Long-Term Tracking With Deep Tracklet Association

Yang Zhang, Hao Sheng, *Member, IEEE*, Yubin Wu, Shuai Wang, Weifeng Lyu, Wei Ke, and Zhang Xiong

***Abstract*— Recently, most multiple object tracking (MOT) algorithms adopt the idea of tracking-by-detection. Relevant research shows that the performance of the detector obviously affects the tracker, while the improvement of detector is gradually slowing down in recent years. Therefore, trackers using tracklet (short trajectory) are proposed to generate more complete trajectories. Although there are various tracklet generation algorithms, the fragmentation problem still often occurs in crowded scenes. In this paper, we introduce an iterative clustering method that generates more tracklets while maintaining high confidence. Our method shows robust performance on avoiding internal identity switch. Then we propose a deep association method for tracklet association. In terms of motion and appearance, we construct motion evaluation network (MEN) and appearance evaluation network (AEN) to learn long-term features of tracklets for association. In order to explore more robust features of tracklets, a tracklet-based training mechanism is also introduced. Tracklet groups are used as the input of the networks instead of discrete detections. Experimental results show that our training method enhances the performance of the networks. In addition, our tracking framework generates more complete trajectories while maintaining the unique identity of each target as the same time. On the latest MOT 2017 benchmark, we achieve state-of-the-art results.**

***Index Terms*— Multi-object tracking (MOT), tracking-by-tracklet, multiple hypothesis tracking (MHT), deep association.**

I. Introduction

WITH the rapid development of artificial intelligence technology in recent years, the demand in the field of safety supervision is also gradually increasing. As the basis of behavior analysis and anomaly detection, MOT is one of the most concerned research topics. Tracking multiple targets refers to obtaining the complete trajectory of each target in an image sequence.

The main difference with object detecting lies in not only accurately locating the position of each target in each frame, but also distinguishing the one-to-one correspondence between each bounding box and each target, thus obtaining independent trajectory for each target. Although new research results are published every year, especially tracking-by-detection methods, there are still some problems that are not effectively solved. In this paper, we mainly focus on the problems of low integrity and high fragmentation of trajectories that often occur in crowded scenarios. The complete trajectory of a target is broken into multiple fragmented trajectories usually because the target is not detected, such as detector failure or mutual occlusion.

To cope with detection failure, tracklet-based trackers are proposed. They use short trajectories, also known as tracklets, as the basis for target association and generates longer trajectories. In this way, trackers are less sensitive to error detector responses and individual missing detections. Tracklets are built by similar detections in consecutive frames. Widely used measurement methods include intersection-over-union (IOU), Euclidean distance, appearance similarity, etc. Although various methods are proposed for tracklet generation, most of them only consider the similarity between targets in adjacent frames and result in drift problem. In this paper, we introduce an iterative clustering method to ensure the high similarity between any two detections in the same tracklet.

Long-term occlusion among targets is another common problem that causes fragmentation of trajectories. Motion and appearance features are changing over time. So, it has great significance to extract features with long-term robustness for associating long-time interval detections or tracklets. Long Short-Term Memory (LSTM) networks have shown strong memory and non-linear transformation ability in detecting, classifying and visual tracking. We further develop its potential to build long-term features and introduce a deep association method for tracklet association. Thus, we build motion evaluation network (MEN) and appearance evaluation network (AEN) to associate tracklets during tracking. Experimental results show that our method has significant improvement on maintaining long-term association of trajectories.

In summary, our main contributions include:

- An iterative clustering method to generate more tracklets with high confidence.

Manuscript received December 1, 2019; revised April 5, 2020; accepted May 4, 2020. Date of publication May 19, 2020; date of current version July 6, 2020. This study was supported in part by the National Key R&D Program of China (No.2018YFB2100500), the National Natural Science Foundation of China (No.61861166002, 61872025, 61635002), the Science and Technology Development Fund, Macau SAR (File no.0001/2018/AFJ), the Fundamental Research Funds for the Central Universities and the Open Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE2019ZX-04). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaolin Hu. *(Corresponding author: Hao Sheng.)*

Yang Zhang, Hao Sheng, Weifeng Lyu, and Zhang Xiong are with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China (e-mail: yang.zhang@buaa.edu.cn; shenghao@buaa.edu.cn; lwf@buaa.edu.cn; xiongz@buaa.edu.cn).

Yubin Wu and Shuai Wang are with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Beihang Hangzhou Institute for Innovation at Yuhang, Beihang University, Hangzhou 311121, China (e-mail: yubin.wu@buaa.edu.cn; shuaiwang@buaa.edu.cn).

Wei Ke is with the School of Applied Sciences, Macao Polytechnic Institute, Macau 999078, China (e-mail: wke@ipm.edu.mo).

Digital Object Identifier 10.1109/TIP.2020.2993073

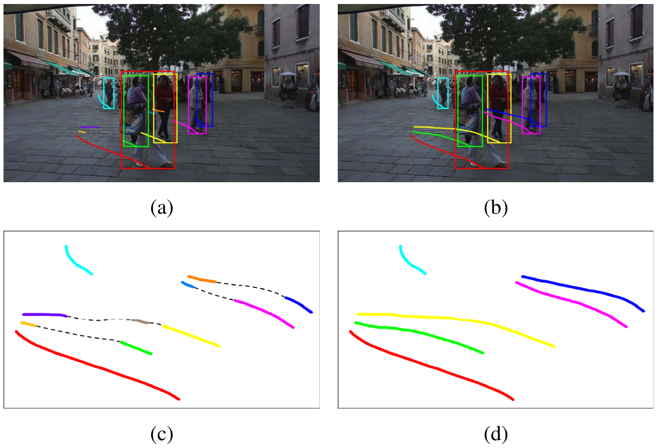


Fig. 1. (a) and (c) show the fragmentation problem in tracking multiple targets. Different colors represent different target identities and dashed lines indicate missing trajectory. The complete trajectory of a target is fragmented into pieces due to detector failure and target occlusion. (b) and (d) is our result by generating convincing tracklets and using deep association method.

- A tracklet-based mechanism for training networks in tracking.
- MEN and AEN networks to learn long-term features of tracklets for association.
- A tracklet-based tracking framework that generates more complete trajectories than previous trackers and achieves state-of-the-art performance on the popular MOT 2017 benchmark.

The rest of the paper is organized as follows. Related work is discussed in Sec.II. Our method to generate tracklets is introduced in Sec.III. The detailed tracking framework is described in Sec.IV. We show and analyze the experiment results in Sec.V. Conclusion is in Sec.VI.

II. Related Work

In recent years, the research on MOT has been further developed, most of which are based on tracking-by-detection approach [1]. It is currently the most mainstream algorithm framework. Different from single target tracking [2], [3], MOT task not only considers feature representation but also focuses on transforming tracking into a series of data association problems. Firstly, all targets in each independent frame are detected by the detector, and then these independent targets are connected by analyzing their similarity, thus forming continuous and complete trajectories. In addition to improving the recall rate and accuracy rate of detectors, how to improve the accuracy and robustness of data association has become an important research direction of multi-object tracking and various types of solutions have been proposed.

A. Tracking Frameworks

Network flow based trackers has shown convincing efficiency on tracking multiple targets. They design different cost functions between nodes and then solve the problem in polynomial time. Zhang *et al*. [4] proposed non-overlapping constraint and found the solution by a min-cost flow algorithm. Pirsiavash *et at*. [5] analyzed the changes in the number of trajectories. Later, Butt *et al*. [6] redefined the meaning of the nodes in the network flow. Each node represents a candidate of pair of matching detections. Pairwise costs were added into the tracker to reduce the influence of detector failure by Chari *et al*. [7]. Mclaughlin *et al*. [8] and Wang *et at*. [9] tried to track occluded targets by motion and interaction information. Benefiting from the advantages of network flow in solving efficiency, these trackers can achieve good speed, but the accuracy decreases obviously in complex scenes.

It is a common phenomenon that multiple detector responses actually belong to the same target, so Tang *et al*. [10]–[12] regarded MOT as a minimum cost subgraph multi-cut problem. It clusters plausible detections of the same target and links them through frames. They presented a method to solve the graph, but it is an expensive approximate algorithm to find the sub-optimal solution.

Some other trackers are developed based on energy minimization method. Andriyenko *et al*. [13]–[17] proposed a continuous energy function to describe tracking task and constructed an optimization scheme to find local minima of the energy. Milan *et al*. [18] used superpixel-level segmentation to capture partial occluded targets and improved the tracking performance in crowded scenes. However, as the scene of the tracking task becomes much more complicated, the difficulty of searching minimum energy has been increased, which affects the robustness and stability of trackers.

Multiple hypothesis tracking (MHT) is another type of the popular trackers. MHT is a breadth-first search algorithm and was originally designed for radar tracking by Reid [19]. Years later, Cox *et al*. [20] suggested that MHT is suitable for visual tracking in certain scenes and presented a feasible implementation. Papageorgiou *et al*. [21] introduced maximum weight independent set problem (MWISP) into MHT for data association. The main defect of MHT is that its exponentially growing search space makes the solution inefficient. Kim *et al*. [22] trained online appearance models for each hypothesis and thus improved the efficiency and robustness of tree pruning. Chen *et al*. [23] extended MHT by analyzing the relationship between detections and scenario to deal with false detector responses. Sheng *et al*. [24] exploited superpixel-level information for recovering missing detections, and then redesigned MHT framework based on tracklets in [25].

B. Deep Learning Methods

In the special field of MOT, the application of deep learning is limited to a certain extent compared with object detection [26] and re-identification [27]. MOT is a typical small sample learning problem, which makes it difficult for deep learning methods to give full play to their advantages. End-to-end deep learning based trackers [28], [29] did not show remarkable improvement. In spite of this, the deep learning methods dramatically promote the development of visual tracking through motion modeling and appearance representation.

In the aspect of motion model, researchers mainly focus on detection noise, object occlusion and object interaction. Alahi *et al*. [30] presented a Long Short-Term Memory (LSTM) model to learn the interactions between targets for motion trend prediction. Robicquet *et al.* [31] collected a large-scale dataset to learn typical motion styles in real world. Zhu *et al.* [32] proposed dual matching attention network to handle noisy detections and frequent interactions. Ren *et al.* [33] described a deep reinforcement learning method to reduce the influence of occlusion and noise.

As for appearance model, deep learning based features have been widely studied [34] in recent years. Due to the improvement of video quality [35], the target has more effective information in the videos. It enables deep learning methods to take better advantage of feature representation and defeat manually constructed features in terms of both distinguishability and robustness. Kim *et al.* [22] used the convolutional neural network (CNN) features from [36] and reduced its dimensionality to 256 for online appearance learning and updating. Leal *et al.* [37] trained a siamese convolutional neural network to generate matching probability between detections. Ristani *et al.* [38] designed an adaptive weighted loss in CNN for appearance matching. In [39], LSTM was adopted to improve the performance of appearance modeling. Maksai *et al.* [40] introduced an iterative scheme to minimize the number of identity switches during training and learned a scoring function for association. Chen *et al.* [41] presented a method to align appearance features of tracklets. They decomposed and aggregated the features of targets with the spatial-temporal attentions.

C. Tracklet-Based Tracking

Tracking by tracklets instead of discrete detections has received more attention in recent years [25], [42]–[44]. It has better performance in avoiding identity switches and recovering missing detections. Wang *et al.* [45] presented an online learned method to describe motion and appearance features of tracklets. Shen *et al.* [46] trained a learnable network flow to associate tracklets. Wang *et al.* [29] proposed epipolar geometry to generate tracklets and built a multi-scale TrackletNet to cluster tracklets into groups considering their appearance and temporal features. Chen *et al.* [41] proposed a multitask CNN with both spatial and temporal features for appearance modeling of tracklets. However, these trackers fail to take full advantage of tracklet in suppressing identity switches and generating longer and more continuous trajectories.

III. Iterative Clustering for Confident Tracklet Generation

In this paper, we propose a tracklet-based tracker for multiple object tracking. Similar to tracking-by-detection framework, the bounding boxes of targets are first captured by detector and then assigned with different labels to represent trajectories. In tracklet-based methods, detections are first organized into groups as tracklet before labeling. All detections of the same tracklet share the same label. Thus, the tracking task turns out to be giving labels to tracklets instead of discrete detections.

Obviously, the recall rate and accuracy of tracklets dramatically affect the performance of tracklet-based trackers. In this section, we introduce our iterative clustering method to generate confident tracklets while achieving a good balance among the length, accuracy and quantity of tracklets.

A. Definition

Let $D = \{d_1, d_2, d_3, \ldots, d_n\}$ denotes the set of all detections of the image sequence and $\tau_i = \{d_1, d_2, d_3, \ldots, d_k\}$ denotes the i^{th} tracklet that contains k detections. Each detection has a 1540-dimensional feature vector including 4-dimensional motion features and 1536-dimensional appearance features. It can be expressed as $v_k = (x_k, y_k, w_k, h_k, a_{1,k}, a_{2,k}, \ldots, a_{1536,k})$ where (x_k, y_k, w_k, h_k) is the location (midpoint of bottom edge of bounding box), weight and height of d_k and $a_k = (a_{1,k}, a_{2,k}, \ldots, a_{1536,k})$ is its 1536-dimensional appearance feature extracted from FC layer in [47]. In addition, the confidence of each detection c_k is also used during tracking which is given by the detector.

B. K-Partite Graph Based Clustering

Tracklet generation is a basic but crucial part of tracklet-based trackers. There are various types of solutions such as greedy algorithm, bipartite graph matching, network flow maximization, subgraph clustering, etc. Some of these methods generate tracklet linearly, so only similarity of detections between adjacent frames is considered. Therefore, the drift problem easily occurs and generated internal identity switch, as shown in Fig.2(a) (tracklet III). In this paper, we use a K-partite graph based clustering algorithm for tracklet generation that considers similarity between any two detections to restrain internal identity switch, as shown in Fig.2(b).

For an image sequence of n frames, we divide the whole sequence into several batches. Each batch is a window that contains l_{max} frames, while the step size is l_0. Then we generate tracklet batch by batch, e.g., frame 1 to frame l_{max}, frame $(1 + l_0)$ to frame $(l_0 + l_{max})$, frame $(n - l_{max} + 1)$ to frame n, etc.

We start with the first batch that contains frame 1 to frame l_{max} and then deal with all batches one by one through timeline. Without loss of generality, we use $D = \{d_1, d_2, \ldots, d_k\}$ to denote all the detections in this batch. Then, we build a K-partite graph $G = \{V; E; W\}$ to generate tracklets. In real tracking task, it is not necessary to link every two detections, so we only link detections with high similarity between different frames. The detailed definition of G is described as follows:

1) Node Set: For each element in node set V, let node v_i denote a detection d_i in the batch, then all the detections are expressed by V.

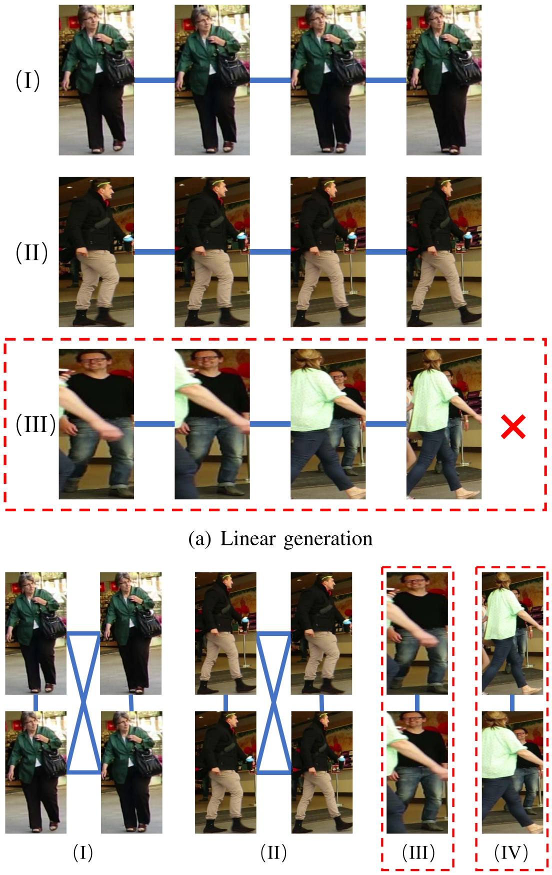
2) Edge Set: For any two node v_i and v_j that represent detections from different frames, there is an edge e_{ij} between them.

3) Weight Set: The weight w_{ij} of each edge e_{ij} describes similarity between d_i and d_j including location and appearance. It can be expressed as follows:

$$w_{ij} = w_{1,ij} \cdot w_{2,ij} \tag{1}$$

$$w_{1,ij} = \left\|(x_i, y_i), (x_j, y_j)\right\|_2 \tag{2}$$

$$w_{2,ij} = cos(a_i, a_j) \tag{3}$$



(a) Linear generation

(b) Clustering generation

Fig. 2. (a) and (b) show two kinds of algorithms for tracklet generation. Internal identity switch occurs in tracklet III in (a), while (b) divides four detections into two separate tracklets.

assigned by the cosine distance between $(a_{1,i}, a_{2,i}, \ldots, a_{1536,i})$ and $(a_{1,j}, a_{2,j}, \ldots, a_{1536,j})$.

Then we can get the best cluster solution of G by finding the maximum sum of all weights as formulated below:

$$\arg\max_{e_{ij}} \sum_{i,j} w_{ij} e_{ij} = \sum_{i,j} w_{1,ij} w_{2,ij} e_{ij} \tag{4}$$

$$s.t. \quad e_{ij} + e_{jk} \leqslant e_{ik} + 1 \tag{5}$$

$$e_{ij} = 0, \quad w_{1,ij} > max(h_i, h_j) \tag{6}$$

$$e_{ij} = 0, \quad w_{2,ij} < app_{TH} \tag{7}$$

$$e_{ij} \in \{0, 1\} \tag{8}$$

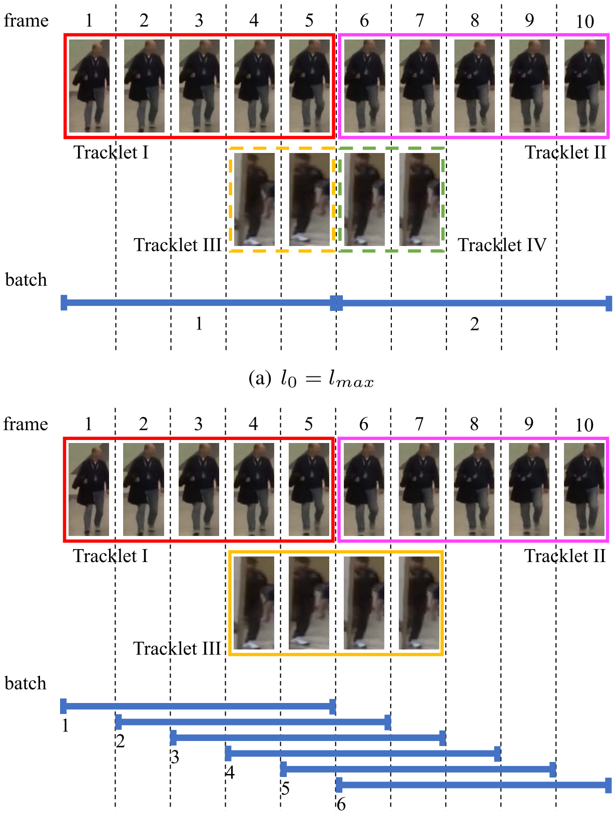
$$e_{ij} = e_{ji} \tag{9}$$

Specifically, Eq.(5) is used to ensure that all the nodes are linked to each other in any clique in the solution. The value of e_{ij} is constrained to 0 or 1, so the solutions of this optimization problem are all integer solution. We use linear program method to solve it and get the optimal solution. Any edge e_{ij} with its weight $w_{ij} = 0$ can be removed, then graph G is divided into several subgraphs that can be solved respectively, thus improving the efficiency of solving the optimization problem.

In the optimal solution, we get a set of clique $C = \{C_1, C_2, \ldots, C_n\}$. For each C_i, it contains at most l_{max} nodes, representing a tracklet that is no longer than l_{max}. We select C_i with no less than l_{min} nodes, and detections corresponding to these nodes will not be used in later batches. After getting all the solutions in all batches, their union C_{all} consists of all candidate tracklets. We do the following processing to get the final tracklets.

- First, we remove all the cliques with less than l_{min} nodes;
- Then, we remove every clique that is a subset of any other clique in C_{all};
- According to time order, we select pairs of cliques from two adjacent batches with non-empty intersection. We keep the clique with higher sum of weight and remove the other clique.

In this way, we get a set of cliques, and each clique contains l_{min} to l_{max} nodes to represent a tracklet. In addition, each node exists in at most one clique, thus avoiding the situation that multiple tracklets share the same detection.



(a) $l_0 = l_{max}$

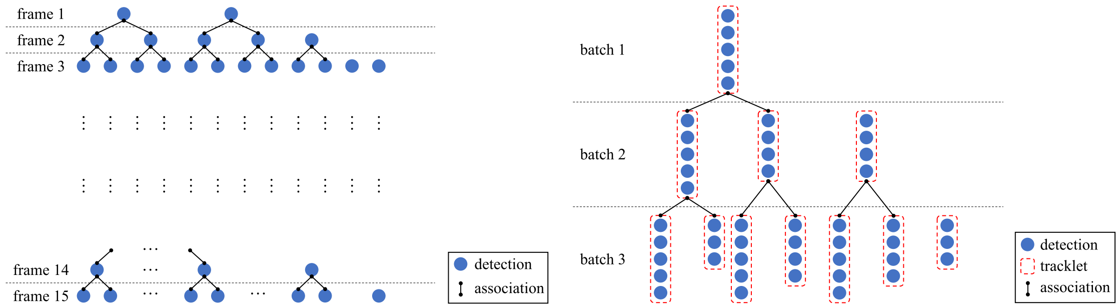
(b) $l_0 = 1$

Fig. 3. (a) and (b) show the different settings of step size l_0. l_{max} is set to 5 in the figures. Different settings will eventually result in different tracklets.

C. Iterative Generation

As the cluster problem is solved iteratively with the window size $= l_{max}$ and step $= l_0$. In order to cover the whole image sequence, l_0 can be set from 1 to l_{max}. A similar clustering method is used for tracklet generation in [25] and they set l_0 to l_{max} to cover all frames. However, setting step size as same as window size arises an unavoidable problem as demonstrated in Fig.3.

In Fig.3(a), tracklet III and IV contain only two frames of detections respectively. They regard these short tracklets as unreliable tracklets and remove them from the candidate



(a) Traditional MHT framework

(b) Tracklet-based MHT framework

Fig. 4. (a) shows traditional tracking-by-detection MHT framework which tracks targets frame by frame. (b) is the tracklet-based MHT framework. It constructs and updates hypothesis trees with tracklets in the batch.

tracklets that are used for tracking. In our method, we set $l_0 = 1$ to scan the whole sequence frame by frame. In Fig.3(b), tracklet III is generated by batch 4 and contains 4 independent detections. Although this method will lead to the same detection being used to generate multiple tracklets, our subsequent processing strategy presented in Sec.III-B ensures that each detection belongs to only one tracklet in the final results. In the following, $l_0 = 1$, $l_{min} = 3$, $l_{max} = 5$ unless otherwise specified. Parameter analysis is in the experimental part in Sec.V.

IV. Long-Term Tracklet Association With LSTMs

In this section, we introduce our deep association method for tracking multiple targets based on LSTMs. We construct motion evaluation network (MEN) and appearance evaluation network (AEN) to learn long-term features of tracklets. Our tracking method is based on MHT framework which is a classical method for tracking multiple targets. It builds hypothesis trees for targets and each tree has several branches to represent potential trajectories. The optimal hypothesis in each tree is selected by strategic delay. At the same time, global optimization and pruning strategies are also used in MHT to deal with conflict and to control the exponential growth of its scale.

A. Overview

In traditional MHT based trackers, the basic nodes are independent detections from detectors. Association relationship is built between detections in adjacent frames. Thus hypothesis trees are constructed and updated frame by frame as demonstrated in Fig.4(a). Compared with them, our tracker focuses on the association between tracklets instead of detections in hypothesis trees. In Fig.4(b), the basic nodes in the trees are tracklets generated in Sec.III. The similarity between independent detections is not used in the tree construction process, but instead the measurement and association between tracklets.

Since detection in each frame is no longer the basic node in our tracklet-based tracker, the hypothesis tree is updated by batch that consists of several continuous frames. In order to avoid ambiguity between tracklets, multiple tracklets of the same target should fall into different batches. Therefore, we set the length of all batches to be l_{max} which is the maximum length of tracklets.

B. Track Tree Construction and Updating

Similar with traditional MHT framework, there are two main process of the growing of hypothesis trees, constructing new trees and updating existing trees. Based on the first frame of each tracklet, we divide all tracklets into different batches.

For a given batch, new hypothesis trees are built for each tracklet in this batch. These new trees represent new targets appearing in the scene. Since each node in the tree represents a tracklet rather than detection, most discrete wrong detections generated by detector are not used in the tracklets. In this way, compared with traditional MHT trackers, our method suppresses false hypothesis trees from the root.

Then, we updates hypothesis trees to represent the temporal extension of targets' trajectories. Already existing trees are extended with tracklets in current batch. Because the time span of adjacent batches is quite small, features such as position and appearance of the target do not change suddenly in most cases. Therefore, we have filtered tracklets through a basic similarity measure, and only tracklets similar to their parent nodes are used for the updating of hypothesis trees. Because it is only a preliminary filtering and considering the computational complexity, we use the average value of detections in tracklets to select candidate tracklets for updating as expressed below:

$$\left\|(\bar{x_p}, \bar{y_p}), (\bar{x_c}, \bar{y_c})\right\|_2 \leqslant \ max(\bar{h_p}, \bar{h_c}) \qquad (10)$$

$$cos(\bar{a_p}, \bar{a_c}) \geqslant \ app_{TH} \qquad (11)$$

For a given tracklet τ_c in current batch, we calculate its average location $(\bar{x_c}, \bar{y_c})$ and appearance feature $\bar{a_c}$. The average of its parent node τ_p is also calculated and denoted as $(\bar{x_p}, \bar{y_p})$ and $\bar{a_p}$. If both of their Euclidean distance of location and cosine distance of appearance satisfy Eq.10 and Eq.11, τ_p will be extend by τ_c as a child node.

In addition, we use the dummy node mechanism to deal with missing detections. When extending a leaf node with similar tracklets in current batch, we use a dummy node to extend this node as well. Dummy nodes share the same features (x_i, y_i, w_i, h_i and a_i) as their parent nodes that are used for selecting potential extension. Although using tracklets instead of detections as nodes in the hypothesis trees can reduce the impact of occasional detector failure, continuous missed detection of targets still leads to trajectory breakage, such as long-term occlusion. These dummy nodes can connect tracklets in non-adjacent batches together, thus improving the integrity of potential trajectories.

C. Deep Association for Tracklets

The core idea of MHT algorithm is to delay decision-making, and select the globally optimal branch by comparing all hypotheses. In this section, we introduce our deep association method to measure and evaluate branches in hypothesis trees.

Each branch in MHT represents a potential trajectory of a target, so evaluating each branch is to score the confidence of this hypothesis. How to construct stable long-term features to accurately describe the changes of target features over time is always an important issue in MOT. Whether in experimental datasets or practical applications, video scenes are becoming much more complex. Many feature description models proposed earlier have been unable to accurately describe the feature change trend of targets, and many errors have occurred in associating targets and evaluating trajectories. In terms of motion features, IOU algorithm, linear model, pairwise model and other methods have been proposed. As for appearance features, there are cosine algorithm, weighted average algorithm, online learning, etc.

Recurrent neural network (RNN) is a kind of network with memory ability. Compared with traditional neural network, it has better processing ability for time series data. Compared with vanilla RNN, LSTM network can effectively avoid the problem of gradient disappearance, and can retain the earlier data features in back propagation. Therefore, LSTM networks have many applications in recent MOT algorithms.

The common usage of LSTM network mainly includes two types, regression and classification. In the former category, the current and future frames are predicted by the existing trajectory, and then compared with detection (e.g., calculating IOU), and similar detections are selected as potential extension of the trajectory. Other methods connect the existing trajectory and the detections in current frame into new trajectories respectively, and then classify them in turn to select those trajectories that are more likely to represent the same target.

By comparing the theoretical analysis and experimental results in many literatures, we find that the regression method has higher accuracy but lower recall rate. This is because the detection in actual tracking often contains noise, such as position offset or inaccurate size. These noises cause errors in the extraction of motion and appearance features of the target, which lead to deviations in the prediction of LSTM networks. Therefore, LSTM networks are required to have strong anti-noise robustness. In addition, the trajectories of various targets vary greatly, and a large amount of training data is needed to obtain reliable prediction, otherwise it is easy to have over-fitting problems.

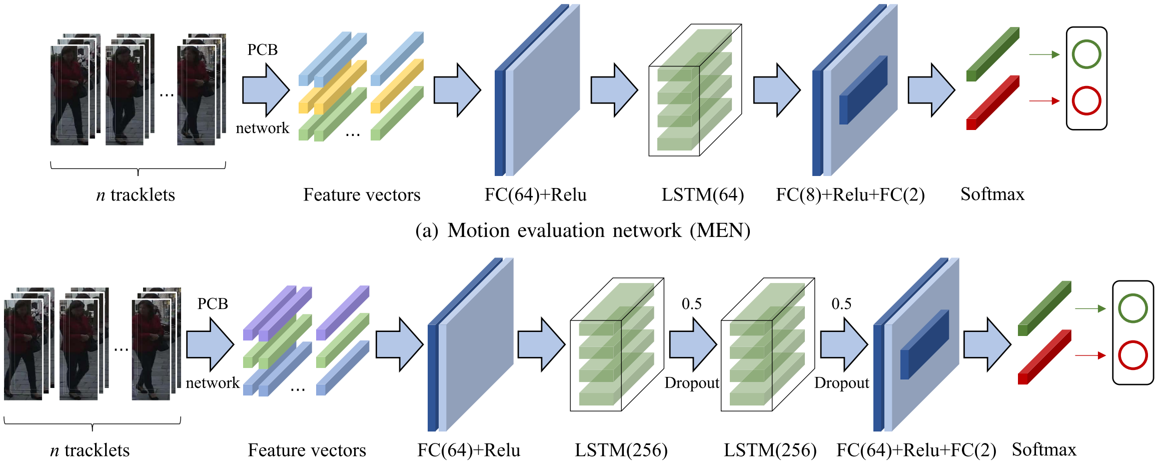
In contrast, the classification method has a false positive problem, but also has a higher recall rate. Although these LSTM networks do not directly predict the motion or appearance features of the target, the potential probability of each detection can be given by comparing the detection of the current frame with the existing trajectory. Because the MHT framework has strong search and comparison decision-making capabilities, we prefer add potential branches to the hypothesis tree as much as possible to avoid false negative. Therefore, we are convinced that LSTM networks for classification are more suitable for MHT.

In addition, it cannot be ignored that the input of the networks has a great influence on the evaluation performance. Early researches [48], [49] on deep learning for tracking mainly focused on the structure of the network. For a long time, tracking networks just took discrete detections as input during training and testing. Recently, tracklet-based tracking has shown impressive performance and training with tracklets has also been widely discussed [39]. However, the input tracklets only simply splicing multiple detections. The rich temporal information among detections in the tracklet is not fully explored. Therefore, we introduce a novel tracklet-based training mechanism in this study to learning deeper features of the tracklets. The detail of our network structure and training method is discussed as follows.

Similar to most popular methods, we evaluate each branch in the hypothesis tree from two aspects: motion and appearance. Appearance feature consists of a 1536-dimensional vector, while motion feature is only a 4-dimensional vector. Considering the huge difference in dimension between the two features, we construct two different LSTM classification networks for motion and appearance to evaluate branches, as demonstrated in Fig.5.

First, we introduce our motion evaluation network (MEN) for motion features. MEN is a sequence-to-one type of LSTM network, in which the LSTM layer contains 64 hidden units. The input of the MEN is n data sets (n tracklets) and the output is p_m, which indicates the probability that the n data sets belong to the same target. Each data set in the input consists of 3 feature vectors, and each vector contains 4-dimensional features. Features include $(\frac{\frac{x_i}{W}-\mu_x}{\sigma_x}, \frac{\frac{y_i}{H}-\mu_y}{\sigma_y}, \frac{\frac{w_i}{W}-\mu_w}{\sigma_w}, \frac{\frac{h_i}{H}-\mu_h}{\sigma_h})$, where W and H are the width and height of the image, while μ and σ represent mean value and standard deviation.

Our appearance network is also a sequence-to-one LSTM network, called appearance evaluation network (AEN). But different from MEN, AEN contains 2 LSTM layers with 256 hidden units. Graves $et\ al$. [50] demonstrated that stacking multiple RNN layers has a better effect than only increasing the number of hidden units, and can learn more abundant features. Due to the high dimension of appearance features, we stack 2 LSTM layers to learn the changes of appearance features of the target with time. The upper LSTM layer outputs a sequence instead of a single value to the LSTM layer below. In order to reduce over-fitting, we add a dropout layer with the probability of 0.5 after the LSTM layers respectively. The input of the network is n data sets (n tracklets) as well, and the output is p_a to indicate the probability of the input belong to the same target. Data set of the input is 3 feature vectors with 1536-dimensional appearance features. Features include $(\frac{a_1-\mu_{a_1}}{\sigma_{a_1}}, \ldots, \frac{a_{1536}-\mu_{a_{1536}}}{\sigma_{a_{1536}}})$, where μ and σ are mean value and standard deviation.



(a) Motion evaluation network (MEN)

(b) Appearance evaluation network (AEN)

Fig. 5. (a) is the structure of Motion Evaluation Network (MEN), and (b) is the structure of Appearance Evaluation Network (AEN).

Both the inputs to MEN and AEN are all nodes of a branch in the track trees. The n inputs consist of n tracklets traced upward by the leaf nodes. Considering the different lengths of tracklets, from $l_{min} = 3$ to $l_{max} = 5$, we unified the lengths of all tracklets to 3 before inputting them into the network. For tracklets longer than 3, we retain the detection of the first and last frames, and take the average value of the other remaining detections as the third node. Then, these adjusted tracklets with the same length are taken as the input of the networks, and the networks output the probability of each hypothetical branch in terms of motion and appearance. For a specific leaf node, its evaluation score can be denoted as S_i, which can be calculated as follows:

$$S_i = S_{i-1} + p_{m,i} + p_{a,i} \qquad (12)$$

$$S_0 = 0 \qquad (13)$$

where S_{i-1} is the score of its parent node, $p_{m,i}$ and $p_{a,i}$ is the probability, 0 to 1, from the classification layer in MEN and AEN. In addition, the score of dummy node is set to 0 as penalty. As described in Eq.12, we do not only use the score of the leaf node, but evaluate the branch by accumulating the scores of all nodes on the branch. In this way, the branches with more nodes can have higher scores, thus stimulating the generation of longer trajectories.

D. Training Sequences

Designing appropriate training data for networks is a key step of deep learning method. By referring to the experience and results of related applications, we construct the training data of our networks by the following methods.

Due to the different forms of trajectories in complex scenes and various camera angles, many literatures have pointed out that tracking tasks require a large amount of data for training networks. Specifically, Manen *et al.* [51] verified through experiments that large-scale training data can effectively improve the performance of tracking algorithm. In this paper, we use datasets including KITTI tracking [52], MOT 2017 [53], CVPR19 challenge [54] and PathTrack [51]. KITTI tracking is a subset a KITTI dataset which is a well-known dataset in computer vision. It consists of 21 training and 29 test sequences for cars and pedestrians. MOT17, CVPR19 and PathTrack are specially designed for pedestrian tracking. Due to the variety of videos, MOT17 has become a popular dataset in recent years containing 21 training and 21 test sequences. In addition, CVPR19 and PathTrack have much larger scale and can be used as an effective supplement. These datasets provide labeled ground truth data and include manually labeled positions of occluded targets. For occluded targets, the datasets have special marks for distinguishing. Meanwhile, they cover a wide range of video types, such as sports competitions, street interviews, car videos, surveillance videos, etc.

First, we introduce the organization of training data for MEN. Different from the bounding boxes in ground truth, the detections in actual tracking are noisy, including missed detection, false detection and position offset. We find that the recall rates of commonly used detectors are about 70% to 80% in datasets mentioned above, such as FRCNN, YOLOv3, SDP, etc. Therefore, in order to make the training data as similar as possible to the real distribution of the detections, we first randomly delete 25% of the detections for each trajectory in the ground truth to simulate the detection noise. In addition, the precision rate of these detectors reach about 90%, so we randomly selected 10% of the remaining ground truth and shifted their positions so that the IOU ratio of the new position to the old position is greater than 0 and less than 0.5.

After that, we randomly divide each trajectory into groups of 3-5 frames in length to represent tracklets in real tracking. Then, we use the same average-based method we present in the previous subsection to unify the number of detections in groups to 3. All groups contained in each trajectory are then used as inputs to the MEN. Each input sequence consists of n_i groups of tracklets. Each group has 3 vectors with 4-dimensional features including $(\frac{\frac{x_i}{W}-\mu_x}{\sigma_x}, \frac{\frac{y_i}{H}-\mu_y}{\sigma_y}, \frac{\frac{w_i}{W}-\mu_w}{\sigma_w}, \frac{\frac{h_i}{H}-\mu_h}{\sigma_h})$, where W and H are the width and height of the image, and $\mu_{x,y,w,h}$ and $\sigma_{x,y,w,h}$ are mean value and standard deviation for each dimension.

As for training AEN, we make additional process on the datasets. Although the ground truth provides the real bounding

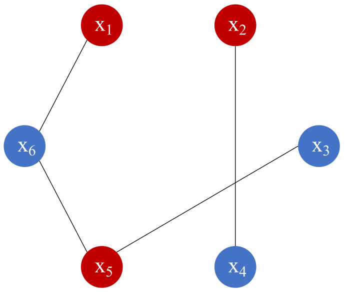


Fig. 6. A demonstration of MWIS problem that describes conflict between hypotheses. Nodes indicate hypotheses in different track trees while edges link hypotheses that have conflict of representing the same trajectory. Red nodes are the solution of the problem that do not have any conflict.

box position of the occluded target, different from motion features, it is meaningless to extract appearance features. When the target is occluded, its position can be gained through ground truth. However, limited to two-dimensional image, the image only contain the information of the occlusion, and cannot represent the information of the occluded target. Therefore, we remove the data in ground truth with a visible rate less than 0.5 or marked as occlusion. (The data provided by different datasets are slightly different.) Then, we process the data in the same way as generating training data for MEN. Finally, we get a series of input sequences. Each sequence represents a trajectory in the ground truth, and contains several groups of tracklets. Each group consists of 3 vectors with 1536-dimensional features including $(\frac{a_1-\mu_{a_1}}{\sigma_{a_1}}, \ldots, \frac{a_{1536}-\mu_{a_{1536}}}{\sigma_{a_{1536}}})$, where $\mu_{1,\ldots,1536}$ and $\sigma_{1,\ldots,1536}$ are mean value and standard deviation for respective dimensions.

In MEN, we set initial learning rate to 0.01 with batch size of 128. For training AEN, we adjust the learning rate to 0.001 with batch size of 64. For both MEN and AEN, we set the drop rate of learning rate to 0.9 for every epoch while the maximum epoch is 100. The Adam optimizer is used for both training as well.

E. Global Optimization

After all branches in each batch are scored, we get a series of hypotheses from different track trees. Since all the tracklets in each batch are used as the root nodes to build new track trees, branches from different trees may represent the same trajectory. Therefore, we need to optimize hypotheses globally to deal with the conflicts among branches. This task can be defined as a problem of finding the maximum weighted independent set (MWIS), as demonstrated in Fig.6. Assuming there are n hypotheses in current batch, the problem can be described as follows:

$$\arg\max_{x_i} \sum_{i=1}^{n} w_i x_i \tag{14}$$

$$s.t. \quad x_i + x_j \leqslant 1, \; x_i \; and \; x_j \; is \; conflicted \tag{15}$$

$$x_i \in \{0, 1\} \tag{16}$$

where x_i is the indicator of the i^{th} hypothesis and w_i is its score calculated by Eq.12. x_i is set to 1 if the i^{th} hypothesis is selected, otherwise set to 0. For multiple hypotheses that conflict with each other, Eq.15 limits that at most one of them can be selected. We solve the problem by the method that we have introduced in our previous work in [25].

V. Experiments

In this section, we compare and analyze the effectiveness of our proposed tracklet generation and deep association method. Then, we show qualitative and quantitative tracking results on the public benchmark. In this study, our method is implemented in MATLAB R2019b and the main hardware configuration includes: Intel(R) Core(TM) i7-9700K, Nvidia GeForce RTX 2080Ti.

A. Datasets and Metrics

We evaluate our tracker on MOT 2017 [53] which is the latest public benchmark for MOT. There are 42 sequences (21 training, 21 test) with 33,705 frames in MOT 2017. It consists of the same video as MOT 2016 but has different sets of detections for each video by three detectors. It is well known that the performance of the detector has a great influence on trackers. Therefore, we use MOT 2017 instead of MOT 2016 for comparison experiments as it can better evaluate the performance of tracking algorithm under different detector conditions. In addition, for fair comparison and to test the real performance of our method, we do not use any private detectors to gain additional detections.

We adopt the widely used CLEAR MOT metrics [55] for quantitive evaluation. There are some basic items such as FP↓ (false positives), FN↓ (false negatives), IDS↓ (identity switches), MT↑ (mostly tracked, $>$ 80%), ML↓ (mostly lost, $<$ 20%) and track fragmentations (FM)↓. MOTA↑ (multiple object tracking accuracy) is a main overall indicator that combines FP, FN and IDS. Another overall evaluation is IDF1↑ [56]. It is the ratio of correctly identified detections over the average number of ground truth and computed detections. MOTA mainly concerns with targets are tracked or not, while IDF1 evaluates whether a target is labeled with a unique ID. The indicator ↑ means the higher the better and ↓ means the lower the better.

B. Tracklet Generation Analysis

The length of tracklet has an important influence on its accuracy and should be carefully considered. On one hand, tracklet should be controlled within a certain length to avoid internal identity switch. On the other hand, tracklet should be long enough to contain more spatial-temporal features to make it more convincing than discrete detections.

The tracklets in our method are generated iteratively based on a sliding window, so the window size, that is, the maximum length of tracklet, affects the total number of tracklets and the time consumption for generation. There is a negative correlation between the total number of tracklets and the computational time. As the window increases, the maximum length of tracklets increases, thus the number of detections contained in each tracklet increases, resulting in a decrease in the total number of tracklets. However, as the window size

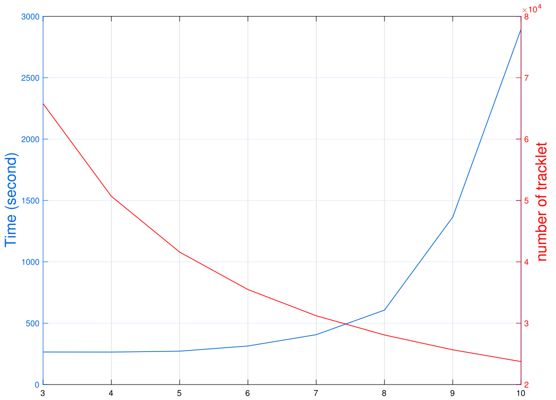


Fig. 7. Time consumption (blue) and the number of tracklets (red) with different window size from 3 to 10.

TABLE I

TRACKLET ACCURACY AND TIME CONSUMPTION

window size	#tracklet	without ID Sw.	with ID Sw.	time (s)
3	65,780	65,056 (98.9%)	723 (1.1%)	265.2
4	50,680	50,174 (99.0%)	506 (1.0%)	264.7
5	41,585	41,253 (99.2%)	332 (0.8%)	272.1
6	35,483	35,163 (99.1%)	319 (0.9%)	314.2
7	31,212	30,837 (98.9%)	374 (1.1%)	406.5
8	28,080	27,715 (98.7%)	365 (1.3%)	606.7
9	25,661	25,302 (98.6%)	259 (1.4%)	1365.1
10	23,746	23,462 (98.8%)	284 (1.2%)	2895.8

increases, the number of frames contained in each window increases and the number of detections increases, which leads to an increase in the complexity of the solving Eq.4, eventually resulting in an increase in time consumption.

We set window size l_{max} from 3 to 10 to compare the difference among tracklets. Experimental results on MOT 2017 training are shown in Fig.7 and Tab.I. There are totally 15,948 frames in the training set. As the window size increases from 3 to 10, the runtime increases exponentially from about 265 to nearly 2900 seconds and the number of tracklets gradually reduces to about 23,000, nearly one third of the maximum number 65,780. In Tab.I, we find that although the number of tracklets various dramatically, the rate of internal identity switches (ID Sw.) always has a high level at about 99% and does not change a lot as the window size changes.

Different from methods that only use the similarity between detections in adjacent frames, our tracklet generation method is based on clustering detections that considers similarity between any two detections. In this way, we can effectively deal with the drift problem, thus avoiding internal identity switches.

Larger window size is beneficial to reduce the number of tracklets that improves the tracking efficiency. However, it costs much more time to generate tracklets instead. Another point that should be noted is that as the window size increases, the length of the tracklet does not always increases to match the allowed maximum length. Because our iterative clustering tracklet generation method requires the high similarity between any two detections, the similarity between detections in two frames with a long distance can not meet the threshold requirement. Since our batch-by-batch tracking algorithm does not associate tracklets of the first frame in the same batch, we need to avoid multiple tracklets of the same target in the same batch. To sum up, considering the above reasons, and in order not to excessively adjust the parameters, we choose window size $l_{max} = 5$ for tracking all sequences in this paper.

C. Tracklet Association Comparison

Another innovation of this paper is the deep association method for tracklets. TLMHT [25] is a similar tracklet-based tracker and uses MHT framework as well. We take it as the baseline method to compare the performance of our deep association method.

We design four groups of comparative experiments, including baseline, baseline with tracklet generation method proposed in this paper (denoted as baseline & T), baseline with deep association (denoted as baseline & M + A), and the complete tracking method including tracklet generation and deep association (denoted as T + M + A). Experimental results are listed in Tab.II, and the best results are shown in bold.

In Tab.II, the first group is the result of the baseline method. Although it has the lowest FP, there are more FN, and the sum of FP and FN is also the highest. In addition, the identity switches problem of baseline is significantly higher serious than other groups. This shows that the baseline method has the worst performance in ensuring the unity of targets' identity among four groups.

In the second group, we replace the tracklet generation method in baseline with the method proposed in this paper. Although FP has increased by about 3000, FN has decreased by about 4000 and IDS has also decreased significantly, about one third. MOTA and IDF1, two comprehensive indicator, also improve slightly.

In the third group, we change the hypothesis branch evaluation method in baseline to the deep association method in this paper. Baseline uses the mean value of appearance features of tracklet to measure similarity by cosine distance. In contrast, our deep association method uses LSTM networks to extract deep features including motion and appearance, and has learned longer term features of tracklet. As can be seen from the table, compared with the first two groups, MOTA and IDF1 have both improved significantly. Compared with baseline, the rise of FP is controlled at about 4000, but FN is reduced by about 12000.

The last group is the complete tracking method proposed in this paper. On one hand, more tracklets with high confidence are gained through clustering-based tracklet generation method; on the other hand, the tracklets are associated and evaluated by deep association. Experimental results prove the effectiveness of the proposed method and show the best MOTA and IDF1, 56.5 and 67.0 respectively. The sum of FP and FN is greatly reduced compared with the other three groups, and IDS is controlled within 600.

In addition, we also conduct a comparison experiment to analyze the input of MEN and AEN. In our study, tracklets are used as the input of LSTMs instead of discrete detections to explore more robust features of trajectories. We compare the performance by using different inputs. As shown

TABLE II

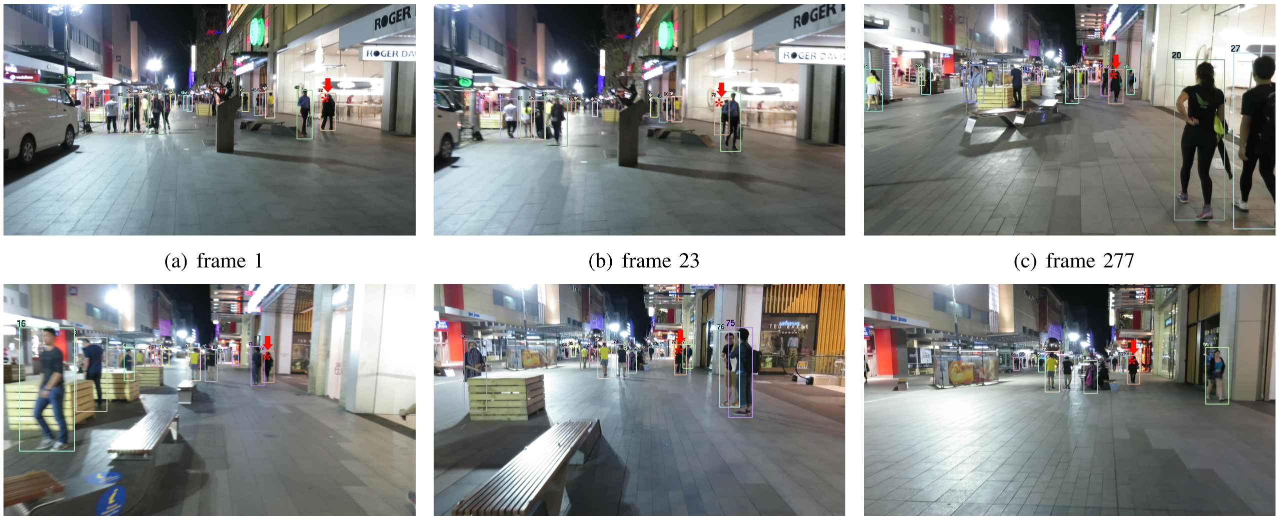
BASELINE COMPARISON ON MOT CHALLENGE 2017 TRAINING

No.	Method	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	FP+FN↓	IDS↓	FM↓
1	baseline [25]	52.6	62.8	26.3%	33.8%	**7,655**	150,529	158,184	1,519	1,505
2	baseline & T	53.2	63.9	22.4%	41.2%	10,814	146,263	157,077	**559**	**941**
3	baseline & M + A	55.0	66.2	26.0%	36.3%	11,871	138,736	150,607	864	1,199
4	T + M + A	**56.5**	**67.0**	**28.4%**	**33.6%**	9,116	**136,572**	**145,688**	572	1,315

TABLE III

MOT CHALLENGE 2017 TRAINING

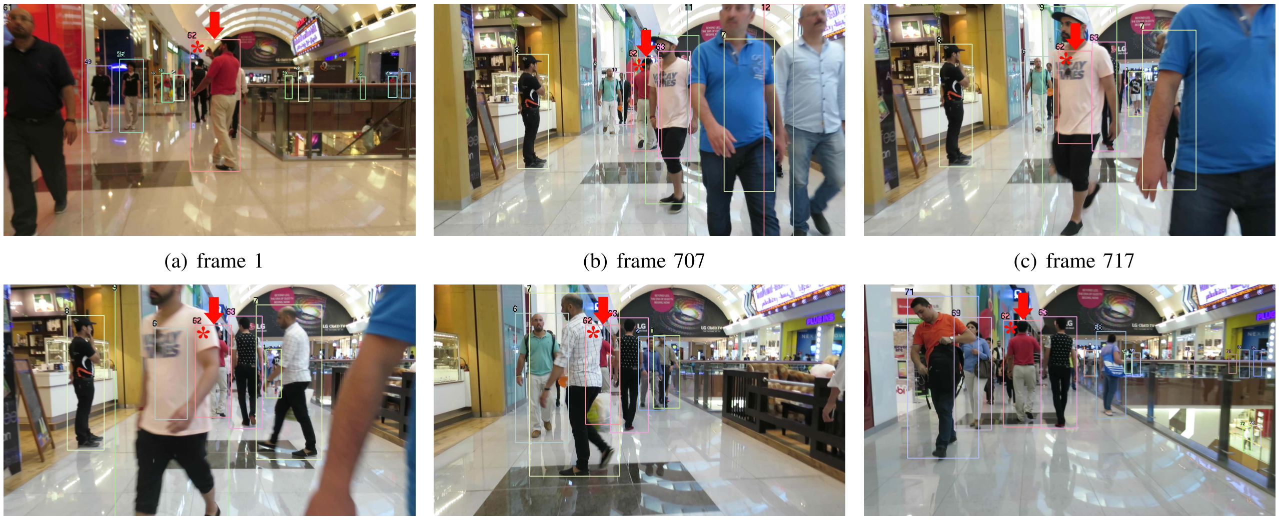
Method	Training Input	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	FP+FN↓	IDS↓	FM↓
baseline [25]	-	52.6	62.8	26.3%	33.8%	7,655	150,529	158,184	1,519	1,505
ours	tracklets	**56.5**	**67.0**	**28.4%**	**33.6%**	9,116	**136,572**	**145,688**	**572**	1,315
ours	detections	56.1	66.0	27.0%	36.1%	**7,267**	139,964	147,231	652	**1,177**



(a) frame 1 (b) frame 23 (c) frame 277

(d) frame 401 (e) frame 543 (f) frame 653

Fig. 8. Sample frames of the tracking result of MOT17-10-SDP, target (ID 72) is always tracked and maintains a uniform identity throughout the entire sequence.



(a) frame 1 (b) frame 707 (c) frame 717

(d) frame 727 (e) frame 757 (f) frame 900

Fig. 9. Sample frames of the tracking result of MOT17-11-SDP, including the first and the last frames and 4 frames that target (ID 62) is heavily occluded.

in Tab.III, whether using tracklets or detections as inputs, the performance of our tracking method is obviously better than the baseline. Specifically, training with tracklets has achieved higher MOTA and IDF1 than using detections. To our knowledge, training with detections for similarity evaluation is still the most common way for multi-object tracking. However, our comparison results prove that the network performance can be improved by taking tracklets as the training basis.

In order to intuitively show the performance of our tracking method in generating complete trajectories, we show the two

TABLE IV

RESULTS ON MOT CHALLENGE 2017 TEST(2019.11)

Method	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	FP+FN↓	IDS↓	FM↓
TT17 (ours)	**54.9**	**63.1**	**24.4%**	38.1%	20,236	233,295	**253,531**	**1,088**	2,392
LSST17 [58]	54.7	62.3	20.4%	40.1%	26,091	**228,434**	254,525	1,243	3,726
Tracktor17 [59]	53.5	52.3	19.5%	36.6%	**12,201**	248,047	260,248	2,072	4,611
LSST17O [58]	52.7	57.9	17.9%	36.6%	22,512	241,936	264,448	2,167	7,443
JBNOT [60]	52.6	50.8	19.7%	35.8%	31,572	232,659	264,231	3,050	3,792
FAMNet [61]	52.0	48.7	19.1%	**33.4%**	14,138	253,616	267,754	3,072	5,318
eTC17 [29]	51.9	58.1	23.1%	35.5%	36,164	232,783	268,947	2,288	3,071
eHAF [24]	51.8	54.7	23.4%	37.9%	33,212	236,772	269,984	1,834	2,739
AFN17 [46]	51.5	46.9	20.6%	35.5%	22,391	248,420	270,811	2,593	4,308
FWT [62]	51.3	47.6	21.4%	35.2%	24,101	247,921	272,022	2,648	4,279
NOTA [41]	51.3	54.5	17.1%	35.4%	20,148	252,531	272,679	2,285	5,798
jCC [63]	51.2	54.5	20.9%	37.0%	25,937	247,822	273,759	1,802	2,984
MOTDT17 [64]	50.9	52.7	17.5%	35.7%	24,069	250,768	274,837	2,474	5,317
MHT_DAM [22]	50.7	47.2	20.8%	36.9%	22,875	252,889	275,764	2,314	2,865
TLMHT [25]	50.6	56.5	17.6%	43.4%	22,213	255,030	277,243	1,407	**2,079**
DEEP_TAMA [65]	50.3	53.5	19.2%	37.5%	25,479	252,996	278,475	2,192	3,978
EDMT17 [23]	50.0	51.3	21.6%	36.3%	32,279	247,297	279,576	2,264	3,260

samples of tracking in MOT 2017 training. In Fig.8, a pedestrian (ID 72, labeled with the red asterisk) always appears in the picture from the first frame to the last frame. Through our tracking method, we have generated her complete trajectory. The whole video was taken with a hand-held camera, she was also moving away from the camera. Through frames, mutual occlusion occurs many times, but we keep her identity unchanged.

Another example is shown in Fig.9. A man in red (ID 62, labeled with the red asterisk) never left the picture. We generate his complete trajectory from the first frame to the last frame. Different from the previous example, he has a more serious occlusion problem. From frame 707 to frame 757, the detector fails to detect his position due to heavy occlusion. Through our deep association method, we learn its long-term features and successfully associate the tracklets before and after occlusion. The missing position is obtained through interpolation, and finally the continuous trajectory is obtained. Both examples show the effectiveness of our method in generating longer and more complete trajectories in complex scenes.

D. Benchmark Comparison

In this section, we test our tracking method on MOT 2017 test dataset and compare it with other competitive methods. In Tab.IV, we list the results of all tracking methods with MOTA higher than 50 and non-anonymous submission. Our method is denoted as TT17 in the table and the best results are shown in bold. The entire results can be found on MOT Challenge website.[1]

Compared with other MHT based tracking methods, such as eHAF [24], MHT_DAM [22], EDMT17 [23] and TLMHT [25], our method outperforms them obviously on both MOTA and IDF1. Compared with other tracket-based trackers, such as AFN17 [46], NOTA [41] and TLMHT [25], especially IDF1 and IDS, we perform much better than their results. These two indicators can effectively evaluate the performance of trackers in avoiding the change of target identity which is the main problem we aims to solve in this paper.

In addition, even compared with all the methods on the list, we get the lowest sum of FP and FN and achieve the best performance on MOTA at 54.9, IDF1 at 63.1. In addition, we get the highest MT rate and the lowest IDS at the same. In terms of FM, although our method is not the best, it ranks second to keep strong competitiveness.

VI. CONCLUSION

In this paper, we propose a clustered-based tracklet generation method to gain tracklets with high confidence. We consider the similarity between any two detections in a tracklet to avoid internal identity switch. Moreover, we discuss and analyze the influence of the length of tracklets on tracking, and verify the effectiveness of our tracklet generation method through comparative experiments. In addition, we introduce a deep association method for tracklet association. By constructing different LSTM networks from the aspects of motion and appearance features, MEN and AEN, we evaluate the hypothesis branches of the tracking tree in MHT. We explain in detail the structure of the networks and a novel tracklet-based training method is also introduced. By analyzing the experimental results, our deep association method shows convincing performance in associating long-term tracklets. On the latest MOT 2017 benchmark, we achieve state-of-the-art results compared with other previous methods.

ACKNOWLEDGMENT

Thank you for the support from HAWKEYE Group.

REFERENCES

[1] W. Luo *et al.*, "Multiple object tracking: A literature review," 2014, *arXiv:1409.7618*. [Online]. Available: http://arxiv.org/abs/1409.7618

[2] P. L. Mazzeo, P. Spagnolo, M. Leo, P. Carcagnì, M. D. Coco, and C. Distante, "Dense descriptor for visual tracking and robust update model strategy," *J. Ambient Intell. Hum. Comput.*, vol. 2017, pp. 1–11, Feb. 2017.

[3] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[4] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[1] https://motchallenge.net/results/MOT17/

[5] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. CVPR*, Jun. 2011, pp. 1201–1208.

[6] A. A. Butt and R. T. Collins, "Multi-target tracking by lagrangian relaxation to min-cost network flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1846–1853.

[7] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic, "On pairwise costs for network flow multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5537–5545.

[8] N. McLaughlin, J. M. D. Rincon, and P. Miller, "Enhancing linear programming with motion modeling for multi-target tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 71–77.

[9] X. Wang, E. Turetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2312–2326, Nov. 2016.

[10] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Subgraph decomposition for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5033–5041.

[11] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 100–111.

[12] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3539–3548.

[13] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Proc. CVPR*, Jun. 2011, pp. 1265–1272.

[14] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1926–1933.

[15] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3682–3689.

[16] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[17] A. Milan, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2054–2068, Oct. 2016.

[18] A. Milan, L. Leal-Taixe, K. Schindler, and I. Reid, "Joint tracking and segmentation of multiple targets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5397–5406.

[19] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[20] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 138–150, Apr. 1996.

[21] D. J. Papageorgiou and M. R. Salpukas, "The maximum weight independent set problem for data association in multiple hypothesis tracking," in *Proc. Optim. Cooperat. Control Strategies*, 2009, pp. 235–255.

[22] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4696–4704.

[23] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong, "Enhancing detection model for multiple hypothesis tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2143–2152.

[24] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang, "Heterogeneous association graph fusion for target association in multiple object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3269–3280, Nov. 2019.

[25] H. Sheng, J. Chen, Y. Zhang, W. Ke, Z. Xiong, and J. Yu, "Iterative multiple hypothesis tracking with tracklet-level association," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3660–3672, Dec. 2019.

[26] X. Cao, L. Yang, and X. Guo, "Total variation regularized RPCA for irregularly moving object detection under dynamic background," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 1014–1027, Apr. 2016.

[27] K. Lv, H. Sheng, Z. Xiong, W. Li, and L. Zheng, "Pose-based view synthesis for vehicles: A perspective aware method," *IEEE Trans. Image Process.*, vol. 29, pp. 5163–5174, Mar. 2020.

[28] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4225–4232.

[29] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with TrackletNet," 2018, *arXiv:1811.07258*. [Online]. Available: http://arxiv.org/abs/1811.07258

[30] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[31] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 549–565.

[32] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 366–382.

[33] L. Ren, J. Lu, Z. Wang, Q. Tian, and J. Zhou, "Collaborative deep reinforcement learning for multi-object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 586–602.

[34] C. Tang, L. Sheng, Z.-X. Zhang, and X. Hu, "Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4997–5006.

[35] X. Yuan, L. Fang, Q. Dai, D. J. Brady, and Y. Liu, "Multiscale gigapixel video: A cross resolution image matching and warping approach," in *Proc. IEEE Int. Conf. Comput. Photography (ICCP)*, May 2017, pp. 1–9.

[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[37] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 33–40.

[38] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6036–6046.

[39] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear LSTM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 200–215.

[40] A. Maksai and P. Fua, "Eliminating exposure bias and metric mismatch in multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4639–4648.

[41] L. Chen, H. Ai, R. Chen, and Z. Zhuang, "Aggregate tracklet appearance features for multi-object tracking," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1613–1617, Nov. 2019.

[42] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 788–801.

[43] A. R. Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 343–356.

[44] H. Sheng *et al.*, "Hypothesis testing based tracking with spatio-temporal joint interaction modeling," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Apr. 20, 2020, doi: 10.1109/TCSVT.2020.2988649.

[45] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association by online target-specific metric learning and coherent dynamics estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 589–602, Mar. 2017.

[46] H. Shen, L. Huang, C. Huang, and W. Xu, "Tracklet association tracker: An End-to-End learning-based association approach for multi-object tracking," 2018, *arXiv:1808.01562*. [Online]. Available: http://arxiv.org/abs/1808.01562

[47] H. Sheng *et al.*, "Mining hard samples globally and efficiently for person re-identification," *IEEE Internet Things J.*, early access, Mar. 13, 2020, doi: 10.1109/JIOT.2020.2980549.

[48] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 300–311.

[49] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6951–6960.

[50] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[51] S. Manen, M. Gygli, D. Dai, and L. Van Gool, "PathTrack: Fast trajectory annotation with path supervision," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 290–299.

[52] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[53] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*. [Online]. Available: http://arxiv.org/abs/1603.00831

[54] P. Dendorfer *et al.*, "CVPR19 tracking and detection challenge: How crowded can it get?" 2019, *arXiv:1906.04567*. [Online]. Available: http://arxiv.org/abs/1906.04567

[55] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Jan. 2008.

[56] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 17–35.

[57] W. Feng, Z. Hu, W. Wu, J. Yan, and W. Ouyang, "Multi-object tracking with multiple cues and switcher-aware classification," 2019, *arXiv:1901.06129*. [Online]. Available: http://arxiv.org/abs/1901.06129

[58] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.

[59] R. Henschel, Y. Zou, and B. Rosenhahn, "Multiple people tracking using body and joint detections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2019, pp. 770–779.

[60] P. Chu and H. Ling, "FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking," 2019, *arXiv:1904.04989*. [Online]. Available: http://arxiv.org/abs/1904.04989

[61] R. Henschel, L. Leal-Taixe, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1428–1437.

[62] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, "Motion segmentation & multiple object tracking by correlation co-clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 140–153, Jan. 2020.

[63] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[64] Y.-C. Yoon, D. Yong Kim, K. Yoon, Y.-m. Song, and M. Jeon, "Online multiple pedestrian tracking using deep temporal appearance matching association," 2019, *arXiv:1907.00831*. [Online]. Available: http://arxiv.org/abs/1907.00831

Yang Zhang received the B.S. degree from the School of Computer Science and Engineering, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Beihang University, China. His research interest is computer vision, especially in multiple object tracking.

Hao Sheng (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Computer Science and Engineering, Beihang University, China, in 2003 and 2009, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University. He is working on computer vision, pattern recognition, and machine learning.

Yubin Wu received the B.S. degree from the School of Computer Science and Engineering, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Beihang University, China. His research interest is computer vision, especially in multiple object tracking.

Shuai Wang received the B.S. degree from the School of Computer Science and Engineering, Beihang University, China, in 2019, where he is currently pursuing the Ph.D. degree. His research interest is computer vision, especially in multiple object tracking.

Weifeng Lyu is currently a Professor, the Dean of the School of Computer Science and Engineering, and the Vice Director of the State Key Laboratory of Software Development Environment, Beihang University. He is also the Leader of the Special Expert Group, Key Technology and Demonstration of Internet of Things and Smart City, Ministry of Science and Technology, China. His research interests include intelligent transportation and data analysis.

Wei Ke received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University. He is currently an Associate Professor with the Computing Program, Macao Polytechnic Institute. His research interests include programming languages and computer graphics. His recent research focuses on the design and implementation of open platforms for applications of computer graphics and pattern recognition, including programming tools, and environments.

Zhang Xiong received the B.S. degree from Harbin Engineering University in 1982 and the M.S. degree from Beihang University, China, in 1985. He is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Engineering, Beihang University. He is working on computer vision, information security, and data vitalization.