Pattern Recognition 107 (2020) 107480

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

TPM: Multiple object tracking with tracklet-plane matching

Jinlong Peng[a,1], Tao Wang[a,1], Weiyao Lin[a,*], Jian Wang[b], John See[c], Shilei Wen[b], Erui Ding[b,2]

[a] School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, China
[b] Department of Computer Vision Technology (VIS), Baidu Inc. China
[c] Faculty of Computing and Informatics, Multimedia University, Malaysia

ARTICLE INFO

Article history:
Received 30 December 2019
Revised 23 March 2020
Accepted 28 May 2020
Available online 30 May 2020

Keywords:
Multiple object tracking
Tracklet
Tracklet-plane
Representative-selection network

ABSTRACT

Multiple object tracking (MOT) aims to model the temporal relationship among detected objects and associate them into trajectories. Thus, one major challenge of MOT lies in the confusion from noisy object detection results. In this paper, we propose Tracklet-Plane Matching (TPM), a new approach which improves the performance of MOT by modeling and reducing the interferences from noisy or confusing object detections. TPM first constructs good temporally-related object detections into short tracklets. Then, a tracklet-plane matching process is introduced to organize related tracklets into planes and associate them into long trajectories. The tracklet-plane matching process assigns visually confusing tracklets into different tracklet planes according to their contextual information, thus properly reducing the confusion among similar tracklets. At the same time, it also allows association among temporally non-neighboring or overlapping tracklets, which provides good flexibility to handle confusion from noisy detections. Under this process, a tracklet-importance evaluation scheme and a representative-based similarity modeling scheme are introduced. These two schemes can properly evaluate the reliability of detection results and identify reliable ones during association so that the impact of noisy or confusing detections can be well-mitigated. Experimental results on benchmark datasets demonstrate that the proposed approach outperforms the state-of-the-art MOT methods.

1. Introduction

Multiple object tracking (MOT) is of increasing importance in many applications including intelligent video surveillance, automatic driving and robotics. After obtaining the results of each frame from the detector, the MOT task aims to model the temporal relationship among detected objects and associate them into trajectories [1]. Since the major target of MOT is to find the correspondences and perform matching among multiple objects in neighboring frames, this remains as a fundamental matching problem in various visual applications.

Basically, since object detection results are the major information cues for MOT, noisy detection results will hurt the performance of MOT [2]; a pictorial example of such an issue is shown in Fig. 1. Thus, one major challenge of MOT lies in properly handling these noisy detections. Most of the existing MOT approaches focused on developing a proper object association strategy such that objects in different frames are optimally matched under some cost functions. However, since different objects in a video may be confused due to their similarity in appearance or motion, the association results are often interfered with by these confusing detections. Some researchers aim to reduce this interference by developing more differentiable feature representations or similarity metrics, or learning them together. These methods still have limitations when handling highly confusing objects. At the same time, since noisy detections are inevitable in the association process, their results are also interfered with by these noisy detections. Some recent approaches aim to reduce the confusion from noisy detections by introducing more accurate object detectors, or developing more reliable object-wise similarity metrics. However, they do not discriminate good detections from noisy ones very well, which in turn, impacts their performance when handling visually similar or easily confusable noisy detections.

We posit that handling confusing or noisy detections is important in MOT. For example, in Fig. 1, the confusing detections (orange box and green box) are visually similar, hence they may easily result in wrong association results. However, if we can find some

* Corresponding author.
E-mail address: wylin@sjtu.edu.cn (W. Lin).
[1] Equal contribution.
[2] This work is supported in part by China Major Project for New Generation of AI Grant (No. 2018AAA0100400), National Natural Science Foundation of China (No. 61971277), and Baidu Research Grants.

https://doi.org/10.1016/j.patcog.2020.107480
0031-3203/

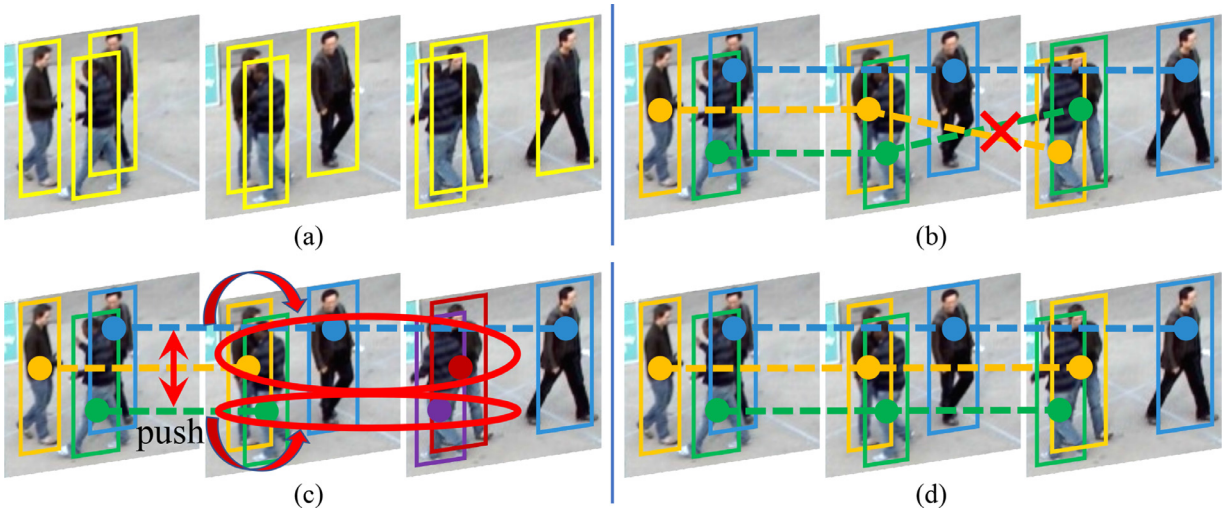

Fig. 1. Example of confusing object detections. (a) The detection result. (b) The wrong tracking result due to overlapping and confusing object detections (orange box and green box, respectively). (c) Our TPM approach separates the confusing objects into different tracklet-planes by using contextual information. (d) The tracking result of our TPM approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

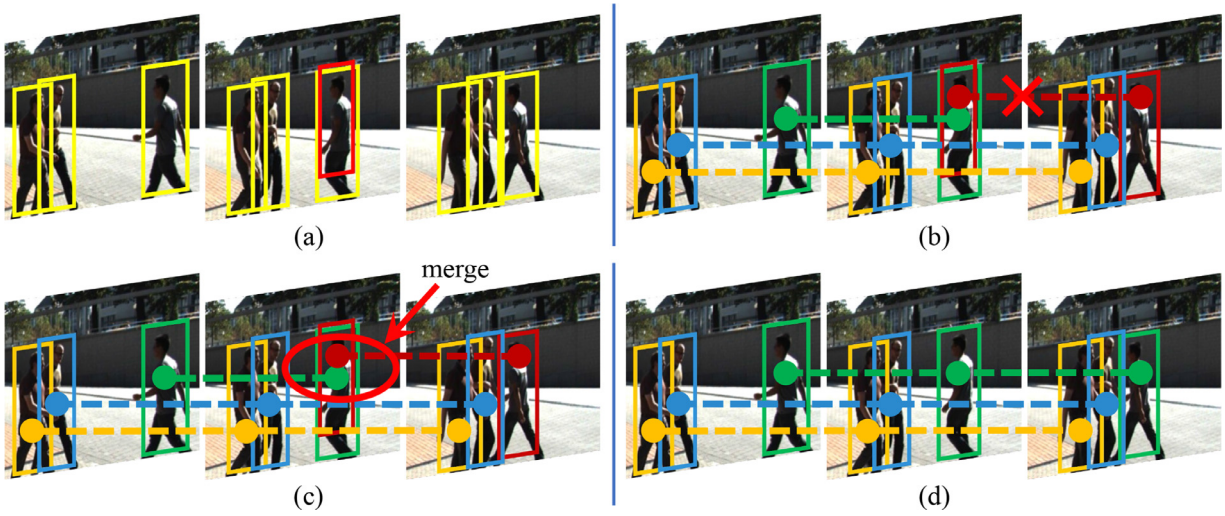

Fig. 2. Example of noisy object detections. (a) The detection result. The red box is the noisy detection. (b) The wrong tracking result is due to noisy detection, which in turn generates a redundant trajectory. (c) Our TPM approach merges the overlapping tracklets of the same object by tracklet-plane. (d) The tracking result by our TPM approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

way to reduce such confusion (e.g., utilizing the contextual information), the association processing can be facilitated, leading to a more reliable tracking result. Furthermore, as in Fig. 2, since noisy detections (red box) include duplicated or misleading information, wrong tracking results are present when utilizing them in the detection matching or association process. However, if we can discriminate and exclude these noisy detections, the tracking results can also be improved.

To this end, we propose a new approach named Tracklet-Plane Matching (TPM) for multiple object tracking, whose framework is shown in Fig. 3. We propose a new concept called **tracklet-plane** (See Fig. 3). A tracklet-plane is a spatial-temporal hyperplane, where tracklets are connected to it by assigning the start and end detections of tracklets to one side of the plane, such that tracklets connected to the same tracklet-plane have a higher probability of being associated to form a long trajectory. The proposed TPM approach first associates the object detections with high similarity into short tracklets. Then, we design an in-plane matching process to organize related short tracklets into their planes and associate these short tracklets on the same plane to generate long trajectories. The tracklet-plane matching process assigns visually confusing tracklets into different tracklet planes according to their contextual information, thus reducing the confusion among similar tracklets. At the same time, it also associates the temporally non-neighboring and overlapping tracklets effectively, which complements missing detections and exclude noisy detections. To improve the performance of this tracklet-plane-based process, we further introduce a tracklet-importance evaluation scheme and a representative-based similarity modeling scheme. These schemes can evaluate the reliability of tracklets and pick up reliable ones during association. Thus overall, the impact of confusing or noisy detections can be further reduced. Extensive ablation studies and comparisons with the state-of-the-art methods are conducted on MOT16 and MOT17 benchmarks, highlighting the promise of TPM.

In summary, the contributions of our approach are three folds: (1) We propose a tracklet-plane matching process, which constructs tracklet-planes to differentiate the easily confusable tracklets and model the association among temporally non-neighboring or overlapping tracklets, and thus providing good flexibility to handle the interferences from confusing or noisy detections. (2) We

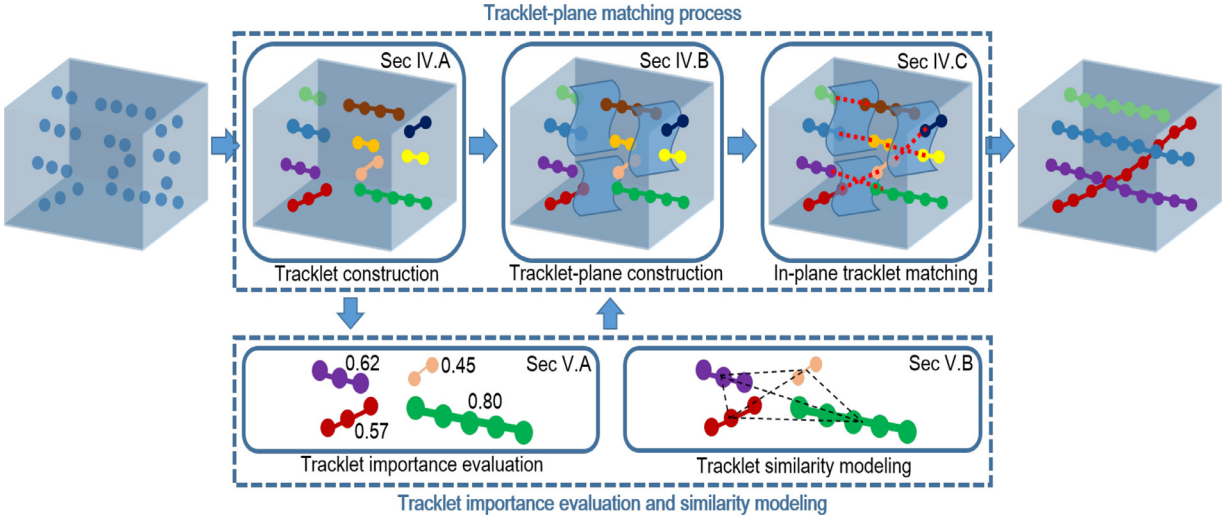

Fig. 3. Framework of the proposed TPM approach.

introduce a tracklet-importance evaluation scheme that measures the reliability of tracklets and excludes the noisy detections. (3) We introduce a representative-based similarity modeling scheme by proposing a deep-based representative-selection network, which can effectively discriminate and pick up reliable and representative detections when calculating tracklet-wise similarity, thus further eliminating the interferences from confusing or noisy detection.

The rest of the paper is organized as follows. Section 2 reviews the related work while Section 3 describes the overall framework of TPM. The details of our proposed tracklet-plane matching process and tracklet importance / similarity modeling are described in Section 4 and Section 5 respectively. Section 6 then presents the experimental results. Finally, Section 7 concludes the paper.

2. Related work

Most existing MOT approaches focus on improving its performance from three aspects: object association, object detection, and object-wise similarity measure.

Object association. Since the key task of multiple object tracking is to associate detected objects, most current research focused on developing proper object association strategies. Basically, object association can be viewed as a fundamental matching problem in the temporal domain, which aims to model the relationship, find the correspondence, and perform matching among detected objects in temporally neighboring frames. Therefore, many researchers have developed matching models or matching theories to address the object association problem.

Tang *et al.* [3] designed a Subgraph Multicut model to deal with the detection association problem and solved it by the Kernighan-Lin algorithm [4].

Ren *et al.* [5] applied a prediction network and a decision network to associate the objects by collaborative deep reinforcement learning. Maksai *et al.* [6] proposed a non-Markovian MOT approach by using behavioral patterns to impose global consistency. Yan *et al.* [7] proposed an affinity optimization method with graduated consistency regularization to improve the accuracy of graph matching. Since these methods include both noisy and easily confusable detections in the association process, their results are often interfered with by these misleading detections. Our proposed TPM can assign the visually similar or easily confusable objects into different tracklet-planes so that they are less associated incorrectly, which is effective in improving the overall tracking performance.

Object detection. Since object detection results play an important role in MOT, some research works also aim to improve object detection capabilities for better MOT accuracy. Henschel *et al.* [8] used a multi-detector to track pedestrians by fusing body and head detections. Furthermore, Chu *et al.* [9] used single object tracking (SOT) to enrich detections in MOT. Kim *et al.* [10] modeled the multi-object state as a labeled random finite set and used Bayes recursion to eliminate false negatives and false positives. Jorquera *et al.* [11] introduced probability hypothesis density filter to avoid data association uncertainty, noise and false alarms.

Object-wise similarity measure. Moreover, some other prior works further develop better object-wise similarity measure to boost the object association results. Some approaches used learning-based methods to calculate pairwise association costs. Tang *et al.* [12] proposed a deep network flow method and a deep matching algorithm to calculate the similarity of objects. In addition, Son *et al.* [13] designed a Quadruplet Convolutional Neural Network to achieve end-to-end tracking which differentiates similar objects by learning. However, some noisy information will be included in the feature if there are serious occlusions in the tracklet. Other research works designed some methods to extract more robust appearance features or motion features, which are also used for obtaining a better object-wise similarity measure. Wang *et al.* [14] proposed a joint learning method for features and distance metrics to distinguish confusing objects. Wu *et al.* [2] designed instance-aware representations to distinguish similar objects. Zhu *et al.* [15] used the estimated trajectory information of future frames to enable more accurate matching between tracklets. Shen *et al.* [16] proposed a patch-based appearance model and spatialâtemporal similarity measurement to increase matching accuracy. Meanwhile, a more recent method by Tian *et al.* [17] introduced a spatial-temporal attention appearance model to solve variations relating to occlusion and illumination, and to calculate a reliable similarity score between the candidate detection and the object. Although these methods can improve the performance of MOT, they still have limitations when handling visually similar or easily confusable noisy detections.

There are also some video re-identification methods [18,19] can be applied to calculate tracklet-wise similarity, but they need to include all information in a video clip to handle large camera-wise variations. This may unfeasibly incorporate noise into the similarity calculation. Comparatively, based on the observation that neighboring tracklets of the same object have small variations, our method focuses on calculating the feature vector for the most representa-

tive detection in every tracklet, which is simpler and more efficient.

Overall, our approach improves MOT from the aspect of object association and object similarity measure. Firstly, we develop a tracklet-plane matching process to differentiate easily confusable tracklets and flexibly model the correlation among tracklets, thus improving the accuracy of object association. Secondly, we introduce a tracklet-importance evaluation scheme together with a representative-based similarity modeling scheme, which discriminates and selects reliable tracklets for a more precise similarity measure.

3. Overview

Fig. 3 illustrates the framework of our approach. As shown in Fig. 3, we first associate initial detections (blue dots) into tracklets in the *tracklet construction* module (Section 4.1). To remove noisy tracklets in these short tracklets, the importance of tracklets are measured in the *tracklet importance evaluation* (Section 5.1) module by confidence of detection results and similarity between adjacent objects. Those tracklets with low importance will be deleted. Since similarity among tracklets will be used in tracklet-plane construction and in-plane tracklet matching, a representative-select network and tracklet similarity metric are designed in the *tracklet similarity modeling* (Section 5.2) module. Based on the evaluated tracklet importance and similarity information, the *tracklet-plane construction* module builds a set of tracklet-planes, in which each tracklet-plane is connected to several tracklets (Section 4.2). Finally, in Section 4.3, the *in-plane tracklet matching* module merges or associates tracklets within each tracklet-plane, and obtains the final long trajectories. Note that our approach uses tracklets as the basic association unit. This way, richer information can be obtained when evaluating object reliability and modeling object-wise similarity.

In contrast to existing work that also employed tracklets to perform object tracking, our approach differs in two major aspects.

(1) Most of the existing tracklet-based methods only view tracklets as an enlarged version of objects intended for the association process, whereby the complex correlations among tracklets (*e.g*, temporally overlapping or non-neighboring) are not well modeled or studied. Comparatively, our approach introduces a tracklet-plane matching process to model tracklet-wise correlations, thus this makes full use of the rich information of tracklets and is capable of obtaining better tracking results.
(2) The existing tracklet-based methods simplistically select the tracklets' terminal objects or extract their global features to evaluate tracklet-wise similarity, which may be easily affected by the interferences of noisy detections. Comparatively, our approach introduces a representative-based similarity scheme, which reduces the effect of noisy object detections by finding the most reliable and representative detections for measuring tracklet-wise similarity.

4. Tracklet-plane matching process

The proposed tracklet-plane matching process aims to associate detections of high similarity into short tracklets, group these highly-related short tracklets into tracklet planes, and further associate these in-plane tracklets into long trajectories. In short, it contains three main steps: tracklet construction, tracklet-plane construction and in-plane tracklet matching. To accurately discriminate and exclude noisy detections / tracklets in the tracklet-plane matching process, it is important to find effective ways to evaluate tracklet reliability and model tracklet-wise similarity. To this end, we also propose a tracklet-importance evaluation scheme and a representative-based similarity modeling scheme.

4.1. Tracklet construction

The tracklet construction process merges highly-related objects into tracklets, which will be used as the basic units in the association process later. In this paper, we first use a min-max normalization process to evaluate the confidence of each detected object; those of low confidence are excluded. Then, we apply Kuhn-Munkres (KM) algorithm [20] to dynamically associate temporally related objects into short tracklets [14]. During object association, we model the similarity between an object D in frame t and a tracklet T constructed in the previous frame $t-1$ as:

$$S_{to}(T,D) = A(T,D) + \lambda_s M(T,D), \tag{1}$$

where $S_{to}(T, D)$ represents the similarity between tracklet T and object D. $A(T, D)$ is the appearance similarity and $M(T, D)$ is the motion similarity. $\lambda_s = 0.5$ is the weight balancing the importance between the appearance similarity and the motion similarity. We compute the cosine similarity between *pool5* features of ResNet-50 [21] as the appearance similarity, while the velocity and position information of the tracklet and the object is computed to obtain the motion similarity [22].

4.2. Tracklet-plane construction

After obtaining short tracklets, we need to further associate them to form long trajectories. However, due to the interferences of confusing or noisy detections, tracklets belonging to the same trajectory may become temporally disconnected, temporally overlapping, or created by noisy detections (Fig. 5b). Directly applying association methods may lead to low performances.

To address the aforementioned interferences, we develop a tracklet-plane matching method to organize related tracklets into planes and apply association methods in each tracklet plane. This resolves the association confusions caused by noisy or missing detections. In our method, the optimization function that constructs the tracklet-planes is given by:

$$(X^*, Y^*, n_p^*) = \arg\min_{X,Y,n_p} \Phi_1(X,Y,n_p) + \Phi_2(X,Y,n_p) + \lambda_p n_p,$$
$$s.t. \quad x_i^m, y_j^m \in \{0,1\}; \sum_{m=1}^{n_p} x_i^m \le 1; \sum_{m=1}^{n_p} y_j^m \le 1 \tag{2}$$

where $X = \{x_i^m\}, i = 1,\dots,n_t, m = 1,\dots,n_p$ and $Y = \{y_j^m\}, j = 1,\dots,n_t, m = 1,\dots,n_p$ are the sets representing the tracklet-plane construction status of all tracklets in a video, $x_i^m = 1$ indicates the end of tracklet t_i that is connected to tracklet-plane P_m and $y_j^m = 1$ indicates the start of tracklet t_j that is connected to tracklet-plane P_m (See Fig. 4). n_t is the total number of tracklets and n_p is the total number of tracklet-planes. $\lambda_p = -0.1$ is the balancing weight. The constraints guarantee that the start and the end of every tracklet are connected to at most one tracklet-plane. $\Phi_1(X, Y, n_p)$ and $\Phi_2(X, Y, n_p)$ are the optimization terms for evaluating tracklet-plane construction qualities, which are defined in Eq. 3 and Eq. 4, respectively:

$$\Phi_1(X,Y,n_p) = -\sum_{m=1}^{n_p}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} 2x_i^m y_j^m W_i W_j S_{tt}(T_i,T_j), \tag{3}$$

$$\Phi_2(X,Y,n_p) = \sum_{m=1}^{n_p}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} (x_i^m x_j^m + y_i^m y_j^m) W_i W_j S_{tt}(T_i,T_j), \tag{4}$$

where W_i represents the importance of tracklet T_i, which is evaluated by the tracklet importance evaluation scheme (Section 5.1).

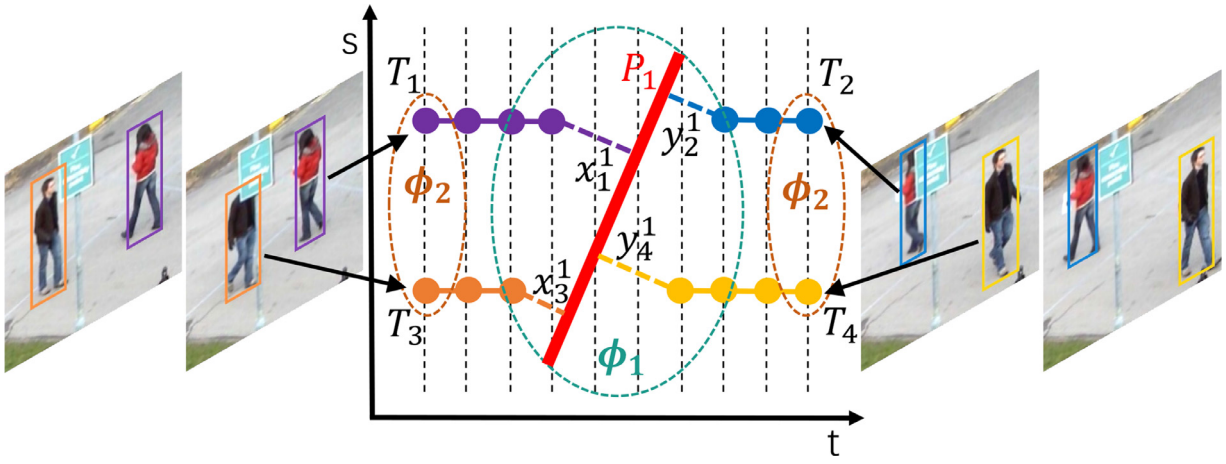

Fig. 4. Schematic diagram with symbols. t axis represents time and s axis represents space.

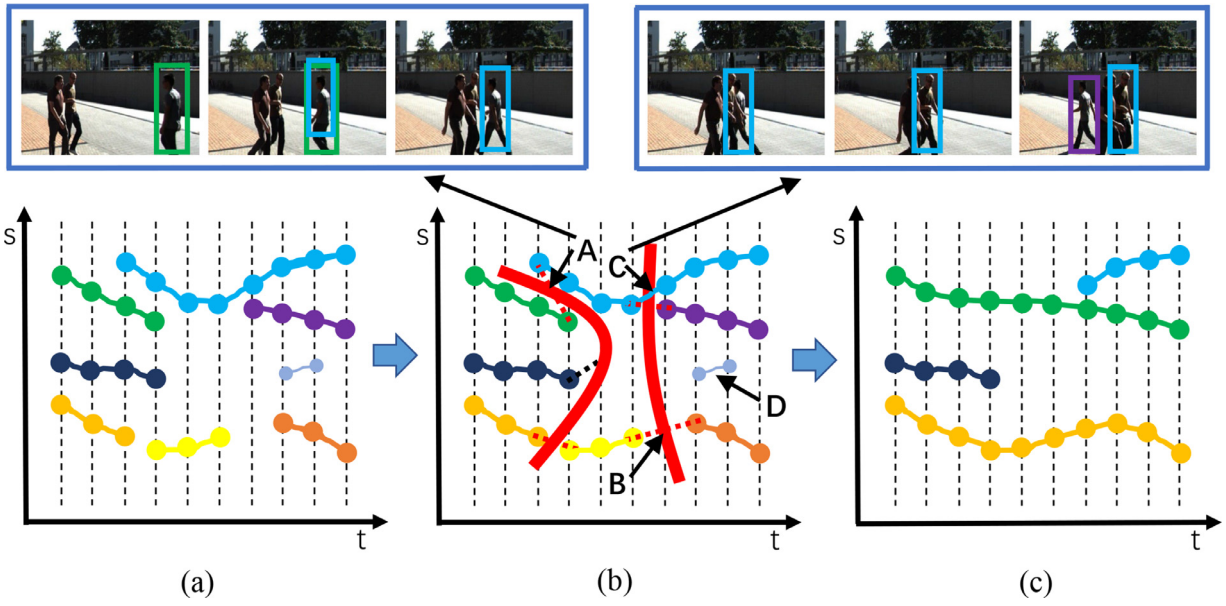

Fig. 5. Tracklet-plane matching process: (a) Short tracklets. (b) Tracklet-plane construction results. A shows temporally overlapping tracklets due to duplicated detections. B shows temporally non-neighboring tracklets due to missing detections. C indicates association errors in tracklets. D shows less reliable tracklets. (c) Trajectories generated from (b).

$S_{tt}(T_i, T_j)$ denotes the similarity between tracklet T_i and tracklet T_j, which is calculated by the representative-based similarity modeling scheme (Section 5.2).

According to Eqs. 2–4, $\Phi_1(X, Y, n_p)$ mainly measures the total similarity among tracklets connected to different sides of a tracklet-plane (Fig. 4). Note that there is a negative sign in Eq. 3. The second term $\Phi_2(X, Y, n_p)$ measures the similarity among tracklets connected to the same side of a tracklet-plane (Fig. 4).

Furthermore, we want tracklets that have the potential of belonging to the same trajectory to be connected to different sides of a tracklet-plane, such that they can be associated during the in-plane tracklet matching step. At the same time, we want the visually similar or easily confusable tracklets to be not connected to the same side of a tracklet-plane, such that they will not interfere with each other in the in-plane tracklet matching step. Thus, by jointly optimizing $\Phi_1(X, Y, n_p)$ and $\Phi_2(X, Y, n_p)$ in Eq. 2, tracklets can be properly organized according to our requirements.

Moreover, we introduce a third term $\lambda_p n_p$ in Eq. 2 to encourage tracklets being organized into more tracklet-planes. In this way, we can reduce the number of tracklets in each tracklet-plane and ease the later in-plane tracklet matching step.

Additionally, since some tracklets may wrongly include the sub-tracklets of two objects when constructing tracklet-planes, we also allow a tracklet to be split into two parts at the point where its two adjacent object pairs have the minimum similarity within that tracklet. And the split tracklets can be then connected to different sides of a tracklet plane. This way, the association errors in tracklets can also be corrected by the tracklet-plane matching process. Fig. 5(b) and Fig. 5(c) show some examples of tracklet-plane construction results and final trajectory results. From Eqs. 2–4 and Fig. 5–6, we can observe the advantages of our tracklet-plane matching process as follows:

(1) Our approach can effectively differentiate and assign visually confusing tracklets into different tracklet-planes. Thus, the interferences among confusing tracklets can be effectively reduced in the tracking results. For example, in Fig. 6b, tracklet C has high visual similarity with a confusing tracklet D. If performing direct association, C may be incorrectly associated to D instead of its correct ground-truth tracklet F. However, by performing our tracklet plane construction process, tracklet D will be 'pushed' towards a tracklet plane that is different from C based on its contextual constraints with

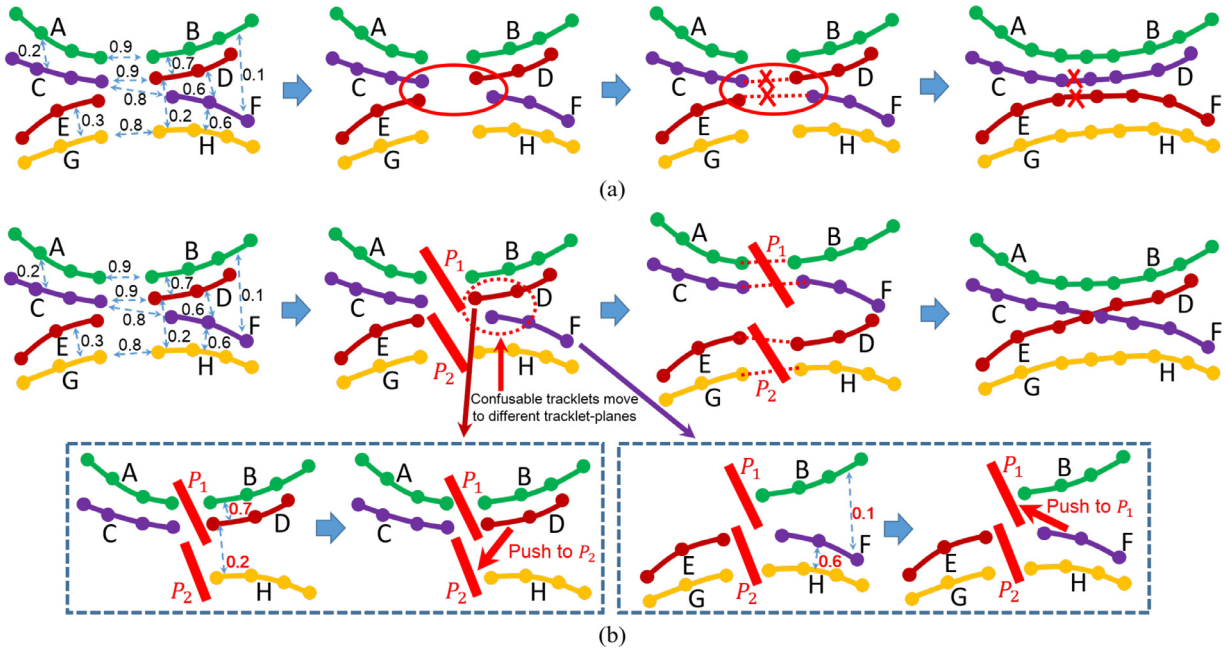

Fig. 6. (a) Traditional tracklet cluster methods. The easily confusable tracklets (the red tracklet and the purple tracklet) which are similar will always be clustered into the same cluster, which is hard to be distinguished in tracklet matching step. (b) Our TPM approach. Tracklet-plane matching could push the easily confusable tracklets into different tracklet-planes, which reduces the confusion in tracklet matching step. In detail, the high similarity between B and D generates a high value of Φ_2, the optimization of tracklet-plane construction pushes D to P_2 from P_1. Similarly, F is pushed to P_1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

other tracklets. In other words, although the similarity between D and C are high, their similarity to other tracklets are different. This difference will push C and D into different tracklet planes according to the constraints Φ_2 in Eq. 2 (e.g. based on left side of Fig. 6(b): when tracklets A and B are assigned to the two sides in a tracklet plane P_1, tracklet C can be assigned to the left side of P_1 since it has large dissimilarity to A. However, tracklet D cannot be assign to the right side of P_1 since it has small dissimilarity to B). Thus, the confusion between C and D can be properly avoided. Similarly, on the right side example, tracklet F is pushed to P_1 and the confusion between E and F is also avoided.

Moreover, it should be noted that although some tracking methods also use contextual information [3], our approach is essentially different from theirs. Generally, since these existing works aim to use contextual information to group visually similar objects or tracklets, the intrinsic confusion among similar objects/tracklets still cannot be avoided inside the groups. Comparatively, our tracklet-plane approach aims to use the contextual information to 'push' visually similar tracklets into *different* tracklet planes. Thus, the confusion among similar tracklets can be effectively avoided.

(2) Since our tracklet-plane construction module allows tracklets at different temporal locations to be linked into the same tracklet-plane, it has large flexibility to handle the association of tracklets with different kinds of issues, *i.e*, temporally non-neighboring tracklets due to missing detections (*i.e*, B in Fig. 5(b)) or temporally overlapping tracklets due to duplicated detections (*i.e*, A in Fig. 5b).

(3) Our tracklet-plane construction module also allows tracklets to be split and connected to different sides of a tracklet-plane. This enhances the capability of our tracklet-plane matching process to correct potential association errors that may occur (*i.e*, C in Fig. 5b).

(4) Our tracklet-plane construction module also integrates the importance weights of tracklets (*i.e*, W_i and W_j in Eqs. 3 and 4). This way, we also have the flexibility to evaluate the reliability of tracklets and to exclude less reliable tracklets in the tracklet-association process (*i.e*, D in Fig. 5b, not linking it to any tracklet-planes).

Inference. Solving Eq. 2 is not trivial as the optimization terms and constraints are discrete and complicated. In this paper, we use Local Gradient Descent algorithm [23] to obtain an approximate solution that iteratively derives candidate tracklet-plane solutions by sequentially connecting tracklets to neighboring planes to find the best one.

We set S as the matrix combined with tracklet similarity and tracklet importance:

$$S_{ij} = W_i W_j S_{tt}(T_i, T_j), \forall i, j \in [1, n_t], \tag{5}$$

then $\Phi_1(X, Y, n_p)$ and $\Phi_2(X, Y, n_p)$ can be expressed as:

$$\Phi_1(X, Y, n_p) = -2 \sum_{m=1}^{n_p} X_m^T S Y_m, \tag{6}$$

$$\Phi_2(X, Y, n_p) = \sum_{m=1}^{n_p} (X_m^T S X_m + Y_m^T S Y_m), \tag{7}$$

where X_m is the mth column of matrix X and Y_m is the mth column of matrix Y. Due to that S is a symmetric matrix, the optimization function $\Phi(X, Y, n_p)$ can be expressed as:

$$\Phi(X, Y, n_p) = \sum_{m=1}^{n_p} (X_m - Y_m)^T S (X_m - Y_m) + \lambda_p n_p, \tag{8}$$

let $A = X - Y$, then:

$$\Phi(A, n_p) = \sum_{m=1}^{n_p} A_m^T S A_m + \lambda_p n_p, \tag{9}$$

the function in Eq. 9 can be made convex by the normalized Laplacian matrix of S:

$$\widehat{S} = I - G^{\frac{1}{2}} S G^{\frac{1}{2}}, \tag{10}$$

where I is the identity matrix and G is the diagonal matrix by the row sums of S. Therefore, $\Phi(X, Y, n_p)$ is convex for both X and Y and we can apply the Local Gradient Descent algorithm to solve Eq. 2. The detailed process is described in the following steps as well as Algorithm 1.

Algorithm 1 Algorithm to find the tracklet-planes (optimal solution of Eq. 2).

Require: $W_i, W_j, S_{tt}(T_i, T_j)\ \forall i = 1, \ldots, n_t\ j = 1, \ldots, n_t$
Ensure: X^*, Y^*, n_p^*
1: Initialize n_p by Eq. 11
2: Randomly initialize X and Y with constraints in Eq. 2
3: Calculate $\Phi(X, Y, n_p)$ by Eq. 8
4: **while** $\Phi(X', Y', n_p') < \Phi(X, Y, n_p)$ **do**
5: $X = X'$, $Y = Y'$
6: Calculate $\Phi(X', Y', n_p')$ by Eq. 8, 12–16
7: **end while**
8: $X^* = X$, $Y^* = Y$, $n_p^* = n_p$

(1) We first set the initial number of tracklet-planes as $\sqrt{n_t}$:

$$n_p = \sqrt{n_t}, \tag{11}$$

which means the columns of X and Y is $\sqrt{n_t}$ and the rows of X and Y is n_t.

(2) Randomly assign all the tracklets to tracklet-planes with the constraints in Eq. 2, which means that we randomly set one element as 1 and other elements as 0 for every row of X and Y.

(3) Then, we apply the Local Gradient Descent algorithm and the normalization operation to update X, Y and n_p. The partial derivative of $\Phi(X, Y, n_p)$ to X and Y is given in Eq. 12 and Eq. 13:

$$\frac{\partial \Phi(X, Y, n_p)}{\partial X} = 2S(X - Y), \tag{12}$$

$$\frac{\partial \Phi(X, Y, n_p)}{\partial Y} = 2S(Y - X), \tag{13}$$

then we can update X and Y by Eq. 14 and Eq. 15:

$$X' = \mathcal{N}(X - \frac{\partial \Phi(X, Y, n_p)}{\partial X}), \tag{14}$$

$$Y' = \mathcal{N}(Y - \frac{\partial \Phi(X, Y, n_p)}{\partial Y}), \tag{15}$$

where $\mathcal{N}(X)$ is the normalization operation, which sets the maximum element of every row of X as 1 and the other elements as 0. In this process, there are some columns of X and Y whose elements are all 0, which means that the tracklet-plane can be removed. Therefore, the actual number of tracklet-planes n_p is the initial value minus the union of all-zero columns of X and Y, as in Eq. 16:

$$n_p' = \sqrt{n_t} - (\odot(X) \cup \odot(Y)), \tag{16}$$

where $\odot(X)$ is the number of all-zero columns of X.

(4) After every iteration we calculate the objective function value in Eq. 2 by X, Y and n_p. If the value does not decrease, the last X, Y and n_p can be seen as the optimal solution of Eq. 2.

4.3. In-plane tracklet matching

After connecting related tracklets onto tracklet-planes, we are able to perform tracklet-wise association (*i.e*, matching) within each tracklet-plane to obtain final trajectories. Particularly, the in-plane tracklet matching process aims to find the best one-to-one matching among tracklets which are connected to different sides of a tracklet-plane. Therefore, the process can be modeled as:

$$Z^* = \arg\max_Z \sum_{m=1}^{n_p} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} x_i^m y_j^m z_{ij} W_i W_j S_{tt}(T_i, T_j),$$

$$s.t. \quad z_{ij} \in \{0, 1\}; \sum_{j=1}^{n_t} z_{ij} \le 1; \sum_{i=1}^{n_t} z_{ij} \le 1 \tag{17}$$

where $Z = \{z_{ij}\}, i = 1 \ldots n_t, j = 1 \ldots n_t$ is the set representing the tracklet association status. $z_{ij} = 1$ means the end of tracklet T_i is connected to the start of tracklet T_j (*i.e*, T_i and T_j are associated). x_i^m and y_j^m are obtained by Eq. 2, which guarantees that only the tracklets from both sides of the same tracklet-plane can be associated. Finally, KM algorithm [20] is applied to solve Eq. 17.

Note that since the associated tracklets may be temporally overlapping / non-neighboring, or may include noisy detections, some interpolation, merging or deleting operations are further applied on the associated tracklets to obtain clean and coherent trajectories, as shown in Fig. 7.

5. Tracklet importance evaluation and similarity modeling

To accurately discriminate and exclude noisy detections / tracklets in the tracklet-plane matching process (Section 4), it is important to find effective ways to evaluate tracklet reliability and model tracklet-wise similarity. To this end, we propose a tracklet-importance evaluation scheme and a representative-based similarity modeling scheme.

5.1. Tracklet importance evaluation

The importance of tracklet T_i is calculated by:

$$W_i = \frac{\sum_{n=1}^{L_i} C_i^n}{L_i} \frac{\sum_{n=1}^{L_i - 1} S_{oo}(D_i^n, D_i^{n+1})}{L_i - 1} (1 - e^{-\sqrt{L_i}}), \tag{18}$$

where D_i^n denotes the n-th object of tracklet T_i, C_i^n represents the confidence of detection D_i^n similar to those by [24], $S_{oo}(D_i^n, D_i^{n+1})$ represents the appearance similarity between two adjacent objects D_i^n and D_i^{n+1}, and L_i represents the length of tracklet T_i.

In general, good tracklets consist of detections with high confidence (modeled by the first term in Eq. 18) and high appearance similarity (second term). Additionally, the longer the tracklet is, the more important the tracklet becomes (third term). As a result, according to Eq. 18, tracklets with high importance will be regarded as more reliable and will have higher chance to join tracklet-planes and to be associated with other tracklets Eqs. 2 and (17). In this way, the tracklet matching process can be more assured of obtaining results containing highly confident tracklets while excluding unwanted noisy tracklets.

5.2. Representative-based similarity modeling

Tracklet similarity $S_{tt}(T_i, T_j)$ is another key factor affecting the performance of the tracklet matching process. Intuitively, tracklet-wise similarity can be modeled by the similarity between the features of tracklets' terminal objects [1,8] (Fig. 8(a)) or between the global features of tracklets [25] (Fig. 8(b)). However, since tracklets may include noisy detections, directly using terminal object features or global features may improperly introduce noisy information and reduce the reliability of the similarity measure.

Therefore, we propose a representative-based similarity modeling scheme, which introduces a neural network to select the most representative objects from each tracklet and use them to model tracklet similarity. Our proposed representative-based similarity modeling scheme is shown in Fig. 8(c). When calculating the

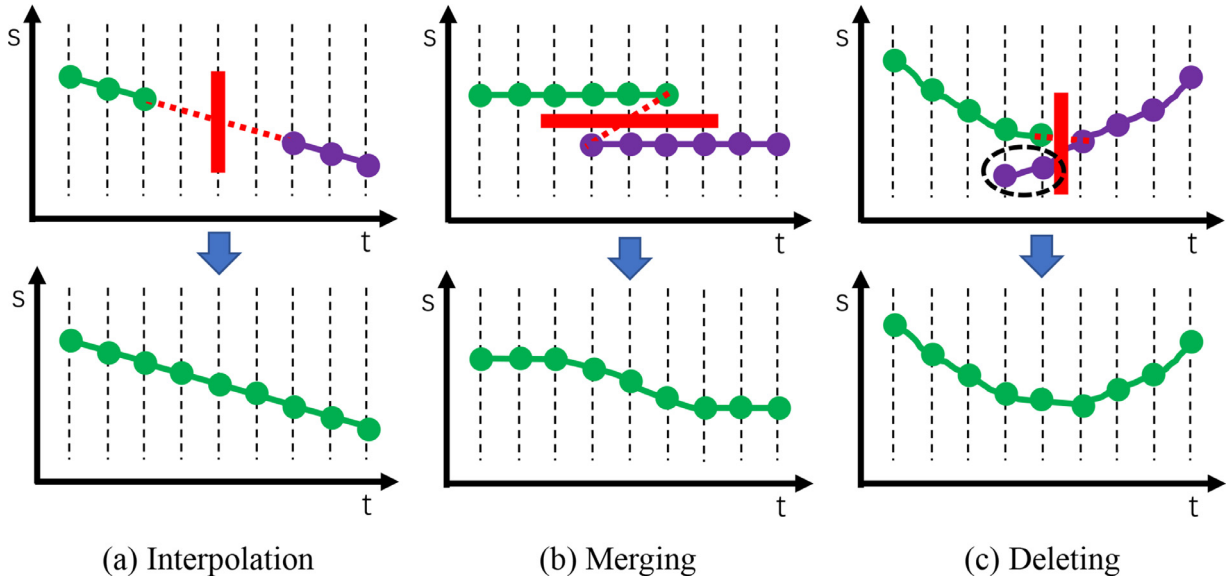

Fig. 7. Three operations of tracklet association: (a) **Interpolation**. The detection boxes will be interpolated into the gap between the two associated tracklets. (b) **Merging**. The overlap detection boxes between the two associated tracklets will be merged by calculating the confidence-based weighted average in the same frame. (c) **Deleting**. The left tracklets (in the dashed circle) with low importance will be deleted.

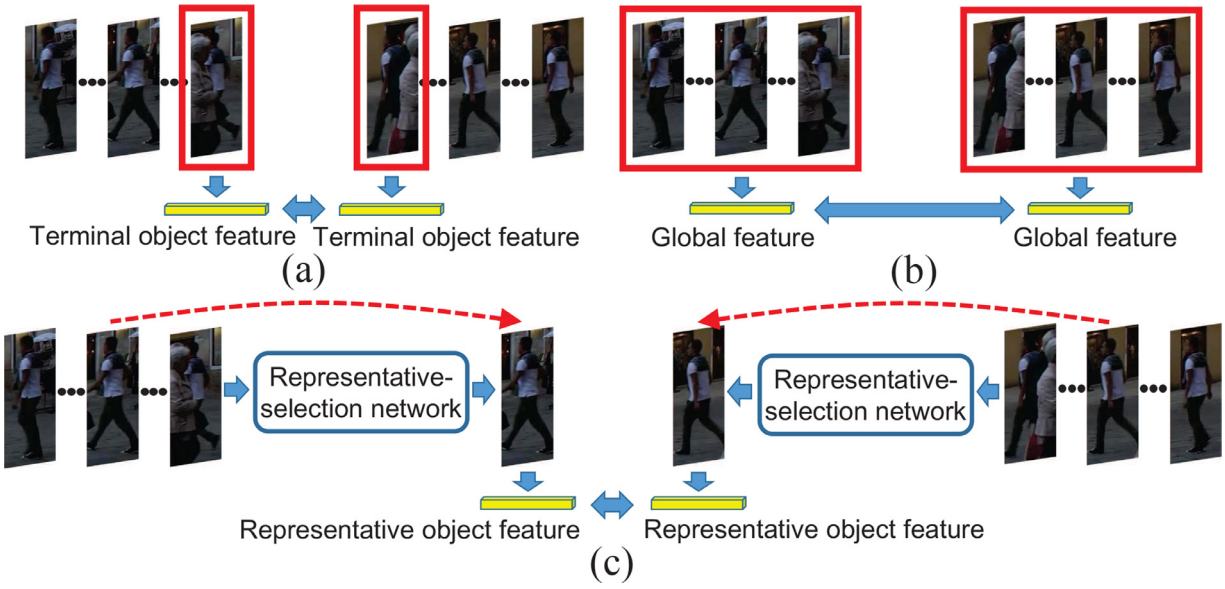

Fig. 8. Three tracklet similarity measurement strategies: (a) By the features of terminal objects. (b) By the global features. (c) By the features of representative objects (Ours).

similarity $S_{tt}(T_i, T_j)$ between two tracklets T_i and T_j, we first input them into a representative-selection network and select the most reliable representative object from each tracklet. Then, the tracklet-wise similarity between T_i and T_j is calculated by the feature similarity between the selected representative objects.

Representative-selection network. In order to capture the temporal variation in a tracklet, we adopt convolutional LSTM [26] as the major structure of the representative-selection network. Moreover, since each tracklet needs to calculate similarity with other tracklets in different temporal locations, we select two representative objects from each tracklet: one is used to measure the similarity with tracklets *before* it, while the other is used to measure the similarity with tracklets *after* it. Therefore, we develop a bi-directional convolutional LSTM as the representative-selection network, as depicted in Fig. 9.

The representative-selection network (as in Fig. 9) contains two LSTM streams, where the forward stream (LSTMs in blue) is designed to select a representative object around the back-end of a tracklet, while the backward stream (LSTMs in yellow) is used to find the representative object around the tracklet's front-end. Each stream has a set of LSTM units where each unit takes an object in a tracklet as input and outputs its representative score. Finally, the object with the highest score will be selected as the representative object and used to calculate tracklet similarity.

Moreover, the forward and backward streams are jointly trained while minimizing the distance between two feature maps of the same detection box. More specifically, a loss function, known as perceptual loss or feature matching loss [27], is adopted to supervise the selection of the two coherent streams. Note that if a tracklet is longer than γ in the inference stage, we will simply ap-

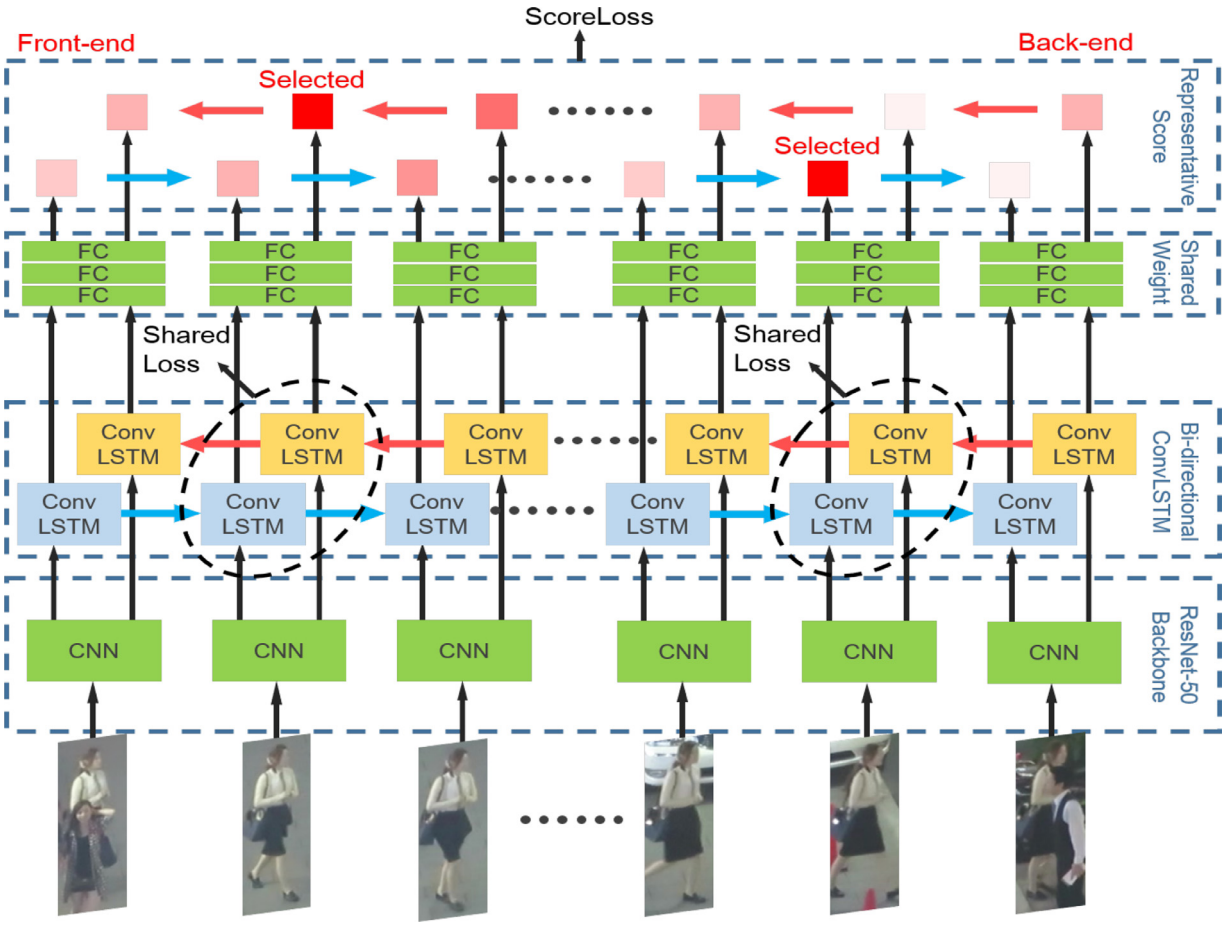

Fig. 9. The representative-selection network. Bi-directional convolutional LSTM units are merged into the representative-selection network. Two score values are calculated from two streams for each object based on the network. The object with the largest score (the brightest red square) is selected for tracklet similarity calculation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ply our network on the first and last γ detections in the tracklet to select representative detections. The value of γ is discussed in Section 6.

Learning and loss function. In order to make the representative-selection network select proper objects, we need to define proper ground-truth representative scores to guide the learning process. In this paper, we define the ground-truth by

$$V_i^t = \frac{1}{\gamma}\sum_{\tau\in G}(S_{oo}(D_i^t, D_i^{t+\tau}) - \max_{j\neq i}(S_{oo}(D_i^t, D_j^{t+\tau}))), \tag{19}$$

where D_i^t is obtained from the training dataset and it denotes the ground-truth detection box of the ith tracklet in frame t. V_i^t is the ground-truth representative score for D_i^t. S_{oo} denotes the appearance similarity between two detections, which is defined in the same way as in Eq. 18. The set G represents the frame clip of the tracklet. For the forward stream, the set $G = \{1, 2, \cdots \gamma\}$, while for the backward stream, $G = \{-1, -2, \cdots - \gamma\}$. An object is more representative if it can differentiate matched objects (same object in different frames as the first term in Eq. 19) from unmatched ones (different objects in different frames as the second term in Eq. 19). Note that since our bi-directional convolutional LSTMs need to select two representative objects from a tracklet, we generate two ground-truth representative scores to guide the forward stream and backward stream. For simplicity, we use the same notation as in Eq. 19 for both streams, yielding two different representative scores V_i^t when given two different sets G corresponding to the two streams.

With the ground-truth representative scores determined by Eq. 19, we define the loss function of the representative-selection network as:

$$\mathcal{L} = \mathcal{L}_F + \mathcal{L}_B + \mathcal{L}_P, \tag{20}$$

$$\mathcal{L}_F = \frac{1}{\gamma}\sum_{\tau=0}^{\gamma}\left\| y_i^{t_B-\gamma+\tau} - V_i^{t_B-\gamma+\tau}\right\|^2, \tag{21}$$

$$\mathcal{L}_B = \frac{1}{\gamma}\sum_{\tau=0}^{\gamma}\left\| y_i^{t_F+\gamma-\tau} - V_i^{t_F+\gamma-\tau}\right\|^2, \tag{22}$$

where $\mathcal{L}_F$ and $\mathcal{L}_B$ are the losses for the forward and backward streams, respectively. $\mathcal{L}_P$ denotes the aforementioned similarity supervision between two feature maps extracted from bi-directional convolutional LSTM units. $y_i^t = f(D_i^t)$ is the prediction score from this network and the function $f(\cdot)$ represents the representative-selection network. $V_i^{t_B-\gamma+\tau}$ and $V_i^{t_F+\gamma-\tau}$ are the ground-truth representative scores for the forward and backward streams, respectively. t_B denotes the back-end frame of tracklet T_i, and t_F denotes the front-end frame of the tracklet. During training, the frame clips are extracted from all tracklets by a sliding window of length γ. At test time, we switched the selections. The clip used for the forward stream is selected from the back-end of the tracklet, while the clip used for the backward stream is selected from the front-end of the tracklet.

Tracklet similarity calculation. After selecting representative objects, the tracklet-wise similarity S_{tt} can be calculated by the

Fig. 10. The blue box represents the selected front-end object obtained by backward stream and the red box represents the selected back-end object obtained by forward stream. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

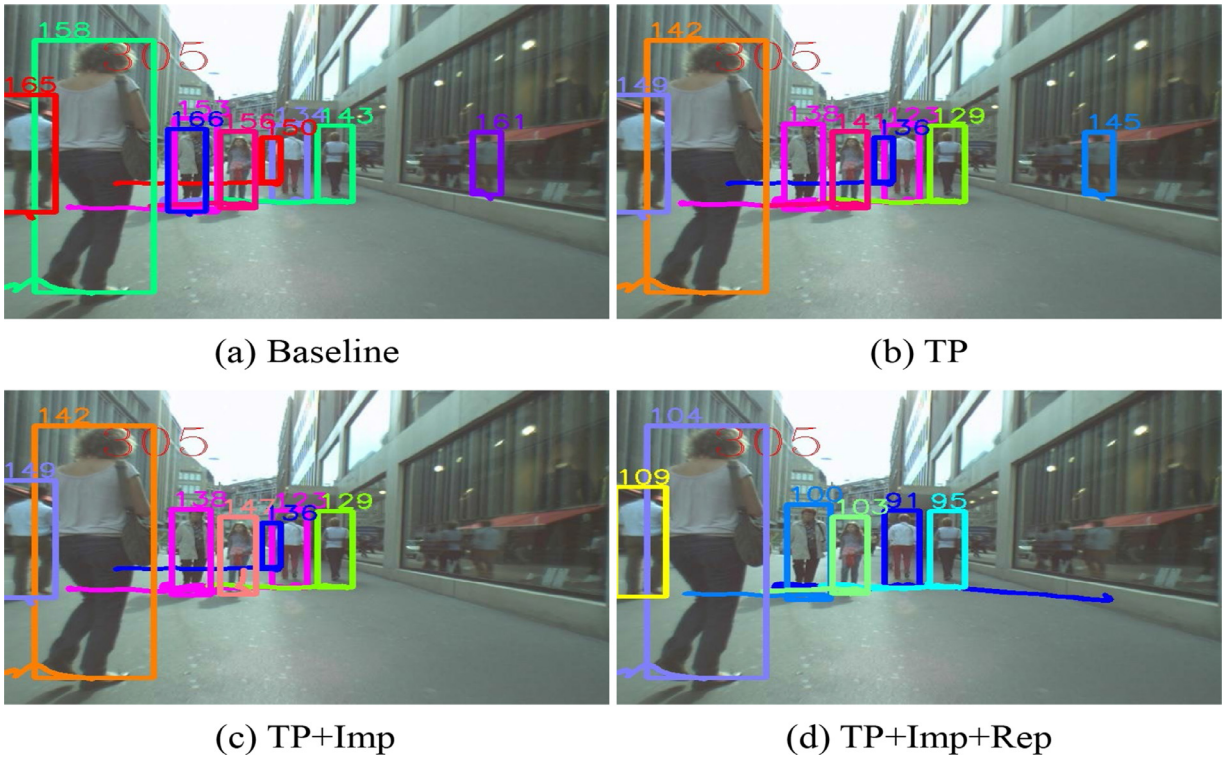

Fig. 11. Qualitative comparisons of different methods on a test sequence (MOT16-06).

combination of the motion similarity and the appearance similarity between the selected representative objects (cf. Eq. 1).

Fig. 10 shows two representative objects selected by our approach. From Fig. 10, we can see that our approach can properly choose reliable and representative objects in tracklets and avoid including noisy objects during tracklet similarity calculation.

6. Experiments

6.1. Datasets and experimental settings

We perform experiments on two benchmark datasets: MOT16 [28] and MOT17.

MOT16 contains 7 training and 7 test sequences. The video sequences in MOT17 are the same as MOT16, except that each sequence is provided with three different detection sets (DPM, Faster-RCNN and SDP), together with newer and more accurate ground-truth. These video sequences are captured by both static and moving cameras, with different scenes and resolutions. Various types of object occlusions and large changes in object appearances render these datasets challenging for MOT research.

Based on the MOTChallenge Benchmark, tracking performance is measured by Multiple object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), the total number of False Negatives (FN), the total number of False Positives (FP), the total number of Identity Switches (IDs), the percentage of Mostly Tracked Trajectories (MT) and the percentage of Mostly Lost Trajectories (ML). Specifically, MOTA measures the overall tracking performance of an approach, together with FN, FP, and IDs.

In our comparisons, we use the public detection results provided by the MOT16 and MOT17 datasets, so that a fair comparison with other MOT methods can be judged. For the representative-selection network in Section 5.2, we use the training sequences of MOT16 as the training data and apply the trained network to perform tracking on all test datasets. While in the ablation study section, we use the training sequences of MOT15 [29] as training data and the training sequences of MOT16 for validation. ResNet-50 [21] pre-trained model is adopted as the backbone of our representative-selection network. SGD optimizer is applied to train the network with batch size of 32 and the initial learning rate is set to 0.0001. After every 20,000 iterations, the learning rate is reduced by half. The training process terminates after 80,000 iterations.

6.2. Ablation study

To evaluate the effectiveness of different components in our approach, we compare the following six methods:

(1) *Baseline*. Directly applying KM algorithm to associate the detected objects into trajectories without constructing tracklets.
(2) *Softassign*. Using KM algorithm to generate reliable tracklets and applying Softassign algorithm [14] to associate the track-

Fig. 12. Some failure cases (highlighted with red circles) when performing tracklet-plane matching on MOT17 test datasets. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

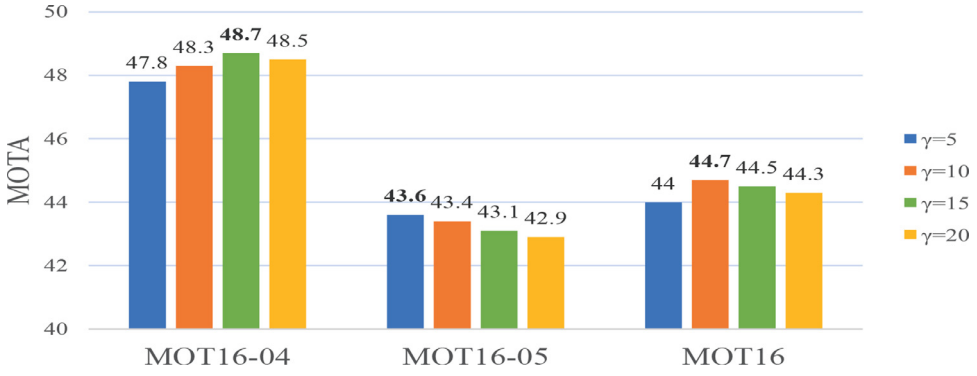

Fig. 13. Tracking performance with different γ values.

Table 1
Ablation study on MOT16 validation dataset.

Method	MOTA↑	MOTP↑	MT↑	ML↓
Baseline	39.6	75.4	14.8%	49.1%
Softassign	40.5	75.4	15.3%	48.2%
TP	42.1	75.4	16.3%	46.7%
TP+Imp	43.0	75.4	16.7%	45.8%
TP+Imp+Glob	43.8	75.5	18.0%	43.4%
TP+Imp+Rep (TPM)	**44.7**	**75.5**	**19.1%**	**40.8%**

Table 2
MOTA results with different hyperparameters.

	0.1	0.3	0.5	0.7	1.0	2.0
λ_s	41.9%	42.6%	44.7%	43.3%	41.7%	40.2%
λ_p	42.3%	44.0%	44.7%	44.2%	43.1%	42.7%

lets into trajectories. The terminal detection in each tracklet is used to calculate tracklet-wise similarity, as shown in Fig. 8(a).

(3) *TP*. Using our tracklet-plane matching process to perform tracking, but excluding both the tracklet-importance evaluation scheme and the representative-based similarity modeling scheme(*i.e*, setting all importance weight W in Eqs. 3–17 to 1 and simply using the terminal detection in each tracklet to calculate tracklet-wise similarity, as shown in Fig. 8(a)).

(4) *TP+Imp*. Using our tracklet-plane matching process to perform tracking, which includes the tracklet-importance evaluation scheme but excludes the representative-based similarity modeling scheme.

(5) *TP+Imp+Glob*. Using our tracklet-plane matching process to perform tracking, including the tracklet-importance evaluation scheme. Global features of tracklets [30] are used to measure tracklet-wise similarity.

(6) *TP+Imp+Rep (TPM)*. Using the full version of our proposed pproach by including both the tracklet-importance evaluation scheme and the representative-based similarity modeling scheme.

Table 1 compares the tracking results on the MOT16 validation dataset and Fig. 11 shows several tracking results of different methods on a sample test sequence (MOT16-06).

From Table 1 and Fig. 11, we can observe that:

TP performs significantly better than *Baseline* and *Softassign*. This indicates that directly performing the association of detected objects or simply using a general tracklet association algorithm can be easily interfered with by noisy detections and similar objects, leading to unsatisfactory results. Comparatively, by applying our tracklet-plane matching process, we can associate the tracklets more reasonably, and have more flexibility to organize tracklets onto tracklet-planes for association even when they are temporally overlapping or non-neighboring. Therefore, we observe stronger capabilities in handling noisy detections such as duplicated (*e.g*trajectory 166 in Fig. 11(a)) or missing detections.

TP+Imp performs better than *TP*. This demonstrates that our tracklet-importance evaluation scheme can effectively evaluate the reliability of tracklets and provide tracking performance improvement by discriminating and excluding the interference of noisy detections and tracklets. For example, our tracklet-importance evaluation scheme identifies the unreliability of tracklet 145 in Fig. 11b, and excludes it from tracklet-plane construction by assigning it with a small importance weight (See Fig. 11(c)). Thus, the incorrect tracking caused by this tracklet can be avoided.

TP+Imp+Rep outperforms *TP+Imp* and *TP+Imp+Glob*, which shows that our proposed representative-based similarity modeling

Table 3
Tracking performance of TPM and state-of-the-art methods on MOT16 and MOT17 test datasets.

Dataset	MOT16						
Method	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓
CDA-DDAL [31]	43.9	74.7	10.7%	44.4%	6450	95,175	676
Quad-CNN [13]	44.1	76.4	14.6%	44.9%	6388	94,775	745
STAM [9]	46.0	74.9	14.6%	43.6%	6895	91,117	473
JMC [12]	46.3	75.7	15.5%	39.7%	6373	90,914	657
NOMT [32]	46.4	76.6	18.3%	41.4%	9753	87,565	**359**
NLLMPa [33]	47.6	78.5	17.0%	40.4%	5844	89,093	629
LMP [34]	48.8	**79.0**	18.2%	40.1%	6654	86,245	481
TPM (Ours)	**50.9**	74.9	**19.4%**	**39.4%**	**4866**	**84022**	619
Dataset	MOT17						
Method	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓
MHT-bLSTM [35]	47.5	**77.5**	18.2%	41.7%	25,981	268,042	2069
PHD-GSDL [36]	48.0	77.2	17.1%	35.6%	23,199	265,954	3998
DMAN [37]	48.2	75.9	19.3%	38.3%	26,218	263,608	2194
EDMT [38]	50.0	77.3	21.6%	36.3%	32,279	247,297	2264
MOTDT [39]	50.9	76.6	17.5%	35.7%	24,069	250,768	2474
JCC [40]	51.2	75.9	20.9%	37.0%	25,937	247,822	**1802**
FWT [8]	51.3	77.0	21.4%	**35.2%**	24,101	247,921	2648
TPM (Ours)	**52.4**	76.6	**22.4%**	40.0%	**19922**	**246183**	2215

Fig. 14. Qualitative results of TPM on different test datasets.

scheme calculates the tracklet-wise similarity more accurately by selecting the most reliable detection in a tracklet. Comparatively, the tracklet terminal-based similarity in *TP+Imp* and the tracklet global feature similarity in *TP+Imp+Glob* are easily affected by noisy or unreliable detections, leading to less satisfactory results. For example, trajectory 123 in Fig. 11c is not complete. By modeling the representative-based similarity, trajectory 91 in Fig. 11d, which corresponds to the same object, is now correctly associated.

Fig. 12 shows some failure cases that commonly occur in TPM. In the case of the first row, trajectory 145 (in the red circle) switches between two neighbouring persons. In the case of the second row, there is another identity switch in trajectory 39 (in the red circle). Basically, both of the aforementioned identity switch failure cases are actually from the errors of the tracklet construction step, *i.e.* the tracklet construction process erroneously put trajectory segments of two persons into a single tracklet. Since the TPM process mainly performs association at the tracklet level, the errors inside tracklets may not be corrected in the final result. One possible solution is to increase the reliability of tracklet construction methods to reduce such kinds of errors. We will explore this issue our future works.

Fig. 13 compares the tracking performance for different values of γ in the representative-selection network. In Fig. 13, MOT16-04 and MOT16-05 are two representative validation sequences, while MOT16 represents all MOT16 validation sequences. We can see that the MOTA score is not that sensitive to the value of γ, which demonstrates the robustness of our proposed approach. From this figure, we see that the best γ for MOT16-04 is 15, while for MOT16-05 is 5. We surmise that the reason here is that MOT16-04 has a higher video frame rate. Based on our observation that the best γ for the entire MOT16 validation dataset is 10, we finally set γ to 10 in all the experiments. Besides, we also study the influence of λ_s and λ_p on the final results, which is shown in Table 2. Empirically, the best numerical value for λ_s and λ_p are both 0.5. From the results, we observe that it is necessary to appropriately decide the weights for appearance feature and motion feature. As for λ_p, if it is set too small a value, there will be too many tracklet-planes generated, which makes merging between tracklet-planes rather challenging. Otherwise, if it is too large, confusing tracklets from different objects are more likely to be placed into the same group, causing mismatches.

6.3. Comparison with state-of-the-art methods

Finally, Table 3 compares the proposed TPM with the state-of-the-art MOT methods on the test sequences of MOT16 and MOT17 datasets. For a fair comparison, all the methods are performed based on the same public detection results.

From Table 3, we make the following observations:

(1) TPM significantly outperforms existing MOT methods in terms of MOTA (the primary metric) on both MOT16 and MOT17, which demonstrates the effectiveness of our approach. Example results of TPM are shown in Fig. 14.

(2) Our TPM approach produces the highest MT and the lowest FN on both MOT16 and MOT17. This shows that our TPM algorithm can associate the tracklets accurately and cater for the missing detections correctly. On the other hand, the MOTP of our approach is slightly lower than some methods because the interpolated detections tend to be ineffective when there are significant camera motions.
(3) On both MOT16 and MOT17, our approach produces the lowest FP among all methods. This demonstrates TPM's advantage of effectively handling the visually-similar or easily confusable objects by discriminating and excluding the less important tracklets and selecting more representative detections for tracklet similarity modeling.

7. Conclusion

This paper introduces a new approach which can alleviate the problem of noisy object detections in MOT by using a new tracklet-plane matching method. To accomplish this, we use existing detection results to construct short tracklets, whereby their individual importance is determined to filter out those of low confidence. The similarity between tracklet pairs are also computed based on a newly designed representative-selection network. The intuition of this is that we can properly evaluate and differentiate the reliability of detection results and select reliable ones during association. We also design a tracklet-plane matching process to put highly-related tracklets into similar planes and likewise, confusing tracklets into different planes to decrease matching errors. Finally, in-plane matching is performed on the associated tracklets with additional post-processing operations to obtain the long, clean trajectories. A standout advantage of our method is that no specially designed feature extraction network or complex matching algorithm is necessary to track good and reliable trajectories. We demonstrate its strengths through extensive experiments on the MOT16 and MOT17 benchmarks. Nevertheless, we also found that sporadic failure cases could happen when short tracklets are not properly constructed. Future research will look into the possibility of using more representative features to reduce such errors when constructing short tracklets.

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.

References

[1] M. Yang, Y. Wu, Y. Jia, A hybrid data association framework for robust online multi-object tracking, IEEE Trans. Image Process. 26 (12) (2017) 5667–5679.
[2] H. Wu, Y. Hu, K. Wang, H. Li, L. Nie, H. Cheng, Instance-aware representation learning and association for online multi-person tracking, Pattern Recognit 94 (2019) 25–34.
[3] S. Tang, B. Andres, M. Andriluka, B. Schiele, Subgraph decomposition for multi-target tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5033–5041.
[4] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, The Bell System Technical Journal 49 (2) (1970) 291–307.
[5] L. Ren, J. Lu, Z. Wang, Q. Tian, J. Zhou, Collaborative deep reinforcement learning for multi-object tracking, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 586–602.
[6] A. Maksai, X. Wang, F. Fleuret, P. Fua, Non-markovian globally consistent multi-object tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2544–2554.
[7] J. Yan, M. Cho, H. Zha, X. Yang, S.M. Chu, Multi-graph matching via affinity optimization with graduated consistency regularization, IEEE Trans Pattern Anal Mach Intell 38 (6) (2015) 1228–1242.
[8] R. Henschel, L. Leal-Taixé, D. Cremers, B. Rosenhahn, Fusion of head and full-body detectors for multi-object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1428–1437.
[9] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, N. Yu, Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4836–4845.
[10] D.Y. Kim, B.-N. Vo, B.-T. Vo, M. Jeon, A labeled random finite set online multi-object tracker for video data, Pattern Recognit 90 (2019) 377–389.
[11] F. Jorquera, S. Hernández, D. Vergara, Probability hypothesis density filter using determinantal point processes for multi object tracking, Comput. Vision Image Understanding 183 (2019) 33–41.
[12] S. Tang, B. Andres, M. Andriluka, B. Schiele, Multi-person tracking by multicut and deep matching, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 100–111.
[13] J. Son, M. Baek, M. Cho, B. Han, Multi-object tracking with quadruplet convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5620–5629.
[14] B. Wang, L. Wang, B. Shuai, Z. Zuo, T. Liu, K. Luk Chan, G. Wang, Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1–8.
[15] S. Zhu, C. Sun, Z. Shi, Multi-target tracking via hierarchical association learning, Neurocomputing 208 (2016) 365–372.
[16] X. Shen, X. Sui, K. Pan, Y. Tao, Adaptive pedestrian tracking via patch-based features and spatial–temporal similarity measurement, Pattern Recognit 53 (2016) 163–173.
[17] S. Tian, L. Zou, C. Fan, L. Chen, Weighted correlation filters guidance with spatial-temporal attention for online multi-object tracking, J. Vis. Commun. Image Represent. 63 (2019) 102576.
[18] J. Wang, S. Zhou, J. Wang, Q. Hou, Deep ranking model by large adaptive margin learning for person re-identification, Pattern Recognit 74 (2018) 241–252.
[19] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, Y. Yang, Improving person re-identification by attribute and identity learning, Pattern Recognit 95 (2019) 151–161.
[20] H.W. Kuhn, The hungarian method for the assignment problem, Naval Research Logistics Quarterly 2 (1–2) (1955) 83–97.
[21] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[22] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, J. Yan, Poi: Multiple object tracking with high performance detection and appearance feature, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 36–42.
[23] S.M. Assari, H. Idrees, M. Shah, Human re-identification in crowd videos using personal, social and environmental constraints, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 119–136.
[24] F. Yang, W. Choi, Y. Lin, Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2129–2137.
[25] S. Li, S. Bak, P. Carr, X. Wang, Diversity regularized spatiotemporal attention for video-based person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 369–378.
[26] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, in: Advances in Neural Information Processing Systems, 2015, pp. 802–810.
[27] A. Dosovitskiy, T. Brox, Generating images with perceptual similarity metrics based on deep networks, in: Advances in Neural Information Processing Systems, 2016, pp. 658–666.
[28] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler, Mot16: a benchmark for multi-object tracking, arXiv Preprint arXiv:1603.00831 (2016).
[29] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. Schindler, Motchallenge 2015: towards a benchmark for multi-target tracking, arXiv Preprint arXiv:1504.01942 (2015).
[30] N. McLaughlin, J. Martinez del Rincon, P. Miller, Recurrent convolutional network for video-based person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1325–1334.
[31] S.-H. Bae, K.-J. Yoon, Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking, IEEE Trans. Pattern Anal Mach Intell 40 (3) (2017) 595–610.
[32] W. Choi, Near-online multi-target tracking with aggregated local flow descriptor, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3029–3037.
[33] E. Levinkov, J. Uhrig, S. Tang, M. Omran, E. Insafutdinov, A. Kirillov, C. Rother, T. Brox, B. Schiele, B. Andres, Joint graph decomposition & node labeling: Problem, algorithms, applications, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6012–6020.
[34] S. Tang, M. Andriluka, B. Andres, B. Schiele, Multiple people tracking by lifted multicut and person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3539–3548.
[35] C. Kim, F. Li, J.M. Rehg, Multi-object tracking with neural gating using bilinear lstm, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 200–215.
[36] Z. Fu, P. Feng, F. Angelini, J. Chambers, S.M. Naqvi, Particle phd filter based multiple human tracking using online group-structured dictionary learning, IEEE Access 6 (2018) 14764–14778.
[37] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, M.-H. Yang, Online multi-object tracking with dual matching attention networks, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 366–382.

[38] J. Chen, H. Sheng, Y. Zhang, Z. Xiong, Enhancing detection model for multiple hypothesis tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 18–27.
[39] L. Chen, H. Ai, Z. Zhuang, C. Shang, Real-time multiple people tracking with deeply learned candidate selection and person re-identification, in: Proceedings of the IEEE International Conference on Multimedia and Expo, 2018, pp. 1–6.
[40] M. Keuper, S. Tang, B. Andres, T. Brox, B. Schiele, Motion segmentation & multiple object tracking by correlation co-clustering, IEEE Trans Pattern Anal Mach Intell 42 (1) (2018) 140–153.

Jinlong Peng received the B.E. and M.E. degrees from Shanghai Jiao Tong University, China, in 2016 and 2019. His research interesets include multiple object tracking, object detection and computer vision.

Tao Wang received his B. E. degree of Electrical Engineering and Information from Shanghai Jiao Tong University, China, in 2019. He is currently a master student in Department of Electrical Engineering in Shanghai Jiao Tong University. His research interests include computer vision and machine learning.

Weiyao Lin received the B.E. and M.E. degrees from Shanghai Jiao Tong University, China, in 2003 and 2005, and the Ph.D degree from the University of Washington, Seattle, USA, in 2010. He is currently a professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. His research interests include image/video processing, video surveillance, computer vision.

Jian Wang received the B.E. and M.E. degrees from Wuhan University, China, in 2010 and 2012. He is currently an R & D Engineer at Baidu Ltd, China. His research interests include object detection, multiple object tracking, person re-identification, computer vision.

John See received his B.Eng., M.Eng.Sc. and Ph.D. degrees from Multimedia University, Malaysia. He is currently a Senior Lecturer and head of the Visual Processing Lab at the Faculty of Computing and Informatics, Multimedia University, Malaysia. His current research interests include computer vision, pattern recognition, video processing and affective computing.

Shilei Wen received the B.E. and M.E. degrees from Huazhong Univerisity of science and technology, China, in 2007 and 2011. He is currently the principal architect of Baidu's computer vision Department. His research interests include image/video understanding, image/video retrieval, video surveillance, computer vision and machine learning.

Errui Ding received Ph.D. degree in 2008 from Xidian University, and currently is the deputy director of Computer Vision Technology Department (VIS) of Baidu Inc. In recent years, he has published tens of papers on top-tier conferences such as ICCV/CVPR/ECCV/AAAI and was awarded Best Paper Runner-up by ICDAR 2019. In addition, his team received the winner prize in the well-known international competitions such as ActivityNet/VOT/WiderFace. He also co-organized two competition tracks, i.e., ArT and CSVT, in ICDAR 2019 and will co-organize the 2nd Workshop on Learning from Imperfect Data in CVPR 2020. In Baidu, his team has been applying these SOTA techniques to the search engine, news feeds, cloud and so forth. As a member, he serves in special committee of China Society of Image and Graphics.