Tracklet Association by Online Target-Specific Metric Learning and Coherent Dynamics Estimation

Bing Wang, Student Member, IEEE, Gang Wang, Member, IEEE, Kap Luk Chan, Member, IEEE and Li Wang, Member, IEEE

Abstract—In this paper, we present a novel method based on online target-specific metric learning and coherent dynamics estimation for tracklet (track fragment) association by network flow optimization in long-term multi-person tracking. Our proposed framework aims to exploit appearance and motion cues to prevent identity switches during tracking and to recover missed detections. Furthermore, target-specific metrics (appearance cue) and motion dynamics (motion cue) are proposed to be learned and estimated online, i.e. during the tracking process. Our approach is effective even when such cues fail to identify or follow the target due to occlusions or object-to-object interactions. We also propose to learn the weights of these two tracking cues to handle the difficult situations, such as severe occlusions and object-to-object interactions effectively. Our method has been validated on several public datasets and the experimental results show that it outperforms several state-of-the-art tracking methods.

Index Terms—Multi-object tracking, tracklet association, target-specific metric learning, motion dynamics, network flow optimization.

1 INTRODUCTION

In this paper, we address the challenges in long-term tracking of multiple persons in a complex scene captured by a single, uncalibrated camera with an aim of achieving consistent person identity tracking (i.e. no identity switches). This is a challenging problem due to many sources of uncertainty, such as clutter, serious occlusions, targets interactions, and camera motion.

Recently, significant progress has been reported in human detection [15], [20], [19], [30], [49], [51], [52], and this promotes the popular tracking paradigm: detect-then-track [31], [38], [54], [62], [9], [44], [11], [6], [12]. The main idea is that a human detector is run on each frame to detect targets of interest, and then detection responses are linked across multi-frames to obtain target trajectories. In [62], [9], [44], [12], the authors formulate the multi-target data association as a network flow optimization problem. Zhang et al. [62] use a push-relabel method [25] to solve the min-cost flow problem. Berclaz et al. [9] and Pirsiavash et al. [44] propose to use successive shortest path algorithms, which can achieve roughly the same tracking results with less computation cost. In a more recent paper, Butt et al. [12] incorporate higher-order track smoothness constraints, such as constant velocity, for multi-target tracking. However, due to the limitation of the appearance cues used for tracking, the methods mentioned above usually cannot deal with longer term tracking to obtain a complete trajectory of a target. This is because prolonged occlusions and target-to-target interactions will result in fragmentation of a trajectory.

By using of the information from previous, current, and subsequent frames, trajectory can be recovered from the fragments and tracking errors such as missed tracks or identity switches can be corrected. In our earlier work [50], we advocate a discriminative target-specific appearance-based affinity model to reinforce the appearance cues for multi-person tracking. Unlike the PIRMPT system proposed by [34], which requires
off-line learned local descriptors, our target-specific metrics are online learned during tracking. In [50], we utilized a motion constraint based on heuristics. In this paper, we exploit motion dynamics to further improve tracking of target’s identity. Furthermore, we study the significance of the appearance and motion cues on tracking performance independently. Different from previous works [34], [56], [58], which simply multiply the motion and appearance affinities to obtain the linking probabilities of two tracklets, we separately develop a learning algorithm to automatically learn the weights of the two terms from labeled training data. The learned weights can enhance the tracking cues with strong discriminative power and suppress the tracking cues with weak discriminative power. As a result, the weighted tracking cues can disambiguate the targets’ respective identities better even in situations such as the one depicted in Figure 1.

A typical way of implementing the popular “detect-then-track” paradigm is to track multiple targets frame by frame, which often encounters irrecoverable errors if a target is undetected in one or more successive frames or if two detections are erroneously linked. To overcome this weakness, global trajectory optimization over batches of frames have been proposed in recent years, using methods such as Linear Programming [48], [33] and Dynamic Programming [22], [44]. These methods are often based on graphical network optimization in which the nodes are represented by detection responses. Such methods often fail to handle the problems of long-term tracking in crowded scenes well. To alleviate this, some researchers [34], [57], [46] try to use the track fragments (tracklets) as graph nodes aiming at linking tracklets into long trajectories. This kind of Tracklet Association-based Tracking (TAT) methods can increase robustness and reduce the computation complexity of the graph optimization. There are two key components of a TAT approach: (1) The tracklet affinity model that estimates the likelihood of two tracklets belonging to the same target; (2) The global optimization framework for tracklet association that determines the links of the tracklets based on their affinity scores.

In this paper, we report our algorithm applied to tracking pedestrians in real scenes, but it can be generalized to tracking any other objects in diverse situations. The framework of this approach is shown in Figure 2.

Given a video sequence, we first detect pedestrians in each frame by an existing detector, such as the Deformable Part Models (DPM) detector [20]. We utilize the strategy introduced in Section 3.1 to generate the initial tracklets, which are mostly reliable. But some errors, though very little, could still exist in initial tracklets. We introduce our target-specific metric learning on these initial tracklets. Then we use the online learned target-specific metrics to refine these initial tracklets for reliable tracklets. The cost-flow network is based on the reliable tracklets and its optimization yields the long-term trajectories of multiple persons. Estimating the transition costs is the key factor in the min-cost network flow optimization. We propose to learn tracklet affinity models, which include weighted discriminative appearance and motion cues, in an online manner for estimating the transition costs.

The main contributions of this paper are: (1) Online learning of target-specific metrics with strong discriminative power through a two-step target-specific metric learning and metric refinement processes. (2) Utilizing both appearance and motion dynamics in the tracklet affinity models, which are updated within
each local segment for reduced computation and locally adaptive affinity models. (3) A learning algorithm to learn the weights of motion and appearance tracking cues for tracklet affinity models.

The rest of this paper is organized as follows: Section 2 describes the cost-flow network formulation. Section 3 presents the online learning of the tracklet affinity models. The learning of weights is presented in section 4. Experimental results and comparisons are shown in section 5. Section 6 concludes the paper.

2 COST-FLOW NETWORK FORMULATION FOR TRAJECTORY RECOVERY BY TRACKLET ASSOCIATION

The cost-flow network has been shown to be effective for estimating trajectories in previous studies [62], [9], [44]. However, in these works, the graph nodes are defined by the detection responses. In recent works [23], [47], [54], [34], [57], [56], Tracklet Association-based Tracking (TAT) methods were proposed for multi-target tracking. In these methods, the tracklets were generated based on association of detection responses. However, in these works, the graph nodes are defined by the detection responses. In this paper, we generate the initial tracklets based the successive shortest path algorithm of [44], which will be described in sub-section 3.3. We can construct a smaller graph learned target-specific metrics for reliable tracklets. This tracklet refinement process will be presented in sub-section 3.3. We can construct a smaller graph based on such tracklets which are of a higher order of abstraction than those based on detection responses. The problems in long-term multi-person tracking can be solved by directly linking tracklets instead of detection responses.

An objective function, which takes a similar form as detection association in [62], is defined for tracklet association. Let \( X = \{F_i\} \) be the collection of all the tracklets. A single trajectory hypothesis is defined as an ordered list of \( N \) tracklets: \( T_k = \{F_{k_1}, F_{k_2}, ..., F_{k_l}\} \), where \( F_{k_i} \in X \), and \( i = 1, ..., l; 1 \leq l < N \). A tracklet association hypothesis \( T \) is defined as a set of single trajectory hypotheses: \( T = \{T_k\} \).

The objective of tracklet association is to maximize the posteriori probability of \( T \) given \( X \):

\[
\mathcal{T}^* = \arg \max_T P(T|X) = \arg \max_T P(X|T)P(T) = \arg \max_T \prod_i P(F_i|T)P(T)
\]

assuming that the likelihood probabilities of \( F_i \) are conditionally independent.

We take the assumption that the motion of each tracklet is independent and one tracklet can only belong to one trajectory. Then the above equation can be further decomposed into:

\[
\mathcal{T}^* = \arg \max_T \prod_i P(F_i|T) \prod_{T_k \in T} P(T_k)
\]

subject to Equ. (6), where

\[
c_i^* = -\log P_s(F_i), \quad c_i^t = -\log P_t(F_i), \quad c_{ij} = -\log P(F_j|F_i)
\]

Equ. (7) ensures that the tracklet association hypothesis \( T \) is non-overlapping. The above formulation can be mapped into a cost-flow network with a source \( s \) and a sink \( t \). Estimating the transition costs \( c_{ij} \) is very critical in solving this min-cost network flow problem. Previous network flow approaches [62], [9], [44], [12] only utilize motion cues across consecutive

\[
T = \arg \min_T \left( \sum_i c_i^* f_i^* + \sum_{ij} c_{ij} f_{ij} + \sum_i c_i^t f_i^t \right)
\]

subject to Equ. (6), and

\[
f_i^* + \sum_j f_{ji} = f_i^t + \sum_j f_{ij}
\]
frames and simple appearance features such as color histograms to calculate $c_{ij}$. Nevertheless, these cues are not very reliable when prolonged occlusions and interactions between targets occur. In this paper, we propose to learn the segment-wise tracklet affinity models, consisting of weighted tracking cues, online for estimating $c_{ij}$.

3 Online Learning of Tracklet Affinity Models

In this section, we introduce the online learning of tracklet affinity models, consisting of online target-specific metric learning and online motion dynamics estimation. The affinity scores of adjacent tracklets, which are used as the transition probabilities between two corresponding nodes in the cost-flow network, can be obtained through tracklet affinity measurements. We perform local transition probabilities estimation within a local segment of $S$ frames ($S = 50$ in our implementation).

In order to obtain effective appearance cues for reliable transition probability estimation, we propose a novel target-specific appearance-based model. The appearance-based model learning problem is formulated as a metric learning problem, which can enhance the features with strong discriminative power and suppress the features with weak discriminative power. Here, we learn target-specific metrics so that target-specific properties can be efficiently explored for more discriminative models. In contrast to the previous work of [34] in which local descriptors are learned offline, our learning is online throughout and our target-specific metrics are adaptive to local segments. Moreover, to create a more discriminative tracklet affinity model, we also explore the motion dynamics cue and embed it into the proposed tracklet affinity model. The motion dynamics are online estimated without any assumed priors. As a result, the learned tracklet affinity models can better represent the appearance and motion cues adaptively and provide reliable transition probability estimation.

3.1 Initial Tracklet Generation

Before introducing the proposed tracklet affinity models, we first present the initial tracklet generation process. Similar to the cost-flow network formulation described in previous section, the problem of initial tracklet generation is also formulated as a network flow optimization problem. Different from the cost-flow network formulation presented in Section 2, the graph nodes are defined by the detection responses and their costs, referred as local observation costs here, (the negative logarithm of the corresponding detection scores) are added in the formulation. The mathematical formulation of the minimization problem can be expressed as follows:

$$F = \arg \min \left( \sum_i \hat{c}_i^0 \hat{f}_i^0 + \sum_{ij} \hat{c}_{ij} \hat{f}_{ij} + \sum_i \hat{c}_i \hat{f}_i + \sum_i \hat{c}_i^1 \hat{f}_i^1 \right)$$

s.t. $\hat{f}_{ij}, \hat{f}_i, \hat{f}_i^0, \hat{f}_i^1 \in \{0, 1\}$,
and $\hat{f}_i^0 + \sum_j \hat{f}_{ji} = \hat{f}_i \hat{f}_i = \hat{f}_i^0 + \sum_j \hat{f}_{ij}$

where $\hat{c}_i^0$, $\hat{c}_i$, $\hat{c}_{ij}$, $\hat{c}_i$ denote the starting, termination, transition and local observation costs, respectively. $\hat{f}_i^0$, $\hat{f}_i^1$, $\hat{f}_{ij}$, $\hat{f}_i$ denote the corresponding flows.

To obtain the initial tracklets (relatively short track fragments), the transition cost $\hat{c}_{ij}$ is set to 0 for each candidate detection pair first. Only the detection responses over consecutive frames are considered to be linkable. The transitions of the cost-flow network across non-consecutive frames are not permitted. Moreover, a constraint is imposed on initial tracklet generation. That is, a track fragment should start from a detection response and terminate at a detection response with detection scores higher than a pre-defined threshold, thus only consecutive detection responses with detection scores above the threshold are used to form the initial tracklets. This constraint ensures that the generated initial tracklets are relatively short and mostly reliable track fragments. The dynamic programming algorithm in [44], which approximates the successive shortest path solution, is employed to optimize the cost-flow network. This initial tracklet generation strategy can be viewed as a simplified version of the method in [44]. The average speed of this process is above 500 fps.

3.2 Online Target-Specific Metric Learning

We aim to online learn discriminative target-specific metrics while keeping the computational complexity low. For each tracklet $F_i$, we learn a distance metric function.

The learning involves feature representation, online training sample collection and online training. To create a strong appearance-based model, we start from a rich set of basic features, which includes color, shape and texture, to describe a pedestrian’s appearance.

Given a training dataset $Z = \{z_i\}_{i=1}^{N}$, where $z_i \in \mathbb{R}^{N_d}$ is a feature vector representing the appearance of the image area under the bounding box where there is a strong detection response in tracklet $F_i$ at frame $t$, $N_c$ is the total number of training samples and $N_d$ is the total number of feature dimensions. The training dataset $Z$ is obtained from reliable tracklets through online training sample collection, which will be introduced in latter part of this subsection. We define a positive difference vector $x_i^p$ computed between a positive sample pair (a pair of detection
responses belonging to the same person) and a negative difference vector $x^p_i$ computed from a negative sample pair (a pair of detection responses belonging to different persons). Here, it is assumed that the first $M$ frames of each initial tracklet are reliable and the detection responses are from the same person. Training samples are therefore collected from these frames. The value of $M$ is empirically determined. It is found that a value $M$ between 6 and 10 frames works well for all sequences. Based on the constraint introduced in previous subsection, most of the generated initial tracklets are relatively short and mostly reliable. However, some unreliable tracklets may still exist due to occlusions of the targets. This kind of target interactions often occur in the middle of initial tracklets. Based on this observation, we assume that frames of each initial tracklet are mostly reliable. However, some unreliable tracklets may still exist due to occlusions of the targets. This kind of reliable. Based on this observation, we assume that target interactions often occur in the middle of initial tracklets. Based on this observation, we assume that frames of each initial tracklet are reliable and exist due to occlusions of the targets.

Multiple short but reliable ones.

where $d$ is an absolute difference function, $z_i$ and $z'_i$ are the feature vectors of two samples from the same tracklet $F_i$, $z'_j$ is the feature vector of a sample from a different tracklet $F_j$.

Given the difference vectors $x^p_i$ and $x^n_i$, a distance function $D_i$ for tracklet $F_i$ can be learned based on relative distance comparison so that $D_i(x^p_i) < D_i(x^n_i)$. This distance function $D_i$ is parameterized as a Mahalanobis distance function:

$$D_i(x) = x^TM_i x, \quad M_i \succeq 0$$

where $M_i$ is a positive semidefinite matrix.

We adopt the logistic function as in [63] to learn $D_i$ to force $D_i(x^p_i)$ to be small, and $D_i(x^n_i)$ to be big:

$$\min_{D_i} r(D_i) = -\log \left( \left( 1 + \exp \left( D_i(x^p_i) - D_i(x^n_i) \right) \right)^{-1} \right)$$

Furthermore, the term $M_i$ in the distance function $D_i$ can be decomposed by eigendecomposition:

$$M_i = A_i \Lambda_i A_i^T = W_i W_i^T, \quad W_i = A_i \Lambda_i^{1/2}$$

where $A_i$ is the orthonormal eigenvector matrix of $M_i$ and the diagonal of $\Lambda_i$ are the corresponding eigenvalues.

Therefore, learning a distance function $D_i$ is equivalent to learning the matrix $W_i$ as follows:

$$\min_{W_i} \left\{ r(W_i), \text{s.t.} \quad w^i_i w_j = 0, \forall i \neq j, w_i, w_j \in W_i \right\}$$

$$r(W_i) = \log(1 + \exp \left\{ \left\| W_i^T x^p_i \right\|^2 - \left\| W_i^T x^n_i \right\|^2 \right\})$$

Online training sample collection is another important issue in online learning. The $q$ strongest ($q = 4$ in this work) detection responses in each tracklet are used as training samples. For $x^p_i$, we collect positive sample pairs from the same tracklet. However, for $x^n_i$, we collect negative sample pairs from different persons. To determine the relevance of sample pairs, two constraints are employed: spatio-temporal and exit constraints. The first constraint is based on the fact that one person cannot appear at two or more different locations at the same time. The second constraint is based on the observation that the person who has already exited the view cannot be the person who is still within the view. We online collect negative samples, which satisfy the above two constraints, from $F_i$ and $F_j$ respectively to form negative sample pairs.

Learning $W_i$ using the optimization criterion (14) is a nonconvex optimization problem. In this work, we utilize the optimization algorithm in [63] to learn $W_i$ for each tracklet $F_i$. Finally, we obtain the target-specific transform matrices for all the tracklets:

$$W = \{ W_i \}, \quad i = 1, ..., N$$

### 3.3 Tracklet Refinement

To solve the objective function in Equ. (6), we need to identify reliable tracklets for the nodes in the network graph. The strategy of initial tracklet generation, which is described in previous sub-section, uses spatio-temporal information such as distance between corresponding observations in adjacent frames to link the detections into tracklets. Without effective use of appearance cues, the initial tracklets may be not consistent in appearance and hence unreliable when there are many interactions or occlusions between targets. A typical error is that there are some detection responses belonging to different persons in one tracklet. Hence, tracklet refinement is needed to separate tracklets into multiple short but reliable ones.

The online learned target-specific metrics are employed to refine the initial tracklets. To construct the probe set, the detection with the strongest detection response, $g_i$, is selected from the first $M$ frames of an initial tracklet, $F_i$, which are assumed to be reliable. It is defined as $G = \{ g_i \}, i = 1, ..., N_s$, where $N_s$ is the number of tracklets in a local segment. Each tracklet $F_i$ has only one selected $g_i$ in $G$.

We learn the target-specific transform matrix $W_i$ for each initial tracklet after collecting training samples as described in previous sub-section. The identity test is carried out within a local segment frame by frame to obtain the relative distance between the detection response at frame $t$ of $F_i$ and the corresponding $g_i$ in the probe set:

$$x^p_i = d(z^p - g^i - g^i, i = 1, ..., N_s)$$

$$d^p_i = \| W_i^T x^p_i \|^2$$

where $z^p$ is an instance from tracklet $F_i$ at frame $t$, $g_i$ is the corresponding detection response of $F_i$ in $G$, and $d^p_i$ is the relative distance between $z^p$ and $g_i$.

To be a reliable tracklet, the relative distance between the current detection response $z^p$ and the probe $g_i$ should be small; otherwise, it is an unreliable tracklet. A distance threshold $\omega$ is used to identify
The goal of tracklet dynamics estimation is, given \( y \) where \( a \) is a well known result from the realization theory \([27]\), we split \( F \) into two parts from the first consecutive detection responses having relative distance values (from \( g \)) above \( \omega \), and then split \( F \) into two parts from the first consecutive detection response. In virtue of the strategy of initial tracklet generation introduced in Section 3.1, the generated initial tracklets are mostly reliable, but some errors, though very little, could still exist in the initial tracklets. This tracklet refinement process is usually repeated no more than twice to obtain reasonably reliable tracklets.

After obtaining the reliable tracklets, the target-specific metrics learned from the initial tracklets are updated. Different from the training sample collection of initial tracklets that the samples are collected from the first \( M \) frames of the tracklets, the training samples of the reliable tracklets are collected from the full-length tracklets. The strategy of online training sample collection is the same as the one introduced in Section 3.2. As shown in Figure 2, a two-step target-specific metric learning/update is used in the proposed framework. The first step is used for tracklet refinement, which is usually repeated no more than two times. The second step is used for the appearance-based tracklet affinity estimation, which executes only once.

### 3.4 Online Tracklet Dynamics Estimation

To disambiguate targets with similar appearance, we propose to exploit motion dynamics together with appearance cues as described above to keep track of target’s identity. The main idea of tracklet dynamics estimation is to model the evolution of target motions as a sequence of piecewise linear regressors whose orders can be estimated from available data.

The dynamics of a tracklet can be constructed as an ordered sequence of dynamic measurements \( \{y_q\}_q \), \( s \leq q \leq e \), where \( s \) and \( e \) are the starting and ending frames, respectively. Similar to \([13], [17]\), we collect the position information of all the detection responses within one tracklet in a vector \( y \) and assume that its value at current time \( q \) is related to its past values \( y_{q-i} \) by an \( m \)th order autoregressive model of the form:

\[
y_q = a_1y_{q-1} + a_2y_{q-2} + \ldots + a_my_{q-m} = \sum_{i=1}^{m} a_iy_{q-i}, \quad m \leq N_f, q \geq s + m \tag{17}
\]

where \( a = [a_1 a_2 \ldots a_m]^T \) is the regressor vector, \( N_f \) is the total number of frames of the tracklet, \( m \) is the number of frames of the dynamic measurement \( y \), and \( s \) is the starting frame of the tracklet.

The order of the autoregressive model \( m \) measures the complexity of the underlying tracklet dynamics. The goal of tracklet dynamics estimation is, given dynamic measurements \( \{y_q\}_q \), to estimate the minimum \( m \) such that the model \([17]\) retains. Specifically, a well known result from the realization theory \([27]\) is that, under mild conditions, given an ordered sequence of measurements \( \{y_q\}_q \) generated by Eq. \([17]\), the order \( m \) of the autoregressive model equals the rank of the corresponding Hankel matrix, i.e., \( m = \text{rank}(H_{F_i}) \) where \( H_{F_i} \) is the Hankel matrix with \( n \geq m \) columns:

\[
H_{F_i} = \begin{bmatrix} y_{s} & y_{s+1} & \ldots & y_{s+n-1} \\ y_{s+1} & y_{s+2} & \ldots & y_{s+n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{t-n+1} & y_{t-n} & \ldots & y_t \end{bmatrix} \tag{18}
\]

where \( n \) is defined based on the length of the tracklet:

\[
n = l_i - \lfloor l_i/3 \rfloor + 1; \quad l_i = t - s + 1 \tag{19}
\]

where \( l_i \) is the length of tracklet \( F_i \).

The motion dynamics similarity \( P_m(F_i, F_j) \) between two tracklets \( F_i \) and \( F_j \), which takes a similar form as in \([17]\), is defined as follows:

\[
P_m(F_i, F_j) = \begin{cases} -\infty, & \text{if temporal conflict exists;} \\ \frac{\text{rank}(H_{F_i}) + \text{rank}(H_{F_j})}{\text{rank}(H_{F_{ij}})} - 1, & \text{otherwise.} \end{cases} \tag{20}
\]

where \( F_{ij} = [F_i \alpha_{ij}^j F_j] \) is the joint tracklet with the gap \( \alpha_{ij}^j \) between \( F_i \) and \( F_j \) interpolated. The joint Hankel matrix \( H_{F_{ij}} \) is formed by combining the dynamic measurements of \( F_i \) and \( F_j \) and the interpolated data.

Here, we take the assumption that the targets do not significantly change their dynamics between tracklets. The intuition of the above motion dynamics similarity is that if two tracklets are from the same trajectory then they can be approximated by one relatively low order regressor. Otherwise, if two tracklets are from different trajectories, the joined trajectory needs a higher order regressor than the regressors of each single tracklet. Hence, if \( \text{rank}(H_{F_{ij}}) = r(F_i) \) and \( \text{rank}(H_{F_{ij}}) = r(F_j) \), then \( \text{rank}(H_{F_{ij}}) = r(F_{ij}) \leq (r(F_i) + r(F_j)) \). Consequently, if \( F_i \) and \( F_j \) are of the same trajectory, then \( r(F_i) = r(F_j) = r(F_{ij}) \) and \( P_m(F_i, F_j) = 1 \), but if not, \( P_m(F_i, F_j) \approx 0 \).

Tracklet dynamics are online estimated, without any prior knowledge, based on the reliable tracklets. The IHTLS (Iterative Hankel Total Least Squares) method in \([16]\) is employed to estimate the rank of the Hankel matrices for tracklet dynamics estimation. The computational complexity of rank estimation is \( O((l_i - m)m^2) \), where \( l_i \) is the length of a tracklet \( F_i \) and \( m \ll l_i \) is the rank of the matrix.

### 3.5 Tracklet Affinity Measurement

In this subsection, we present the measurement of the affinity between \( F_i \) and \( F_j \), or equivalently, the transition probability, \( P_{ij} \), in the network graph between node \( i \) and node \( j \). The tracklet affinity score, \( S_{ij} \), which is equivalent to \( P_{ij} \), is defined as follows:

\[
S_{ij} = P_m(F_i, F_j)P_n(F_i, F_j)C_{ij} \tag{21}
\]
where \( P_m(F_i, F_j) \) is the motion-based affinity model, which is defined by Eqn. (23). \( P_a(F_i, F_j) \) is the appearance-based affinity model and \( C_{ij} \) is a limiting function.

To obtain the appearance-based affinity model \( P_a(F_i, F_j) \), we first compute the relative distances \( d_{ij} \) between each detection response in \( F_i \) and the probe \( g_i \), and \( d'_{ij} \) between each detection response in \( F_j \) and the probe \( g_i \):

\[
x_{ij} = |z_i^t - g_j, x'_{ij} = |z_j'^t - g_i|; i, j = 1, ..., N_s
\]

\[
d_{ij} = \|W^T_i x_{ij}\|^2, d'_{ij} = \|W^T_j x'_{ij}\|^2
\]

where \( z_i^t \) denotes the feature vector of a detection response in tracklet \( F_i \) at frame \( t \), \( z_j'^t \) denotes the feature vector of a detection response in tracklet \( F_j \) at frame \( t' \), and \( g_i, g_j \in G \).

Subsequently, we calculate the mean values of the relative distances and use them to define the appearance-based affinity model \( P_a(F_i, F_j) \):

\[
d_{ij} = (\sum d_{ij})/m, d'_{ij} = (\sum d'_{ij})/n
\]

\[
P_a(F_i, F_j) = (d_{ij}d'_{ji})^{-1}\gamma
\]

where \( \gamma \) is a normalization term and \( m, n \) are the number of frames of \( F_i \) and \( F_j \) respectively.

We do not have to apply Eqn. (21) to every pair, since there are a lot of obviously non-related tracklet pairs which do not belong to the same trajectory. Because a limiting function \( C_{ij} \) is included in Eqn. (21), we actually apply it to every tracklet pair. This limiting function \( C_{ij} \) is proposed based on spatio-temporal, and exit constraints:

\[
C_{ij} = C_t(F_i, F_j)C_e(F_i, F_j)
\]

The spatio-temporal constraint is defined as follows:

\[
C_t(F_i, F_j) = \begin{cases} 1, & \text{if } F_i \cap F_j = \phi \\ 0, & \text{otherwise} \end{cases}
\]

where \( \cap \) is an intersection operator that is used to find the overlap between two tracklets and \( \phi \) is the empty set.

The exit constraint is defined based on the observation that the person who has already exited the scene cannot be the person who is still within the scene:

\[
C_e(F_i, F_j) = \begin{cases} 1, & \text{if } t_s^i > t_e^j \text{ and } p_j^t \notin E \\ 0, & \text{otherwise} \end{cases}
\]

where \( t_s^i \) is the starting frame of tracklet \( F_i \), \( t_e^j \) is the ending frame of tracklet \( F_j \), \( p_j^t \) is the position of the detection response of tracklet \( F_j \) at time \( t_j^t \) and \( E \) is the exit area which is near image borders. For static cameras, we adopt the incremental learning algorithm for exit map as in [56] to obtain \( E \).

\( C_t(F_i, F_j) \) and \( C_e(F_i, F_j) \) associate \( F_i \) and \( F_j \) if they have no overlap and \( F_j \) does not exit the screen when \( F_j \) appears.

The transition costs of the adjacent nodes in the cost-flow network is obtained by taking negative logarithm of the affinity scores between corresponding tracklets:

\[
c_{ij} = -\log S_{ij}
\]

Finally, we can estimate the optimal tracklet association hypothesis \( T^* \) in Eqn. (6) based on \( c_{ij} \). Although the transition costs \( c_{ij} \) are estimated within local segments, the final tracking trajectories are obtained through network flow optimization on the whole sequence. After tracklet association, there may exist some gaps between adjacent tracklets in each trajectory due to missed detections and occlusions. The possible gaps in the tracking trajectories are interpolated linearly.

However, in the proposed tracklet affinity model as depicted in Eqn. (21), the motion-based affinity model, \( P_m(F_i, F_j) \), and appearance-based affinity model, \( P_a(F_i, F_j) \), are treated equally without any voting weights. This may result in inaccurate affinity scores if one of the tracking cues is dominant and the other one is confusing, having an effect as a noise factor. Therefore, we investigate the influences of the two tracking cues on tracking performance in next section.

### 4 Learning of Affinity Weights

In difficult situations, where severe occlusions and interactions occur, the motion-based affinity model and appearance-based affinity model may not be consistent. Hence, we need to weight them properly for stable performance.

We propose to add a weighting parameter \( \lambda \), which controls the weight of the motion-based affinity score, in Eqn. (21).

\[
S_{ij} = [P_m(F_i, F_j)]^\lambda P_a(F_i, F_j)C_{ij}, 0 \leq \lambda \leq 1
\]

where \( \lambda \) is learned from labeled data. If the value of \( \lambda \) is larger, the motion-based affinity score, \( P_m(F_i, F_j) \), contributes more to \( S_{ij} \).

#### 4.1 Assessment of Difficult Situations for Motion Dynamics

To investigate the difficult situations where the motion affinities are not reliable, we only utilize the motion-based affinity model to estimate \( c_{ij} \) for Eqn. (6) in the experiments. After analyzing the inconsistencies between the tracking results and the labeled ground truth data, we obtain these typical situations where the motion affinities are not reliable. Based on the analysis, we design a rule to automatically assess these difficult situations.

There are two constraints in this rule. The first one is that two tracklets should have a certain amount
of overlap in the starting or ending frames, which indicates the occlusion of two targets:

\[ S(z_i^t) \cap S(z_i^{t^*}) \geq \eta \min(S(z_i^t), S(z_i^{t^*})) \]  

\[ S(z_i^t) \cap S(z_i^{t^*}) \geq \eta \min(S(z_i^t), S(z_i^{t^*})) \]  

(30)

where \( z_i^t \) and \( z_i^{t^*} \) are the detection responses of \( F_i \) at the ending frame \( t^* \) and the starting frame \( t^* \), respectively. \( z_k^t \) and \( z_k^{t^*} \) are the detection responses of \( F_k \) at the ending frame \( t^* \) and the starting frame \( t^* \), respectively. \( S(\cdot) \) is the operator to capture the area of the detections in pixels. \( \eta \) is a sensitivity threshold (\( \eta = 0.3 \) in our implementation).

The second constraint is that tracklets \( F_i \) and \( F_j \) must have a gap for it to be linkable. That is,

\[ t_j^t - t_i^e > 1, \quad i \neq j \]  

(31)

where \( t_j^t \) is the starting frame of tracklet \( F_j \), \( t_j^e \) is the ending frame of tracklet \( F_j \), and \( t_i^e > t_i^e \).

If the tracklet pair \( \{ F_i, F_k \} \) match the above two conditions, then we add the weighting parameter \( \lambda \) to the motion-based affinity models related to \( F_i \) and \( F_k \) as \([P_m(F_i, F_j)]^h \) and \([P_m(F_k, F_j)]^h \). \( F_j \) and \( F_k \) are the linked candidate tracklets of \( F_i \) and \( F_k \) respectively. As we can see in Figure 3 the tracklet pairs with identities 269, 289 and 305, 306 match these two constraints. In such situation, due to interactions of the two targets, the two corresponding trajectories become ambiguous. Hence, we add the weighting parameter \( \lambda \) for these 4 tracklets in terms of the motion-based affinity models.

![Fig. 3: An example of the difficult situations.](image)

**4.2 Learning of the Weighting Parameter**

The weighting parameter \( \lambda \) in Equ. (29) defines the weight of the motion-based affinity model for tracklet association. Based on the number of frames of the gaps between tracklets, we divide \( \lambda \) into 2 levels:

\[ \lambda = \begin{cases} 
\lambda_1, & 1 \leq u \leq B_1 \\
\lambda_2, & u > B_1 
\end{cases} \]  

(32)

where \( u \) is the number of frames in the gap between corresponding tracklets (\( B_1 = 20 \) in our implementation).

**Algorithm 1: Weighting parameter learning for tracklet association**

**Input:**
- Reliable tracklets;
- Labeled ground truth data;

**Output:**
- The learned weighting parameters: \( \{ \lambda_1, \lambda_2 \} \);

1. Initialize the weighting parameters: \( \lambda_1 = \lambda_1^0 = 0, \lambda_2 = \lambda_2^0 = 0 \) and the step value \( \Delta \lambda = 0.1 \);
2. Online estimate transition costs for all graph node (tracklet) pairs based on:

\[ c_{ij} = - \log([P_m(F_i, F_j)]^h P_s(F_i, F_j) P_c(i, j)) \]

3. for \( i = 1 \) to \( 2 \) do
4. while \( \lambda_i^1 \leq 1 \) do
5. for all graph node (tracklet) pair \( \{ F_i, F_k \} \) matches the rule of the assessment of difficult situations do
6. if \( u \geq 1 \) and \( u \leq B_1 \) then
7. \( \lambda = \lambda_1 \);
8. else if \( u > B_1 \) then
9. \( \lambda = \lambda_2 \);
10. end if
11. for all tracklet pairs related to \( F_i, F_k \) do
12. \( c_{ij} = - \log([P_m(F_i, F_j)]^h P_s(F_i, F_j) C_{ij}) \)
13. \( c_{ik} = - \log([P_m(F_k, F_i)]^h P_s(F_k, F_i) C_{ik}) \)
14. end for
15. end for
16. Obtain tracking results through network flow optimization;
17. if Current tracking results better converge to ground truth data then
18. \( \lambda_i = \lambda_i^1 \);
19. end if
20. \( \lambda_i^1 = \lambda_i^1 + \Delta \lambda_i \)
21. end while
22. return \( \{ \lambda_1, \lambda_2 \} \).

The intuition is that if the gaps are longer, it is more difficult to accurately estimate the joint tracklet dynamics. Therefore, we define 2 difficulty levels as in Equ. (32), making the weighting parameter \( \lambda \) adaptive to the difficult situations. The number of difficulty levels and the upper bound value of level 1 (\( B_1 \)) are empirically determined. Furthermore, we employ tracking performance evaluation and network flow optimization jointly to optimize the weighting parameters for tracklet association.

The learning algorithm is summarized in Algorithm 1.

Given the reliable tracklets and the labeled ground truth data of a video sequence, we aim to learn the weighting parameters in a supervised manner so that the tracking performance can be optimized. The two proposed weighting parameters are optimized independently in a greedy fashion. The weighting parameters are initialized as \( \lambda_1 = 0, \lambda_2 = 0 \). After some iterations with a fixed step value, the weighting parameters \( \{ \lambda_1, \lambda_2 \} \) are obtained.

The weighting parameters \( \lambda_1 \) and \( \lambda_2 \) are learned from the ground truth data of PETS 2009 [21] in this paper. Our tracking algorithm is then run with the learned weighting parameters (\( \lambda_1 = 0.5, \lambda_2 = 0.2 \)) on all the datasets for evaluation. Based on the analysis of the experimental results with different \( B_1 \), we find that the tracking performance is slightly affected by the changes of \( B_1 \). An upper bound value \( B_1 \) of (20-30) frame gap works well for all sequences.
5 Experiments

5.1 Datasets

To evaluate the performance of the proposed approach, we experiment on five challenging, publicly available pedestrian datasets.

TUD. The TUD Crossing sequence 3 and TUD-Stadtmitte sequence 4 are real-world videos filmed in busy pedestrian streets. The cameras are positioned at a quite low angle, resulting in more complex occlusion patterns and rather inaccurate ground plane locations. Furthermore, for TUD-Stadtmitte, the size of the pedestrians on the image plane vary drastically.

PETS 2009. This benchmark dataset 21 presents an outdoor scene with large number of pedestrians captured from multiple cameras at 7 fps. The pedestrians vary significantly in appearance due to shadows and lighting changes. Moreover, there are frequent occlusions, caused by pedestrian occluding each other, or static occlusions such as the traffic sign. In the experiments, we use the sequences S2L1 and S2L2 in the first view, which are widely used in literature.

Town Centre. The Town Centre dataset 8 is captured by a single elevated camera in a busy street. There are 16 pedestrians visible at any time on average, leading to frequent dynamic occlusions and interactions. Furthermore, due to the severe occlusions caused by static obstacles, many pedestrians are not detected by the state-of-the-art detectors.

ETH. The ETH BAHNHOF and SUNNY DAY sequences 18 show busy street scenes from a pair of cameras on a moving stroller. The stroller is moving forward at most of the time, however there are still some panning motions, which leads to the unreliable motion affinities between tracklets. Moreover, frequent full or partial occlusions occur due to the low view angles of cameras. The size of the pedestrians also varies significantly on the image plane.

MOTChallenge. The MOTChallenge 2D Benchmark is an up-to-date multiple object tracking benchmark. It consists of a total of 22 sequences, in which half of them are used for training and half of them are used for testing. The test sequences cover many different situations, such as different viewpoints, static or moving camera, different weather conditions. This makes MOTChallenge benchmark very challenging.

5.2 Experimental Settings

The online collected training samples from video frames are normalized to $128 \times 64$ pixels for target-specific metric learning. For the color feature, RGB, YCbCr and HSV color histograms are extracted with 16 bins for each channel respectively and concatenated into a 144-element vector. To capture shape information, we adopt the Histogram of Gradients (HOG) feature 15 by setting the cell size to be 8 to form a 3968-element vector. Two types of texture features are extracted by Schmid and Gabor filters. In total, 13 Schmid channel features and 8 Gabor channel features are obtained to form a 336-element vector by using a 16-bin histogram vector to represent each channel. Each person image is thus represented by a feature vector in a 4448-dimensional feature space.

5.3 Evaluation Metrics

Since it is difficult to use one single score to evaluate multi-target tracking performance, we utilize the evaluation metrics defined in 38, 12, as well as the standard CLEAR MOT metrics 10:

- MOTA (↑): Multi-object tracking accuracy.
- MOTP (↑): Multi-object tracking precision.
- Recall (↑): correctly matched detections / total detections in ground truth.
- Precision (↑): correctly matched detections / total detections in the tracking result.
- FAF (↓): number of false alarms per frame.
- GT: number of trajectories in ground truth.
- MT (↑): number of mostly tracked trajectories.
- PT: number of partially tracked trajectories.
- ML (↓): number of mostly lost trajectories.
- Frag (↓): number of fragmentations.
- IDS (↓): number of id switches.
- IDS (correctly matched detections (↓)).

For evaluation measures with (↑), higher scores denote better performance; for evaluation measures with (↓), lower scores denote better performance. A tracking bounding box in the result having more than 50% overlap with the corresponding groundtruth bounding box is considered as true positive. The evaluation codes are downloaded from 39.

5.4 Influence of Tracklet Refinement

To investigate the influence of the repetitions of tracklet refinement process, we run our tracking algorithm and experimented with different number of repetitions while keeping all the other conditions fixed. As shown in Figure 4 for each dataset, the changes in performance (measured by MOTA score) is plotted against the repetitions of tracklet refinement process. 0 repetitions in Figure 4 means that no tracklet refinement is utilized for our tracking algorithm. Note that the tracking performance has been improved by exploiting the tracklet refinement for all datasets. Moreover, it is found that the optimal tracking performance is achieved by no more than 2 repetitions of tracklet refinement process for the four exemplary datasets. From the statistics as shown in Figure 4 we can conclude that the proposed tracking algorithm with only one time tracklet refinement process can achieve near-optimal tracking performance.

5.5 Quantitative Evaluation

The quantitative evaluations are presented in three sub-sections: abbreviations of the proposed methods
in the experiments, comparison with network flow based methods, and comparison with other state-of-the-art methods on benchmark datasets. The tracking results of other methods are extracted from the published papers and the MOTChallenge benchmark website [1] for the ease of reference.

5.5.1 Abbreviations of Different Methods
- CML: The proposed method with only an online learned common class metric for all tracklets.
- TSML: The proposed method with online target-specific metric learning.
- TD: The proposed method with only tracklet dynamics.
- TSML+TD: The proposed method with online target-specific metric learning and tracklet dynamics.
- TSML+TD+WP: The proposed method with full tracklet affinity model including target-specific metric learning (TSML), tracklet dynamics (TD) and weighting parameters (WP).

CML and TSML are from our previous work [50].

5.5.2 Comparison with Network Flow based Methods
We first evaluate our method on the popular TUD Crossing sequence [3] and ETH BAHNHOF sequence [18]. For a fair comparison, we use the same sequences and pre-trained pedestrian detector as used in [12]. The quantitative metric that we use is ID switches / total number of correct observations used in the trajectories (IDS / correctly matched detections), which is the same as in [12]. Table 1 gives the quantitative results computed on the TUD Crossing sequence, and the first 350 frames of the ETH BAHNHOF sequence. Due to the forward and panning motions of the cameras, the motion affinities between tracklets are unreliable for the BAHNHOF sequence. Therefore, we do not add the online tracklet dynamics estimation for the experiments in this subsection. We report the results of five methods: DP algorithm of [44] is our baseline work without adding the online learning of tracklet affinity models. The results are also compared to those of other network flow based methods: MCNF [62] and LRMCNF [12]. The proposed method with online target-specific metric learning is denoted as TSML. Furthermore, we also report the results of the proposed method with only a common class metric for all tracklets, which is denoted as CML.

Note that our method gives better results when compared with the three network flow methods [44], [62], [12]. Moreover, the noticeable improvement in ID switches indicates that our method can better deal with long-term tracking, where the traditional motion models are less reliable.

5.5.3 Comparison with State-of-the-art Methods
To show the effectiveness of our method, we further compare our method with other state-of-the-art methods on more publicly available datasets. We use the pre-trained human detector of [19] to generate the detections. For the MOTChallenge benchmark [36], we utilize DPM detections [19] and the public detections from this benchmark for the evaluation. In the result tables, the numbers ranked in the first place of the respective evaluation measures are marked in bold.

**TABLE 1: Comparison of tracking results with network flow based methods on TUD Crossing and ETH BAHNHOF (first 350 frames) sequences.**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>TUD Crossing</th>
<th>ETH</th>
<th>ETH (GT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF [44]</td>
<td>32/768</td>
<td>37/1387</td>
<td>25/1648</td>
</tr>
<tr>
<td>LRMCNF [12]</td>
<td>14/819</td>
<td>23/1514</td>
<td>14/1783</td>
</tr>
<tr>
<td>CML</td>
<td>10/845</td>
<td>5/1728</td>
<td>3/1786</td>
</tr>
<tr>
<td>TSML</td>
<td>7/862</td>
<td>17/1990</td>
<td>0/1820</td>
</tr>
</tbody>
</table>

**PETS2009-S2L1.** For a fair comparison, we utilize the same ground truth as in [40] for the experiments, in which all the occurring pedestrians have been annotated. The quantitative results are shown in Table 2. As expected, taking tracklet dynamics into account increases the overall tracking performance. Our full tracklet affinity model (TSML+TD+WP) further raises the MOTA by 0.6% and reduces the ID switches by ≈ 43%. This indicates that our method with full tracklet affinity model combines motion and appearance cues properly, resulting in further improvement on tracking performance. On the whole, our method with full tracklet affinity model achieves the best performance compared with 16 state-of-the-art methods in terms of MOTP, Precision, FAF, ML, and IDS. For other evaluation measures, our approach also achieves comparative performance.
TABLE 2: Comparison of tracking results between state-of-the-art methods and ours on PETS2009-S2L1.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA</th>
<th>MOTP</th>
<th>Recall</th>
<th>Precision</th>
<th>FAP</th>
<th>GT</th>
<th>MT</th>
<th>PT</th>
<th>ML</th>
<th>Frag</th>
<th>IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMAT</td>
<td>95.9%</td>
<td>78.2%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>82.6%</td>
<td>17.4%</td>
<td>0.0%</td>
<td>21</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>KSP</td>
<td>80.3%</td>
<td>72.0%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>73.9%</td>
<td>17.4%</td>
<td>8.7%</td>
<td>22</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>MTMM [26]</td>
<td>82.3%</td>
<td>71.1%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>89.5%</td>
<td>10.5%</td>
<td>0.0%</td>
<td>45</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>UHMTGDA [28]</td>
<td>97.8%</td>
<td>75.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>8.8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HJMRMT [29]</td>
<td>98.0%</td>
<td>82.8%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>11</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(MP)^2T [32]</td>
<td>90.7%</td>
<td>76.0%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>DTLLE Tracking [41]</td>
<td>90.3%</td>
<td>74.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>78.3%</td>
<td>21.7%</td>
<td>0.0%</td>
<td>15</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>CEMMIT [40]</td>
<td>90.6%</td>
<td>80.2%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>91.3%</td>
<td>4.4%</td>
<td>4.3%</td>
<td>6</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>GMC-Tracker [61]</td>
<td>90.3%</td>
<td>69.0%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>89.5%</td>
<td>10.5%</td>
<td>0.0%</td>
<td>54</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>OMTD [43]</td>
<td>79.7%</td>
<td>56.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>OMAT [45]</td>
<td>92.8%</td>
<td>74.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>PMPTCS [59]</td>
<td>76.0%</td>
<td>53.8%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>OGOMT [45]</td>
<td>98.1%</td>
<td>80.5%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>16</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>CSL-VOX [14]</td>
<td>89.78%</td>
<td>-</td>
<td>98.28%</td>
<td>91.07%</td>
<td>0.14</td>
<td>19</td>
<td>94.7%</td>
<td>5.3%</td>
<td>0.0%</td>
<td>26</td>
<td>28</td>
</tr>
<tr>
<td>CSL-DPT [14]</td>
<td>88.13%</td>
<td>-</td>
<td>97.64%</td>
<td>90.45%</td>
<td>-</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 3: Comparison of tracking results between state-of-the-art methods and ours on Town Centre dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA</th>
<th>MOTP</th>
<th>Recall</th>
<th>Precision</th>
<th>FAP</th>
<th>GT</th>
<th>MT</th>
<th>PT</th>
<th>ML</th>
<th>Frag</th>
<th>IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Minimization</td>
<td>81.4%</td>
<td>76.1%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>82.6%</td>
<td>17.4%</td>
<td>0.0%</td>
<td>21</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>DC Tracking</td>
<td>95.9%</td>
<td>78.2%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>KSP</td>
<td>80.3%</td>
<td>72.0%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>73.9%</td>
<td>17.4%</td>
<td>8.7%</td>
<td>22</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>MTMM [26]</td>
<td>82.3%</td>
<td>71.1%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>89.5%</td>
<td>10.5%</td>
<td>0.0%</td>
<td>45</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>UHMTGDA [28]</td>
<td>97.8%</td>
<td>75.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>8.8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HJMRMT [29]</td>
<td>98.0%</td>
<td>82.8%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>11</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(MP)^2T [32]</td>
<td>90.7%</td>
<td>76.0%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>DTLLE Tracking [41]</td>
<td>90.3%</td>
<td>74.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>78.3%</td>
<td>21.7%</td>
<td>0.0%</td>
<td>15</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>CEMMIT [40]</td>
<td>90.6%</td>
<td>80.2%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>91.3%</td>
<td>4.4%</td>
<td>4.3%</td>
<td>6</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>GMC-Tracker [61]</td>
<td>90.3%</td>
<td>69.0%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>89.5%</td>
<td>10.5%</td>
<td>0.0%</td>
<td>54</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>OMTD [43]</td>
<td>79.7%</td>
<td>56.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>OMAT [45]</td>
<td>92.8%</td>
<td>74.3%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>PMPTCS [59]</td>
<td>76.0%</td>
<td>53.8%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>OGOMT [45]</td>
<td>98.1%</td>
<td>80.5%</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>16</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>CSL-VOX [14]</td>
<td>89.78%</td>
<td>-</td>
<td>98.28%</td>
<td>91.07%</td>
<td>0.14</td>
<td>19</td>
<td>94.7%</td>
<td>5.3%</td>
<td>0.0%</td>
<td>26</td>
<td>28</td>
</tr>
<tr>
<td>CSL-DPT [14]</td>
<td>88.13%</td>
<td>-</td>
<td>97.64%</td>
<td>90.45%</td>
<td>-</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

**Town Centre.** To show the generality of the learned weighting parameters, we evaluate our approach with the learned weighting parameters on Town Centre dataset. The ground truth we used here is provided by [8], which is the same as in the compared methods. Due to severe occlusions caused by static obstacles (such as benches) and more frequent dynamic interactions between pedestrians, many pedestrians cannot be detected by the state-of-the-art detectors. Hence, the Recall (as shown in Table 3) is lower than the other datasets. As we can see in Table 3, the full tracklet affinity model with the weighting parameters achieves better or nearly the same performances on all evaluation items. Compared with the tracklet affinity model without weighting parameters, the MOTA is improved by about 8.4%; recall and precision are improved by about 1.6% and 3.4% respectively; fragments and ID switches are reduced by 23.6% and 24.3% respectively. The obvious improvements in performance indicate that the learned weighting parameters are applicable to new data.

**TUD-Stadtmitte.** To make a fair comparison, the experiments are conducted using the same ground truth as defined in [57]. The quantitative tracking results are shown in Table 4. Though there are occlusions and interactions between pedestrians, the number of pedestrians appearing in the scene is less than the other datasets. Hence, our method can generate better optimal tracking results on TUD-Stadtmitte than other datasets. Note that our proposed method with target-specific metric learning (TSML) has achieved very good performance. We also provide the tracking results of the proposed method with TSML+TD and TSML+TD+WP, which show the same optimal tracking results as shown in Table 4. Compared with [34], [57], the improvement is obvious for some metrics. Our method achieves the highest recall and the mostly tracked score (MT) among all the methods. It also achieves the lowest ID switches. Meanwhile, our method achieves competitive performance on precision, false alarms per frame and fragments compared with [34], [57].

**ETH.** To see the effectiveness of the proposed method, we further evaluate it on the challenging ETH dataset [18]. Due to the unreliable motion affinities between tracklets of this dataset, we use the tracklet affinity model without TD and WP for the experiments. For a fair comparison, we use the ground truth provided by [57]. The quantitative tracking results are shown in Table 5. We can see that our method can achieve better or competitive performance on all the commonly used evaluation measures. Compared with
TABLE 4: Comparison of tracking results between state-of-the-art methods and ours on TUD Stadmitte dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA</th>
<th>MOTP</th>
<th>Recall</th>
<th>Precision</th>
<th>FAP</th>
<th>GT</th>
<th>MT</th>
<th>PT</th>
<th>ML</th>
<th>Frag</th>
<th>IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CML</td>
<td>94.5%</td>
<td>72.7%</td>
<td>95.1%</td>
<td>99.4%</td>
<td>0.030</td>
<td>10</td>
<td>100%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>TSLM</td>
<td>97.3%</td>
<td>71.3%</td>
<td>98.0%</td>
<td>99.3%</td>
<td>0.040</td>
<td>10</td>
<td>100%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>TSLM+TD</td>
<td>97.3%</td>
<td>71.3%</td>
<td>98.0%</td>
<td>99.3%</td>
<td>0.040</td>
<td>10</td>
<td>100%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>TSLM+TD+WP</td>
<td>97.3%</td>
<td>71.3%</td>
<td>98.0%</td>
<td>99.3%</td>
<td>0.040</td>
<td>10</td>
<td>100%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

5.6 Computational Speed

The computation speed depends on the number of targets in a video sequence. Our method is implemented using MATLAB on a 3.3 GHz, 4 core PC with 8 GB memory. The speed of the proposed method with target-specific metric learning (TSLM) is about 13, 6, 10 and 9 fps for PETS 2009, Town Centre, TUD and ETH datasets, respectively, excluding the detection time; for PETS 2009 and Town Centre datasets, the speed of the proposed method with our full tracklet affinity model is 11 and 5 fps respectively. The average speed of the proposed method on MOTChallenge 2D Benchmark is about 7 fps. The online learning of tracklet affinity models is the most time consuming part of our method, which takes up about 90% of the total computation time. The breakdown is as follows: the learning of appearance-based tracklet affinity model and motion dynamics estimation take up about 80% and 10% respectively. Speed-up can be achieved by parallel implementations of the online learning of target-specific metrics. Furthermore, the learning of appearance-based tracklet affinity model and motion dynamics estimation can also be implemented in parallel.

6 Conclusion

We have presented our method developed for tracking multiple objects of interest in the scene over a longer period with the aim to maintain consistent tracking and tagging of objects, reducing identity switches. Our method processes the initial tracklets (track fragments) produced by a simple trajectory based tracking algorithm. We propose a two-step online target-specific metric learning to improve the similarity measure based on the appearance cues, and together with coherent dynamics estimation for tracklets based on the motion cues, we establish our new affinity model. The tracking of objects is accomplished by performing tracklet association with network flow optimization where the nodes in the network are tracklets. Thus, our proposed method exploiting both appearance and motion cues is capable to prevent identity switches during tracking and recover missed detections. Our method is found to be effective even when the appearance or motion cues fail to identify or follow the target due to occlusions or object-to-object interactions. To further improve our method, we also propose to learn the weights of these two tracking cues in our affinity model. Our tracking algorithm has been validated on several public datasets and the experimental results show that it outperforms several state-of-the-art tracking algorithms.


