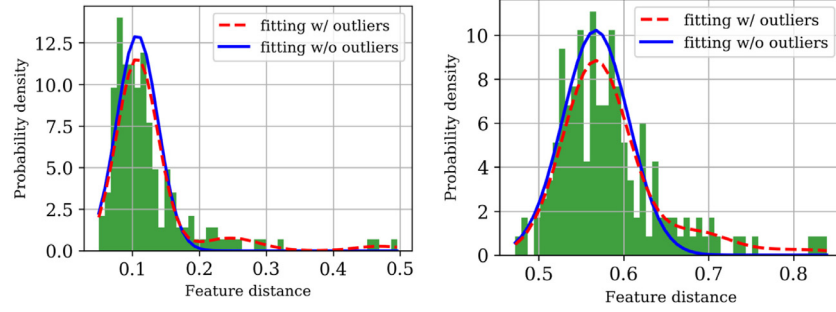


Fig. 2. Fitting a chi-square distribution (left) and the fourth root of a chi-square distribution (right) with an IGMM. The dashed red and solid blue curves show the fitting results for all data and data without outliers, respectively.

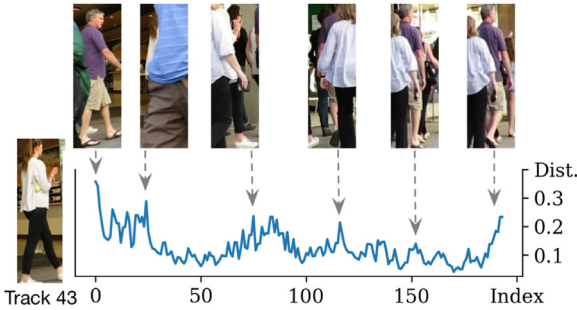


Fig. 3. Examples of outliers appearing in Track 43 of sequence MOT16-09 in the MOT16 benchmark. These outliers are typically caused by occlusions or inaccurate detections, which result in relatively higher values of d_n .

than creating a new component:

$$\begin{cases} v_{k,n+1} = v_{k,n} + 1 \\ N_{k,n+1} = N_{k,n} + p(z_k = 1 | d_n) \\ \mu_{k,n+1} = \mu_{k,n} + \xi_{k,n}(d_n - \mu_{k,n}) \\ \sigma_{k,n+1}^2 = \sigma_{k,n}^2 - \xi_{k,n}(\sigma_{k,n}^2 - (d_n - \mu_{k,n+1})^2) - \omega_{k,n}^2(d_n - \mu_{k,n})^2 \\ \pi_{k,n+1} = N_{k,n+1} / \sum_{m=1}^{K_n} N_{m,n+1}, \end{cases} \quad (5)$$

where $\xi_{k,n}$ is a coefficient accounting for the contribution of d_n to the update of the k^{th} component:

$$\xi_{k,n} = \frac{p(z_k = 1 | d_n)}{N_{k,n+1}}. \quad (6)$$

The interfering outliers (e.g. due to occlusions and incorrect detections as shown in Fig. 3) may also form additional components with small mixture weights. If we increase the number of components without limitation, these small components may eventually overwhelm the importance of the components corresponding to the ground truth detections. Thus, similarly to the work in [15], we set a maximum number of components $K_{\max} = 5$ and discard the component with the smallest $\pi_{k,n}$ when K_{\max} is reached.

Procedure 3: removing spurious components

A component k is considered spurious and removed whenever $v_{k,n} > v_{\min}$ and $N_{k,n} < N_{\min}$. These two parameters suggest that: if at least v_{\min} data points have been recorded since component k was created, i.e. $v_{k,n} > v_{\min}$, a sufficiently significant change in component k in terms of the accumulated posterior probability still cannot be observed, i.e. $N_{k,n} < N_{\min}$, then component k is deemed to be spurious and should be removed. As in [6], we set $v_{\min} = 5$ and $N_{\min} = 3$. Each of the above three procedures is followed by renormalization to ensure that $\sum_{m=1}^{K_{n+1}} \pi_{m,n+1} = 1$.

3.3. Track association

As some outliers are inevitably introduced to a track due to occlusions or inaccurate detections (see the examples in Fig. 3), we need to determine which components are most likely to be generated by the ground truth detections in the track. Heuristically, a detection belonging to the track tends to have a smaller appearance distance. Therefore, we sort the components of the estimate mixture model by the mean $\mu_{k,n}$ in descending order and choose the first M components that account for at least a portion Υ of the model:

$$M = \arg \min_m \left(\sum_{k=1}^m \pi_{k,n} > \Upsilon \right), \quad (7)$$

where $\Upsilon \in [0, 1]$ is the minimum portion of the data that should be accounted for the ground truth detections of a track. In other words, at most $1 - \Upsilon$ portion of the data is considered as ‘outliers’. A small Υ usually results in a unimodal IGMM, while a high Υ could result in a multi-modal IGMM, which allows for modeling more dynamic distributions, e.g. caused by scenes where tracked targets change pose constantly. We empirically set $\Upsilon = 0.8$ in our experiments. An example is shown in Fig. 2, where the dashed red and solid blue curves show the fitting results for all data and data without ‘outliers’, respectively.

With the above IGMM incrementally estimated for each track, the remaining questions are when and how to integrate such ‘statistical’ information with the distance-only based track association method. Like many statistical models, the IGMM model will only be statistically reliable when sufficient data has been observed. For this reason, we set a minimum track length $\mathcal{L} = 15$ to ensure that an IGMM will be estimated only for the tracks with a length no less than \mathcal{L} . For the tracks with a length less than \mathcal{L} , we only use the appearance distance as the association cost. Otherwise, we use the following hybrid association cost, hence the name *Hybrid Track Association* (HTA):

$$C_t = \lambda d_t + (1 - \lambda) \frac{\sum_{k=1}^M \pi_{k,n} \int_{-\infty}^{d_t} p(x | \mu_{k,n}, \sigma_{k,n}^2) dx}{\sum_{k=1}^M \pi_{k,n}}, \quad (8)$$

where d_t is the distance-based term representing the appearance distance between a detection and a track at the current frame t . The probability-based term in the box is the cumulative probability of the outlier-free IGMM estimated with the appearance distance records observed for the track. λ acts as a weighting factor to balance the importance of the distance-based and the probability-based terms. As can be expected, the probability-based term is not as discriminative as the distance-based term because the IGMM is only the statistical summary of the historical distances. We thus use a large weighting factor $\lambda = 0.9$.

4. Experiments

4.1. Implementation details

Our implementation is based on the DeepSORT tracker [5], which is mainly composed of a pedestrian detector and a track association module relying on the information provided by an appearance model and a motion estimation model based on Kalman filtering. Our proposed HTA algorithm resides in the track association module and exploits the statistical information of historical appearance distances of a track. We use YOLOv4 [16] for pedestrian detection. To make the detector more generalizable, we train it on a large-scale dataset (109,471 images in total) consisting of 4 public pedestrian datasets: the Caltech dataset [17], CUHK-SYSU dataset [18], PRW dataset [19] and CrowdHuman dataset [20]. For the appearance model, we adopt the CNN architecture proposed in [5] and train it on the MARS person re-identification dataset [21], which contains 1,067,516 bounding boxes of 1261 pedestrians. The appearance model outputs a 512-dimensional feature for each detected bounding box. All the experiments were conducted on a PC with an NVIDIA GeForce RTX2080 Ti GPU (11 GB VRAM), Intel Core i7-8086K CPU (6 cores, 4.0 GHz), and 32 GB RAM.

4.2. Evaluation metrics

To evaluate the performance of MOT, we adopt the following metrics as proposed in [22] and [23]:

- Identification F1 score (IDF1 \uparrow): The harmonic mean of identification precision (the ratio of the computed detections that are correctly identified) and recall (the ratio of ground truth detections that are correctly identified).
- Multiple Object Tracking Accuracy (MOTA \uparrow): Overall tracking accuracy computed by combining three sources of errors: the number of false positive detections (FP \downarrow), the number of false negative (i.e. missed) detections (FN \downarrow), and the number of identity switches (IDS \downarrow).
- Multiple Object Tracking Precision (MOTP \uparrow): Average intersection over union between the true positive detections and their corresponding ground truth targets.
- Mostly Tracked (MT \uparrow): Percentage of the ground-truth targets correctly tracked for at least 80% of their life span.
- Mostly Lost (ML \downarrow): Percentage of the ground-truth targets correctly tracked for at most 20% of their life span.
- Fragmentations (Frag \downarrow): Number of times a ground truth track changes its status from 'tracked' to 'untracked'.

Here, ' \uparrow ' and ' \downarrow ' respectively represent that higher and lower values are preferred. IDF1 and MOTA have been widely accepted as the two most important gauges of tracking performance. They measure different aspects of a tracker: IDF1 emphasizes the accuracy and consistency of target identification, while MOTA is closely related to the detection performance.

4.3. Benchmarking datasets

For the evaluation of tracking performance, we use three MOT benchmark datasets: 2D MOT15 [31], MOT16 [22], and MOT17 [22]. These three benchmarks, respectively, contain 22 (11 training, 11 test), 14 (7 training, 7 test), and 14 (7 training, 7 test) videos sequences in unconstrained environments filmed with both static and moving cameras. The ground truth annotations of the training sequences are released but those of the test sequences are unpublished to avoid over-fitting to the specific sequences [22]. However, because the sequences in the training and test sets of these three benchmarks were captured in the same or similar environments, training on the training sets is beneficial for improving the

tracking performance on the test sets. For this reason, many top-ranked trackers on the MOT benchmarks [22,31] are trained on the training sets. It is noteworthy that we do not train on the training sets of the MOT benchmarks nor train on any dataset, e.g. the ETH dataset [32] and PETS dataset [33], that partially overlaps with the MOT benchmarks. This prevents the detector from being biased to the MOT benchmarks and allows us to conduct fair performance analysis on the training sets.

4.4. Performance analysis

In this section, we will investigate the effect of the parameters of the proposed HTA algorithm and compare different track association methods. Experiments are conducted on the training sets of MOT15, MOT16, and MOT17. For MOT16 and MOT17, the detection score threshold for the detector is set to 0.3, while for MOT15, the threshold is set to 0.7 due to the higher number of false positives. As in [5], we use the cosine distance to measure the distance between appearance features and set the permissible maximum distance $d_{\max}=0.2$. We set $\sigma_{ini}^2=0.005$ in Eq. (4) for the appearance feature used in [5].

We create four different trackers by replacing the appearance-based association module in the DeepSORT tracker [5] with four different track association strategies. The details of these trackers are as below:

- Cascade Matching Strategy (CMS) [5]: This association strategy first matches the detections with the tracks that are most recently updated, and then the tracks that are second-most recently updated and so on, until all the tracks or detections are examined. The cost of associating a detection with a track is calculated as the appearance distance between the detection and the nearest detection (i.e. with the smallest distance) stored in the track.
- k Nearest Neighbors (k NN): This strategy associates the detections with all existing tracks at once without giving priorities to any tracks. For each detection, the association cost is the average appearance distance between the detection and its top k nearest detections in each of the tracks. If the length of a track is smaller than k , all the detections in the track are considered. We set $k=5$ to strike a balance between the robustness to outliers and the accuracy of association.
- Exponential Moving Average (EMA) [11,24]: This strategy performs association based on the distance between the appearance features of the detections and the 'smoothed' feature of each track, which is an exponential moving average of the temporally adjacent appearance features in the same track. Unlike CMS and k NN, this strategy takes into consideration the temporal information and thus is expected to be more accurate than CMS and k NN. As in [11,24], we set the weighting-decrease coefficient $\eta = 0.9$.
- Hybrid Track Association (HTA): This strategy differs from EMA in that it considers not only the feature distance at the current frame but also the statistical information provided by the historical appearance distance records of a track.

We first investigate the effect of the parameters of the proposed HTA algorithm on the tracking performance. There are two important parameters for the HTA algorithm: the minimum track length \mathcal{L} for reliably estimating an IGMM and the weighting factor λ in Eq. (8). We investigate the effect of these two parameters on the IDF1 tracking performance using the training sets of MOT15. Fig. 4 a and Fig. 4 b show the results for fixed $\lambda = 0.9$ and varying $\mathcal{L} \in [5, 100]$ and for fixed $\mathcal{L} = 15$ and varying $\lambda \in [0, 1]$, respectively. As shown in Fig. 4 a, the performance improves until the minimum track length \mathcal{L} reaches 15 and trends downward as \mathcal{L} continues to increase. This is not surprising because integrating

Table 1

Performance comparison of four different track association methods on the **training** sequences of 2D MOT15, MOT16, and MOT17. The best results for each metric are highlighted in bold. The frames per second (FPS) in the last column measures the speed of the entire algorithm including detection and association.

	Tracker	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓	FPS↑
MOT15	CMS [5]	59.7%	65.3%	78.9%	54.6%	16.8%	5282	9482	211	643	~23
	kNN ($k=5$)	63.0%	65.3%	79.0%	55.1%	17.5%	5359	9413	209	641	~23
	EMA [11,24]	64.1%	65.4%	79.0%	55.1%	17.0%	5283	9418	215	642	~24
	HTA	67.7%	65.1%	79.0%	53.5%	17.2%	5192	9580	278	647	~24
MOT16	CMS [5]	57.3%	54.6%	76.8%	33.2%	20.5%	9964	39,529	633	1513	~22
	kNN ($k=5$)	60.4%	54.4%	76.9%	35.0%	20.1%	10,299	39379	657	1528	~21
	EMA [11,24]	60.6%	54.5%	76.8%	34.6%	20.9%	10,235	39,420	633	1545	~21
	HTA	62.7%	54.3%	76.8%	33.3%	20.3%	9991	39,735	647	1526	~19
MOT17	CMS [5]	57.0%	54.6%	76.8%	31.9%	23.3%	9329	41,068	641	1529	~19
	kNN ($k=5$)	60.2%	54.4%	76.9%	32.8%	22.9%	9658	40912	659	1546	~18
	EMA [11,24]	60.4%	54.4%	76.9%	32.4%	23.4%	9585	40,946	637	1560	~19
	HTA	62.5%	54.3%	76.9%	33.1%	24.0%	9358	41,275	660	1539	~18

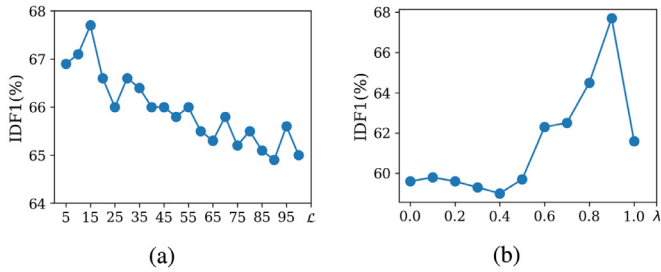


Fig. 4. Tracking performance for various values of (a) the minimum track length L for reliably estimating an IGMM and (b) the weighting factor λ (right) in Eq. (8).

the ‘statistical’ information derived from the IGMM is more beneficial in the earlier period of a track, when the appearance cues have not been sufficiently collected. In Fig. 4 b, $\lambda = 0$ and $\lambda = 1$, respectively, correspond to the probability-only and distance-only associations. We can see that the peak performance is achieved at $\lambda = 0.9$. We will use $L = 15$ and $\lambda = 0.9$ for the HTA algorithm in the following experiments.

We then compare the tracking performance of different track association methods. The results on MOT15, MOT16, and MOT17 are reported in Table 1. As we can see, there is no substantial difference in MOTA for the four trackers as the MOTA is highly dependent on the detection performance. While in terms of IDF1, the proposed HTA outperforms the other three algorithms consistently, followed by EMA and kNN. CMS proposed in [5] is the worst-performing tracker, with about 6%~8% lower IDF1 than HTA and 3%~4% lower IDF1 than EMA and kNN. Based on these results, we can gain some insights for track association: 1) CMS is not as effective as expected in improving MOTA and tends to have a negative impact on IDF1. 2) The performance gap between HTA & EMA and CMS & kNN implies that temporal information is important for accurate track association. 3) Integrating the historical information of a track is beneficial for enhancing the performance of target identification. We also note that HTA may give rise to more frequent ID switching. A qualitative analysis is shown in Fig. 5, where HTA makes 2 ID switches and EMA makes only 1 ID switch. Despite EMA’s lower occurrences of ID switching, the wrong ID assignment at frame 432 persists until the end of the track due to the lack of an effective error correction mechanism. While for HTA, the integration of the statistical information of historical records gives a chance to quickly correct the wrong ID assignment at frame 285. As for the tracking speed, HTA and kNN are slightly slower than CMS and EMA, with about 1~3 FPS slower, because extra computation is needed for estimating the IGMM or searching for the top k nearest appearance features of a track.



Fig. 5. Qualitative evaluation on the sequence MOT16-09. This example shows how the ID of a pedestrian changes from frame 244 to frame 432 for two trackers, EMA (1st row) and HTA (2nd row). The number on each frame is the enlarged ID assigned to the pedestrian and a color change of the number indicates an ID switch. Note that for the same video sequence, different trackers may assign different IDs to the same person.

4.5. Comparisons with state-of-the-art trackers

We compare the aforementioned four trackers with several state-of-the-art *online* trackers on the **test** sequences of MOT16 benchmark. All the algorithms are run under the private protocol, i.e. using private detectors. The comparison results are presented in Table 2. For CMS, kNN, EMA, and HTA, we use the same parameter settings as on the **training** sequences. The best results for these four trackers are highlighted in bold, while the best results for other trackers are underlined. We also show in the last two rows of Table 2 the comparison results between HTA and the association method based on tracklet confidence and online discriminative appearance learning (TCODAL) [9] on 4 sequences of MOT15, namely PETS09-S2L1, PETS09-S2L2, ETH-Bahnhof and ETH-Sunnyday. Note that the tracking speed of TCODAL shown in Table 2 does not include the time used for detection.

Compared to other state-of-the-art trackers, the proposed HTA achieves better or comparable performance in terms of the balance of tracking quality and speed. We can see that many top-performing trackers (e.g. those indicated with superscript \star) use the person detector provided in [10], which is based on the faster-RCNN [25] and trained on both public and private datasets. The faster-RCNN detector delivers high-quality detections, as reflected by the high MOTA and relatively low FP and FN of POI in the second row of Table 2, but it also substantially compromises the speed of tracking. For instance, most of them can only process less than 8

Table 2

Performance comparison with state-of-the-art online trackers on the test sequences of MOT16 and a subset of MOT15. The symbol '*' represents that the tracker uses the faster-RCNN [25] based person detector provided by POI [10], the symbol '§' means that the tracker is trained on the training sequences of MOT16, and the symbol '▲' means that the results are obtained on 4 sequences of MOT15, namely PETS09-S2L1, PETS09-S2L2, ETH-Bahnhof and ETH-Sunnyday. In each column (excluding the last two rows), the bold and underlined values represent the best result of the four association methods and the best result of other methods, respectively. Except for TCODAL [9], the frames per second (FPS) in the last column measures the speed of the entire algorithm including detection and association.

Tracker	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓	FPS↑
POI* [10]	66.1%	65.1%	79.5%	34.0%	20.8%	5061	55,914	805	3093	<5
TAP* [26]	<u>73.5%</u>	64.8%	78.7%	38.5%	21.6%	12,980	50,635	571	1048	<8
SORT* [4]	53.8%	59.8%	79.6%	25.4%	22.7%	8698	63,245	1423	1835	<12
VMaxx* [27]	49.2%	62.6%	78.3%	32.7%	21.1%	10,604	56,182	1389	1534	~7
DeepSORT* [5]	62.2%	61.4%	79.1%	32.8%	18.2%	12,852	56,668	781	2008	<8
LM-CNN§* [28]	61.2%	67.4%	79.1%	38.2%	19.2%	10,109	48,435	931	1034	~2
EAMTT§ [29]	53.3%	52.5%	78.8%	19.0%	34.9%	<u>4407</u>	81,223	910	1321	~12
CNNMTT§ [30]	62.2%	65.2%	78.4%	32.4%	21.3%	6578	55,896	946	2283	<6
JDE-1088§ [11]	55.8%	64.4%	--	35.4%	20.0%	--	--	1544	--	~19
FairMOT§ [24]	70.4%	<u>68.7%</u>	<u>80.3%</u>	<u>39.5%</u>	<u>19.0%</u>	11,695	<u>44411</u>	953	2424	~26
CMS [5]	58.7%	62.5%	79.2%	39.3%	11.5%	19,284	47,570	1456	2510	~16
kNN (k=5)	61.3%	62.3%	62.3%	39.4%	11.3%	19,897	47116	1692	2591	~15
EMA [11,24]	61.7%	62.3%	79.2%	39.3%	11.5%	19,876	47,182	1757	2635	~17
HTA	64.2%	62.4%	79.3%	37.5%	12.1%	19071	47,839	1619	2529	~15
TCODAL▲ [9]	--	74.3%	62.9%	79.8%	1.5%	--	--	85	132	<0.5
HTA▲	69.2%	78.4%	78.5%	76.8%	5.9%	2607	2747	219	455	~30

frames per second, which makes them unsuitable for time-critical applications.

The two algorithms that are faster than our proposed HTA are JDE-1088 [11] and FairMOT [24]. Both trackers use a joint framework that shares the feature maps for the tasks of object detection and appearance feature learning, thus boosting the tracking speed. They mainly differ in the backbone network and object detection mechanism (i.e. anchor-based or anchor-free). It is noteworthy that both of JDE-1088 and FairMOT are trained on the training sets of MOT benchmarks. While this is beneficial for improving the performance on the test sets of MOT benchmarks, it may lead to overfitting to the MOT benchmarks for both detection and target identification. For instance, we have observed a large performance gap (in terms of miss detections and ID switches) for FairMOT when evaluated on videos that are quite different from the sequences in MOT benchmarks (e.g. the videos downloaded from YouTube), but the lack of tracking annotations on these datasets prevents us from providing quantitative results here. This is an issue that should not be overlooked because training on self-collected datasets may be infeasible for many practical scenarios. This is the reason why we train our pedestrian detector and appearance model on datasets that do not overlap with the MOT benchmark datasets. This enables our trackers to deliver relatively consistent performance in a plug-and-play way for unseen datasets.

5. Conclusions

In this paper, we have presented a hybrid track association method that enables a more accurate and robust online detection-to-track association. Our proposed method efficiently models the historical appearance distance records of a track with an incremental Gaussian mixture model and integrates the derived statistical information into the calculation of track association cost. Evaluations on public multi-object tracking benchmarks demonstrate that, with detections provided by a real-time object detector, our proposed hybrid track association strategy achieves better or comparable performance than many other state-of-the-art online trackers in terms of the balance between tracking speed and quality.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is jointly supported by the Defence Science and Technology Laboratory (DSTL) of the Ministry of Defence of the United Kingdom through the project entitled R-DIPS: Real-time Detection of Concealment of Intent for Passenger Screening (Project No. ACC6008031) and PETRAS National Centre of Excellence for IoT Systems Cybersecurity (Project No. EP/S035362/1).

References

- [1] J. Berclaz, F. Fleuret, E. Turetken, P. Fua, Multiple object tracking using k-shortest paths optimization, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (9) (2011) 1806–1819.
- [2] A.A. Butt, R.T. Collins, Multi-target tracking by lagrangian relaxation to min-cost network flow, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1846–1853.
- [3] S. Tang, M. Andriluka, B. Andres, B. Schiele, Multiple people tracking by lifted multicut and person re-identification, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3539–3548.
- [4] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 3464–3468.
- [5] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3645–3649.
- [6] R.C. Pinto, P.M. Engel, A fast incremental gaussian mixture model, *PLoS ONE* 10 (10) (2015) e0139931–e0139931.
- [7] C.-H. Kuo, C. Huang, R. Nevatia, Multi-target tracking by on-line learned discriminative appearance models, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 685–692.
- [8] B. Yang, R. Nevatia, Online learned discriminative part-based appearance models for multi-human tracking, in: *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 484–498.
- [9] S.-H. Bae, K.-J. Yoon, Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1218–1225.
- [10] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, J. Yan, POI: Multiple object tracking with high performance detection and appearance feature, in: *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 36–42.
- [11] Z. Wang, L. Zheng, Y. Liu, S. Wang, Towards real-time multi-object tracking, 2019 arXiv preprint arXiv:1909.12605.
- [12] R. Han, W. Feng, J. Zhao, Z. Niu, Y. Zhang, L. Wan, S. Wang, Complementary-view multiple human tracking, in: *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10917–10924.

- [13] Y. Huang, F. Zhu, Z. Zeng, X. Qiu, Y. Shen, J. Wu, SQE: A self quality evaluation metric for parameters optimization in multi-object tracking, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8306–8314.
- [14] D.M. Hawkins, R. Wixley, A note on the transformation of chi-squared variables to normality, *Am. Stat.* 40 (4) (1986) 296–298.
- [15] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2, 1999, pp. 246–252.
- [16] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: optimal speed and accuracy of object detection, 2020 arXiv preprint arXiv:2004.10934.
- [17] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: A benchmark, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 304–311.
- [18] T. Xiao, S. Li, B. Wang, L. Lin, X. Wang, End-to-end deep learning for person search, arXiv preprint arXiv:1604.01850 2 (2) (2016).
- [19] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, Q. Tian, Person re-identification in the wild, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1367–1376.
- [20] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, J. Sun, Crowdhuman: a benchmark for detecting human in a crowd, 2018 arXiv preprint arXiv:1805.00123.
- [21] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, Q. Tian, MARS: A video benchmark for large-scale person re-identification, in: *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 868–884.
- [22] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler, MOT16: A benchmark for multi-object tracking, 2016 arXiv preprint arXiv:1603.00831.
- [23] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in: *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 17–35.
- [24] Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, A simple baseline for multi-object tracking, 2020 arXiv preprint arXiv:2004.01888.
- [25] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: *Proc. the Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [26] Z. Zhou, J. Xing, M. Zhang, W. Hu, Online multi-target tracking with tensor-based high-order graph matching, in: *Proc. IEEE Int. Conf. Pattern Recognit.*, 2018, pp. 1809–1814.
- [27] X. Wan, J. Wang, Z. Kong, Q. Zhao, S. Deng, Multi-object tracking using online metric learning with long short-term memory, in: *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 788–792.
- [28] M. Babaei, Z. Li, G. Rigoll, A dual CNN-RNN for multiple people tracking, *Neurocomputing* 368 (2019) 69–83.
- [29] R. Sanchez-Matilla, F. Poiesi, A. Cavallaro, Online multi-target tracking with strong and weak detections, in: *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 84–99.
- [30] N. Mahmoudi, S.M. Ahadi, M. Rahmati, Multi-target tracking using CNN-based features: CNNMTT, *Multimedia Tools Appl.* 78 (6) (2019) 7077–7096.
- [31] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. Schindler, Motchallenge 2015: towards a benchmark for multi-target tracking, 2015 arXiv preprint arXiv:1504.01942.
- [32] A. Ess, B. Leibe, K. Schindler, L. Van Gool, A mobile vision system for robust multi-person tracking, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [33] L. Patino, T. Nawaz, T. Cane, J. Ferryman, PETS 2017: Dataset and challenge, in: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 2126–2132.