Online Social Behavior Modeling for Multi-Target Tracking

Shu Zhang[1] Abir Das[1] Chong Ding[2] Amit K. Roy-Chowdhury[1]
University of California, Riverside, CA 92521 USA
[1]{szhang,adas,amitrc}@ee.ucr.edu [2]cding@cs.ucr.edu

Abstract

People are often seen together. We use this simple observation to provide crucial additional information and increase the robustness of a video tracker. The goal of this paper is to show how, in situations where offline training data is not available, a social behavior model (SBM) can be inferred online and then integrated within the tracking algorithm. We start with tracklets (short term confident tracks) obtained using an existing tracker. The SBM, a graphical model, captures the spatio-temporal relationships between the tracklets and is learned online from the video. The final probability of association between the tracklets is obtained by a combination of individual target characteristics (e.g., their appearance), as well as the learned relationship model between them. The entire system is causal whereby the results at any given time depend only upon the part of the video already observed. Experimental results on three state-of-the-art datasets show that, without having access to any offline training data or the entire test video a priori (conditions that may be restrictive for many application domains), our proposed method obtains results similar to those that do impose the above conditions.

1. Introduction

Robust multi-target tracking is a fundamental task for automated video content analysis and remains a challenging problem. A popular recent approach has been to generate tracklets (short-term tracks), which can be done reliably using many existing tracking methods, and then computing associations between them to obtain longer tracks [10, 22, 25, 27]. Though great progress has been made, targets with similar appearance or under high clutter still limit the performance of current tracking systems. Since groups of people often walk together and affect each others behavior (e.g., two people walking together can be expected to be seen together in the near future), robust tracking schemes should consider this aspect. This requires development of mathematical models that represent the social interactions between people and incorporation of these models into the tracklet association schemes. Similar to [7, 19], we term such models as "social behavior models" (SBMs).

In many application domains, the tracker needs to work without the advantage of having seen videos captured under similar circumstances from which the SBMs can be learned a priori (as in [15, 21, 26]). This implies that the SBMs need to be learned online. In this paper, we focus on the problem of obtaining robust, long-term tracks by exploiting the behaviors between the individual targets in situations where the above-mentioned constraints are imposed. We learn the parameters of the SBM model online, and then integrate it within the scheme for tracklet association. Our proposed method is causal unlike [18, 25] which are batch processing approaches (they tradeoff causality for improved performance, which is possible in off-line application scenarios). We show results on multiple publicly available datasets and demonstrate that without having access to any offline training data or the entire test video a priori (conditions that are restrictive for many application domains), our proposed method obtains results similar to or better than those that do impose the above conditions.

In building our SBM model, we exploit both the spatial and temporal information between neighboring targets. As an example, consider Fig. 1. The upper row shows the results of the tracking with the SBM, while the lower ones shows the result of the same tracker without the SBM. In the third frame of the lower row, it is seen that an ID switch happens because two people with very similar appearances are seen close together. However, in the upper row, the SBM captures the information that there was a group of three people (shown with the circle) and one separate individual. When the appearance information is combined with this group information, the overall accuracy of the tracking improves. Therefore, it can be said that, in our framework all the targets act as a "spatial-temporal" context to each other. This leads to the *joint estimation* of the associations between all the targets.

An overview of the framework is given in Fig. 2. We first use a state-of-art detector [8] to detect persons in each frame. The input to our algorithm are the tracklets generated by a basic tracker using a particle filtering method.

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.113

IEEE computer society

Figure 1. Illustration of how SBM works. The pink circle represents a group of people. The dotted lines represent their walking paths. The first row shows that considering the walking behavior in the group (inside the pink circle), the tracking results are correct. The second row shows the independent estimation of individual target tracks yields wrong association results.

Our algorithm associates the short term tracklets to form long term tracks by finding the best similarity between the tracklets. The similarity computation has two parts: an individual tracklet affinity model, and the learned SBM that considers the spatio-temporal motion relationships between multiple tracklets (as described in detail below). The model parameters are learned online using a belief propagation framework. We show experimental results on multiple publicly available datasets and clearly describe the improvements generated by using the SBM.

2. Related Work

We briefly review the most relevant papers so as to better explain the contribution of the proposed approach. One of the most popular recent class of methods has been Data Association based Tracking (DAT), in which there are two main components: a tracklet affinity model and the association optimization framework. The tracklet affinity models can be divided into two classes: those using only past observations to estimate the current state (online method) [4, 6] and those using both past and future information to estimate the current state (batch method) [25, 30]. The methodology for computing the associations relies on minimizing a defined cost function using existing approaches, e.g. Linear Programming [12] and Markov Chain Monte Carlo [22].

The accuracy of the above methods is dependent on how reliably they can compute the affinity scores between the tracklets, and many researchers have looked at this problem. The authors in [10] have described a hierarchical association framework to link tracklets into longer ones, while [4] uses dynamic feature selection. The method in [24] is designed to exploit image appearance cues to prevent identity switches.

The use of context in tracking is being actively explored currently. The paper [9] makes use of local image features to vote for the object positions. These features are learned online to find "supporters" of the target. The authors in [29] define "auxiliary objects" based on data mining techniques. Our proposed method uses the other tracked targets in the scene as context information rather than trying to extract new information from the images. This notion of context is based on the dynamics of the targets and is the reason why we use the term "social behavior" to describe it. The use of tracklets acting as each others context information has the advantage that it can be used even when it is difficult to find other "supporting" objects in the scene. A typical example is a multi-camera setup where the background objects can change significantly between the views, but the social relationships can remain intact. It is possible to consider a future scheme that combines both these notions of context into a single framework, but that is beyond the scope of this paper.

Multiple researchers have proposed various behavior models in other applications [19], and some of these have been applied to tracking. The parameter learning of SBMs can be divided into two categories - offline learning and online learning. In the offline learning category, [21] was possibly the first to consider incorporating social behavior models into tracking. Their model was built on avoiding collisions when people are walking and was incorporated into the search mechanism for tracking. Our proposed method considers a different scenario by modeling the probability that people will stay together. Thus, their models are complimentary to ours. The work in [26] can be seen as an extension of [21], which combines several factors that impact how a person walks, including the presence of other people. The goal of [26] was on target position prediction while ours is on data association between multiple targets. The recent work [23] describes a method for forming groups for better tracking given that the entire video is available, while [20] requires a certain amount of latency but not the entire time window. The use of the SBMs in our method goes beyond simple grouping since it combines the groups with inter- and intra-person characteristics for more accu-

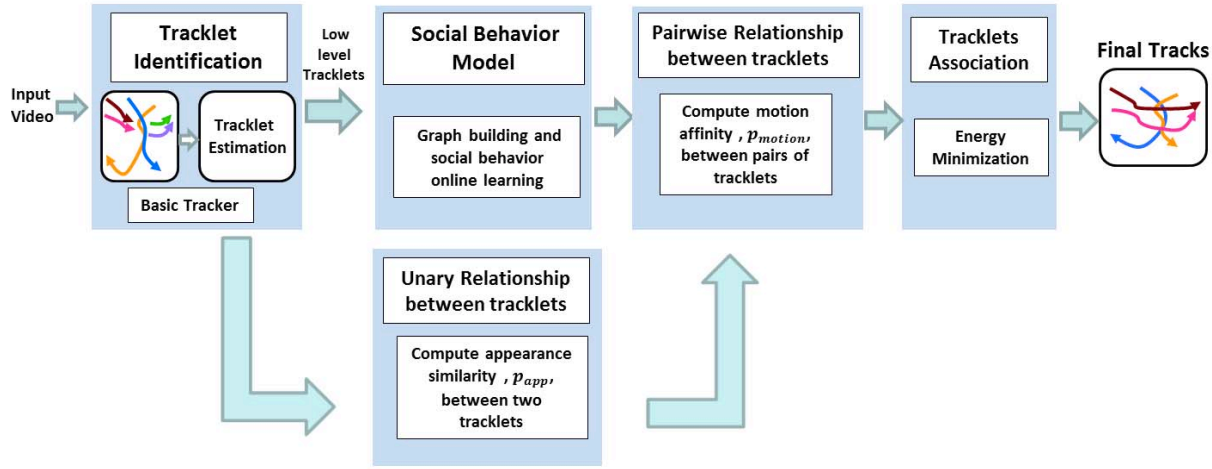


Figure 2. Overview of proposed approach

rate association. The work [15] also use the idea of grouping targets where the parameters are learned offline. A key difference of all these approaches with the proposed work is that we present a method where the relationship model is learned online.

In the set of papers that learn a relationship model online, [17] considers a problem similar to [21] with a complex person interaction model. As mentioned above, pairwise context information is ignored in this paper and it is complimentary to the proposed method. The recent paper [5] mentioned exploiting the pairwise relationships between people and proposed a simple model for this purpose. A pairwise model for inter-person activities is considered in [28]; the scheme for computing the affinities between detected targets is a two-step process. At first, a preliminary tracklet association is computed based on the appearance and motion similarities between individual targets and then switches are made to the association based on the pairwise relations. The two-step optimization strategy is similar to [25] in the sense that they also, preliminarily, associate tracklets based on appearance information and then exploit the smoothness of features along a long track to correct wrong associations. For both cases, the results after the first step are the global optimums that can be obtained with the set of features used in this step. This is due to use of the Hungarian algorithm in this step. However, the ultimate association results may not be the global optimum for the combined features as the association is being altered through the second set of features and the optimization is run on the first set of features. We combine the appearance based and the motion based features via the SBM and run the Hungarian algorithm on the combined score which guarantees the global optimum for all the features. The recent work [7] provides a graphical framework for modeling the social relationships of actions that can be inferred from videos; however, this approach has not been combined with tracking.

3. Social Behavior Model (SBM)

We begin by describing the construction of the SBM based on the motion characteristics of the tracklets. Let us consider two consecutive time windows, (T) and $(T+1)$, over which we need to find the associations between the tracklets (Fig. 3(a)). Assuming there are p tracklets in (T) and q in $(T+1)$, the goal of our model is to find the best match between these two sets of tracklets. This is achieved by considering not only the characteristics of the individual tracklets, but also the *motion relationship* between pairs of tracklets in (T) and $(T+1)$. The underlying notion is to compute the probability that two pairs of tracklets in (T) and $(T+1)$ have similar motion characteristics (direction, position, speed). The method does not require the tracklets to cover the entire time window, *i.e.*, they can be of varying lengths, extending over multiple windows.

The SBM is represented as a graph $G = (N, E)$, where N is the set of the nodes of the graph and E is the set of edges connecting the nodes (Fig. 3(b)). Each node, in this graph, represents a pair of tracklets at different time steps, *i.e.*, $N_{ij} = (\mathcal{X}_i^{(T)}, \mathcal{X}_j^{(T+1)})$. The weight of the node represents the possibility that the two tracklets $\mathcal{X}_i^{(T)}$ and $\mathcal{X}_j^{(T+1)}$ belong to the same target and is represented by $w_n(\mathcal{X}_i^{(T)}, \mathcal{X}_j^{(T+1)})$. The weight of the edge between two nodes depicts the joint occurrence probability of the two tracklets represented by the nodes. That is, if two nodes N_{ij} and N_{mn} have an edge of weight $w_e(N_{ij}, N_{mn})$ between them, then the probability that tracklet $\mathcal{X}_i^{(T)}$ is connected to $\mathcal{X}_j^{(T+1)}$ and tracklet $\mathcal{X}_m^{(T)}$ is connected to tracklet $\mathcal{X}_n^{(T+1)}$ is $w_e(N_{ij}, N_{mn})$. In time window (T), for any tracklet $\mathcal{X}_i^{(T)}$, we consider other nearby tracklets in the scene as context information for this tracklet. Let us call these context tracklets "supporting targets", denoted by $(\mathcal{X}_i^{(T)})^-$. For a node N_{ij}, the set of nodes N_{ij}^- are the supporting nodes of N_{ij}.

Table 1. Notation Table

Name	Definition
(T)	time interval
$(T+1)$	another time interval after (T)
$\underline{\mathbf{X}}$	a set of tracklets $\{\mathcal{X}_1, \mathcal{X}_2, \cdots\}$
$\mathcal{X}_i^{(T)}$	tracklet $\mathcal{X}_i$ in (T)
N_{ij}	a node consists of a pair of tracklets $(\mathcal{X}_i^{(T)}, \mathcal{X}_j^{(T+1)})$
$p_{app}(N_{ij})$	the probability of association between tracklets $\mathcal{X}_i^{(T)}$ and $\mathcal{X}_j^{(T+1)}$ based on appearance similarity
$p_{motion}(N_{ij})$	the probability of association between tracklets $\mathcal{X}_i^{(T)}$ and $\mathcal{X}_j^{(T+1)}$ based on motion similarity
N_{ij}^-	supporting nodes of N_{ij}
$(\mathcal{X}_i^{(T)})^-$	supporting targets of $\mathcal{X}_i^{(T)}$
$w_e(N_{ij}, Nmn)$	edge weight between nodes N_{ij} and Nmn
$w_n(\mathcal{X}_i^{(T)}, \mathcal{X}_j^{(T+1)})$	node weight N_{ij}

Fig. 3(b) shows a pictorial representation of the SBM for a pair of tracklets $\mathcal{X}_i^{(T)}$ and $\mathcal{X}_j^{(T+1)}$. The top layer of the graph represents the set of supporting nodes for N_{ij}, which can be represented precisely as

$$N_{ij}^- = \{(\mathcal{X}_l^{(T)}, \mathcal{X}_k^{(T+1)})\} \quad \text{where } \mathcal{X}_l^{(T)} \in \{(\underline{\mathbf{X}}^{(T)})\} \backslash \mathcal{X}_i^{(T)}\} \text{ and } \mathcal{X}_k^{(T+1)} \in \{(\underline{\mathbf{X}}^{(T+1)})\} \backslash \mathcal{X}_j^{(T+1)}\}. \tag{1}$$

Estimating the node and edge weights requires us to model the walking behavior of groups of people. In the next two sections, we explain how we mathematically model this behavior as the edge weights. Fig. 4 shows a pictorial representation of the process.

3.1. Computing Node Weights

To consider the motion characteristics of individual tracklets $\mathcal{X}_i^{(T)}$ and $\mathcal{X}_j^{(T+1)}$, we assume that each target follows a mean straight line path over a short time period. In our paper, we assume a constant velocity motion model. Let us define the individual time instants corresponding to the time windows as $[t_0, t_1, \cdots] \in (T)$ and $[t_0', t_1', \cdots] \in (T+1)$. Let $P_i^{t_u}$ be a point in the tracklet $\mathcal{X}_i^{(T)}$ and $P_j^{t_u'}$ be a point in $\mathcal{X}_j^{(T+1)}$. Defining the overhead bar as the expectation operator, the relationship between the two tracklets based on the velocity difference can be written as:

$$E_v = \begin{cases} 1, & \text{if } \overline{\dot{P}_i^{(T)}} - \overline{\dot{P}_j^{(T+1)}} < \delta_v, \\ \varepsilon_v, & \text{otherwise} \end{cases} \tag{2}$$

where the first expectation is taken over $\forall t_u \in (T)$ and the second is taken over $\forall t_u' \in (T+1)$. δ_v is a predefined threshold of the velocity difference between two people and ε_v is a small value. If E_v is small, then these two people have a small probability of walking together.

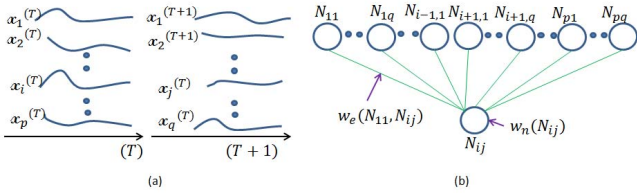


Figure 3. SBM formulation. (a) shows several tracklets in (T) and $(T+1)$, (b) illustrates the graphical model where the weight of each node is the independent estimation of probability of association of two tracklets in two time windows. The weight on the edge is based on the motion relationship between two pairs of tracklets.

We can also project ahead $P_i^{t_u}$ according to the current motion direction with time difference $t_u' - t_u$ and find the position difference between this projected point, $\hat{P}_i^{t_u'}$, and $P_j^{t_u'}$. The position of the projection point can be computed as:

$$\hat{P}_i^{t_u'} = P_i^{t_u} + \dot{P}_i^{t_u} \cdot (t_u' - t_u). \tag{3}$$

We can now compute the affinity between two tracklets based on the position information as:

$$E_P \propto e^{-\overline{(\hat{P}_i^{t_u'} - P_j^{t_u'})}} \tag{4}$$

Combining (2) and (4), the node weight of N_{ij}:

$$w_n(N_{ij}) \propto E_v \cdot E_P \tag{5}$$

Note that the idea in [21] can be incorporated into the calculation of node weights when the two targets representing the two tracklets move towards each other. If two people are not walking together but towards each other, our motion model can switch to their model in those time windows for those tracklets.

3.2. Computing Edge Weights

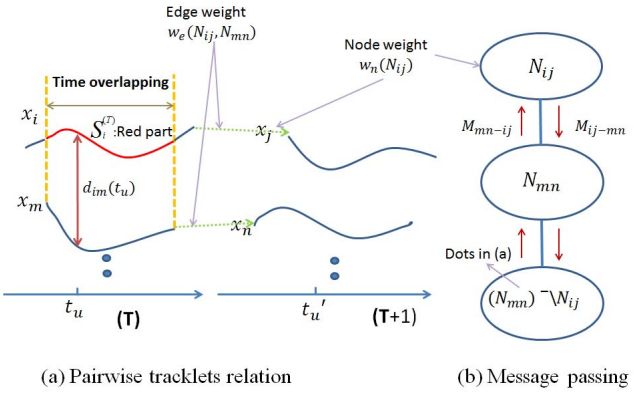
We consider two pairs of tracklets, $(\mathcal{X}_i^{(T)}, \mathcal{X}_j^{(T+1)})$ and $(\mathcal{X}_m^{(T)}, \mathcal{X}_n^{(T+1)})$. In this subsection, the superscript (T) and $(T+1)$ only consider the portion of tracklets overlapping in time (See Fig. 4 (a)). We define $S_i^{(T)}$ as a vector of all time overlapping positions of $\mathcal{X}_i^{(T)}$, (see the red line of Fig. 4 (a)). Similarly, $S_m^{(T)}$ is the corresponding part of $\mathcal{X}_m^{(T)}$. We assume a linear motion model with parameters $A^{(T)}$ and $B^{(T)}$ between them as:

$$S_i^{(T)} = S_m^{(T)} \cdot A^{(T)} + B^{(T)}, \tag{6}$$

where $S_i^{(T)} \in \mathbb{R}^{s \times 2}$, $S_m^{(T)} \in \mathbb{R}^{s \times 2}$, $A \in \mathbb{R}^{2 \times 2}$ and $B \in \mathbb{R}^{s \times 2}$.

We can obtain the least square estimate of $A^{(T)}$ and $B^{(T)}$ as:

$$\hat{A}^{(T)} = \left[(S_m^{(T)} - \bar{S_m}^{(T)})^T (S_m^{(T)} - \bar{S_m}^{(T)}) \right]^{-1} \cdot (S_m^{(T)} - \bar{S_m}^{(T)})^T (S_i^{(T)} - \bar{S_i}^{(T)}) \tag{7}$$



(a) Pairwise tracklets relation (b) Message passing

Figure 4. Pictorial representation of the notation used in (a) Sec. 2 and (b) Sec. 3. (a) corresponds to the Sec. 2 and (b) corresponds to Sec. 3.

and

$$\hat{B}^{(T)} = S_i^{(T)} - \hat{A}^{(T)} \cdot S_m^{(T)}. \quad (8)$$

Using the above estimated parameters between two tracklets in the time window (T), we want to compare if this linear model fits a pair of tracklets in the next time window $(T+1)$. The edge weight between N_{ij} and N_{mn} is given by the affinity between pairs of tracklets based on their motion information:

$$w_e(N_{ij}, N_{mn}) = c \cdot exp\{-\frac{1}{2\sigma^2} \cdot (S_j^{(T+1)} - S_n^{(T+1)} \cdot \hat{A}^{(T)} - \hat{B}^{(T)})^T \cdot (S_j^{(T+1)} - S_n^{(T+1)} \cdot \hat{A^{(T)}} - \hat{B}^{(T)})\} \quad (9)$$

where c is a normalization factor and σ is the variance of motion between tracklets.

4. Tracklet Association Framework

The overall tracklet association methodology combines the appearance and motion similarity between individual tracklets, as well as the similarity in the motion behavior between all pairs of tracklets. Recall that the SBM captures the motion similarity between individual tracklets as well as pairs of them. Therefore, we first show how to compute similarity based on the SBM described above using the max-product algorithm [3]. We then show how to obtain the final set of tracks by combining the SBM-based similarity with appearance information in a global optimization framework.

4.1. SBM-based Tracklet Similarity Computation

Given that the node and the edge weights on the SBM are established, we show how the max-product algorithm is used for calculating the similarity between the tracklets.

The max-product algorithm is an efficient algorithm for finding a set of values that jointly have the largest probability [3]. Using this knowledge, our goal is to find the maximum probability of association of all pairs of tracklets. In other words, to compute the probability of association between $\mathcal{X}_i^{(T)}$ and $\mathcal{X}_j^{(T+1)}$, we not only rely on the characteristics of the tracklets (independent estimation of association), but also on those of their supporting nodes N_{ij}^-. The message is sent from one node to another, which is denoted by M. The way a message between two nodes can be passed is given in the following steps.

Step 1: Message initialization. Without any prior knowledge of the messages, we assume the messages sent from one node are uniformly distributed.

Step 2: Message passing. After initialization, we need to find the message sent from every element of the set $\{(\mathcal{X}_i^{(T)})^-, (\mathcal{X}_j^{(T+1)})^-\}$ to $(\mathcal{X}_i^{(T)}, \mathcal{X}_j^{(T+1)})$. Similar to the previous section, we assume that $(\mathcal{X}_m^{(T)}, \mathcal{X}_n^{(T+1)}) \in \{(\mathcal{X}_i^{(T)})^-, (\mathcal{X}_j^{(T+1)})^-\}, \forall m, n$. We call this message passing process a "conversation". The max-product algorithm makes sure that this conversation stops after either convergence (the message changing will be less than a small value) or if the number of iterations of the "conversation" exceeds a predefined maximum iteration number. Given that the neighbor of supporting nodes is independent of tracking nodes, the message sent from N_{mn} to N_{ij} denoted by M_{mn-ij}, is defined as:

$$M_{mn-ij}(N_{mn}) = \max_{N_{mn}} \{w_n(N_{mn}) \cdot w_e(N_{ij}, N_{mn}) \cdot \prod_{l=(N_{mn})^-\backslash N_{ij}} M_{lr-ij}\}, \quad (10)$$

where r is the index of supporting tracklet of $\mathcal{X}_n^{(T+1)}$ except j. The messages pass from the supporting nodes N_{mn} to N_{ij} and this message is sent back to N_{mn}^-.

The message passing scheme is shown in Fig. 4 (b). As one target may serve as a supporting target for another target, there may be cycles in the graph. The loopy algorithm (*e.g.* asynchronous message passing schedule) is shown in [13] to achieve good results in practice. Even though there is a way to deal with loopy belief propagation, we limit the number of supporting targets to simplify the graph and reduce the computation. One example of implementation is given in Sec. 5.

Step 3: Belief readout. We read out the belief of associating $\mathcal{X}_i^{(T)}$ and $\mathcal{X}_j^{(T+1)}$ by using the product of all possible messages sent from N_{mn} to N_{ij} to help modifying the independent association results. This is given by:

$$p_{motion}(N_{ij}) \propto w_n(N_{ij}) \cdot \prod_{N_{mn}} M_{mn-ij}(N_{mn}). \quad (11)$$

Thus the algorithm gives a weight between a pair of tracklets by considering the relationships of this pair with its neighbors through the passing of the messages.

4.2. Appearance-based Tracklet Similarity Computation

The appearance information includes the color histogram (HSV space) and HOG feature. We use Bhattacharyya distance to find the appearance correlation between two tracklets, which can then yield the appearance-based similarity measure, p_{app}.

4.3. Tracklet Association

By combining the above SBM-based probability with the appearance similarity between tracklets, we can define the overall optimization function for computing the similarity between two tracklets using a weighted linear combination as:

$$p_{final}(N_{ij}) = \alpha \cdot p_{app}(N_{ij}) + \beta \cdot p_{motion}(N_{ij}). \quad (12)$$

α and β can be chosen as design parameters to decide on whether to weight appearance or motion information. This separation is intuitive and allows an user to easily set the weights on each part based on the characteristics of the video. Given the combined similarity scores, the final tracklet associations are computed using a bipartite matching scheme [11]. We generate a new graph where a node represents a tracklet and the weights between nodes are the affinity scores, p_{final} between two tracklets. We split the beginning and end of each tracklet into two subsets and build a weighted bipartite graph with k nodes, where k is the number of tracklets. We generate a $k \times k$ matrix, where each column belongs to the tracklet end set and each row belongs to the tracklet beginning set. The Hungarian Algorithm [11] is used to find only one best match for each row and column by minimizing the total cost.

The overall SBM-based tracking algorithm is given in Algorithm 1.

Algorithm 1: Overview of the SBM-based Tracking Algorithm

input : Tracklets
output: Associated tracklets

begin
- build a graph G with nodes as pairs of tracklets (Sec. 3)
- calculate the weights of nodes $w_n(N_{ij})$ (Sec. 3.1)
- **while** *message passing between nodes does not converge or stop at a predefined iteration number* **do**
  - consider one node N_{ij}
  - **for** *all neighbor nodes of* N_{ij} **do**
    - find out two pairs of tracklets in (T) and $(T+1)$, *i.e.* N_{ij}, N_{mn}
    - $M_{mn-ij} \leftarrow w_e(N_{ij}, N_{mn})$ (Equ. 10)
  - **end**
- **end**
- $p_{motion} \leftarrow M_{mn-ij}$ (Equ. 11)
- Update p_{final}
- Using Hungarian algorithm to find overall tracklet association results.

end

5. Experimental Results

To evaluate the performance of our algorithm, we test it on three different challenging datasets. The CAVIAR [1] dataset is captured in a shopping mall corridor with high occlusion, intersection and scale changes. The second dataset is the TUD Crossing dataset [2], which is a street view of pedestrians walking. The third one is the ETHZ central [16] dataset, where pedestrians are far away from the camera and people look very small.

There are different evaluation methods for multi-target tracking. We adopt the evaluation metrics used in [14, 18, 25, 27] for comparison. The definition of each metric is the same as these papers, where MT represents mostly tracked trajectories, ML represents mostly lost trajectories, Frag means fragments and IDS means ID switches.

In the implementation, we define a group model first. That is, we use the motion information (*i.e.* position, velocity, distance between targets, direction,time overlap) to cluster the tracklets in each time window. Because the number of tracklets in each time window is known (generated by a basic tracker), the clustering method is similar to [23], where the K-means clustering algorithm is applied here to get the clustering results. A cluster should have several targets with similar motion information; or in other words, these targets have high possibility to walk together. For example, assuming the number of tracklets is N_t, we can reduce the number of targets in a cluster $\lceil N_t/N_g \rceil$, where N_g is the number of clusters. The SBM considers relations between tracklets in the same cluster.

The CAVIAR dataset is a very popular public dataset. Especially in some video sequences, multiple people are walking in a group or interacting with other people which makes such sequences very challenging. Table 2 shows the comparison between some state-of-art works which contain HybridBoosted affinity modeling method [18], the min-cost flow approach [30], online learned discriminative appearance approach [14] and stochastic graph evolution framework [25]. Please note that [18, 25, 30]assume all the tracklets to be available and processes in batch mode to get the current association, while we only use previous and current observations in several time windows. Table 2 shows that our results are comparable to theirs even though we use less information. Our method does not need future observations and can be implemented as an on-line system. We also compare our results to the method without SBM,

and pictorial results are given in Fig. 5(a) and (b), where (a) shows the results with some errors and (b) uses SBM to correct them.

Table 2. Tracking results on the CAVIAR dataset. Results of [14, 18, 30] are reported on 20 sequences and [25] on 7 datasets which is the same as ours. (the most challenging parts of the CAVIAR dataset)

	MT	ML	Fg	IDS
Zhang *et al.* [30]	85.7%	3.6%	20	15
Li *et al.* [18]	84.6%	1.4%	17	11
Kuo *et al.* [14]	84.6%	0.7%	18	11
Song *et al.* [25]	84.0%	4.0%	6	8
Proposed method	85.3%	4.0%	7	7

We next show the results on the ETHZ central dataset where the camera is far from the tracked targets. Thus the people are very small in the scene. We demonstrate our results compared to the model without SBM in Table 3.

Table 3. Tracking results on the ETHZ central dataset. The number of fragments is reduced using the SBM.

	MT	ML	Fg	IDS
Without SBM	77.1%	14.3%	9	3
With SBM	82.9%	14.3%	6	3

The third dataset we use is the TUD crossing dataset which has lot of interacting people. The results show that we are able to improve the tracking results using SBM. We are able to track all the visible people in the scene. The pictorial results are shown in Fig. 5(c) and (d), where (c) is before using SBM and (d) uses SBM. Both fragments and ID switch can be reduced using SBM. [5] also shows the results from this dataset, but they use different evaluation metrics, so we are not able to compare our results with theirs.

Table 4. Tracking results on TUD crossing dataset. The number of fragments and ID switches is reduced using the SBM.

	MT	ML	Fg	IDS
Without SBM	85.7%	14.3%	4	4
With SBM	85.7%	14.3%	2	1

6. Conclusion

In this paper, we addressed the problem of tracking multiple targets by considering that people tend to walk together in most scenarios. Assuming that we are able to generate reasonable tracklets, we present a new tracklet association approach considering both spatial and temporal relationship between them. The spatio-temporal relationships are captured through a social behavior model. The method can be extended to track objects other than people, *e.g.*, cars on a highway also have an inter-relationships which can be similarly exploited. We compared our results to some state-of-art works and demonstrated promising results on several challenging datasets.

References

[1] CAVIAR dataset. `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1`.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[3] C. M. Bishop. Pattern recognition and machine learning. Springer New York, Secaucus, NJ, 2006.

[4] M. D. Breitenstein, F. Reichlin, B. Leibe, and E. Koller-Meier. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009.

[5] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011.

[6] Y. Cai, N. d. Freitas, and J. J. Little. Robust visual tracking for multiple targets. In *ECCV*, 2006.

[7] L. Ding and A. Yilmaz. Inferring social relations from visual concepts. In *ICCV*, 2011.

[8] P. Felzenszwalb, R. Girshick, D. Mcallester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010.

[9] H. Grabner, J. Matas, L. V. Gool, and P. Cattin. Tracking the invisible: learning where the object might be. In *CVPR*, 2010.

[10] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008.

[11] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *ICCV*, 2003.

[12] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007.

[13] D. Koller and N. Friedman. Probabilistic graphic models: principles and techniques. The MIT Press, Cambridge, MA, 2009.

[14] C. H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010.

[15] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: modeling social and grouping behavior on a linear programming multiple people tracker. In *CVPR Workshops*, 2011.

[16] B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

[17] G. Li, W. Qu, and Q. Huang. A multiple targets appearance tracker based on object interaction models. *IEEE Trans. on Circuits and Systems for Video technology*, (99), 2011.

[18] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009.

[19] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behavior of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5(4), 2010.

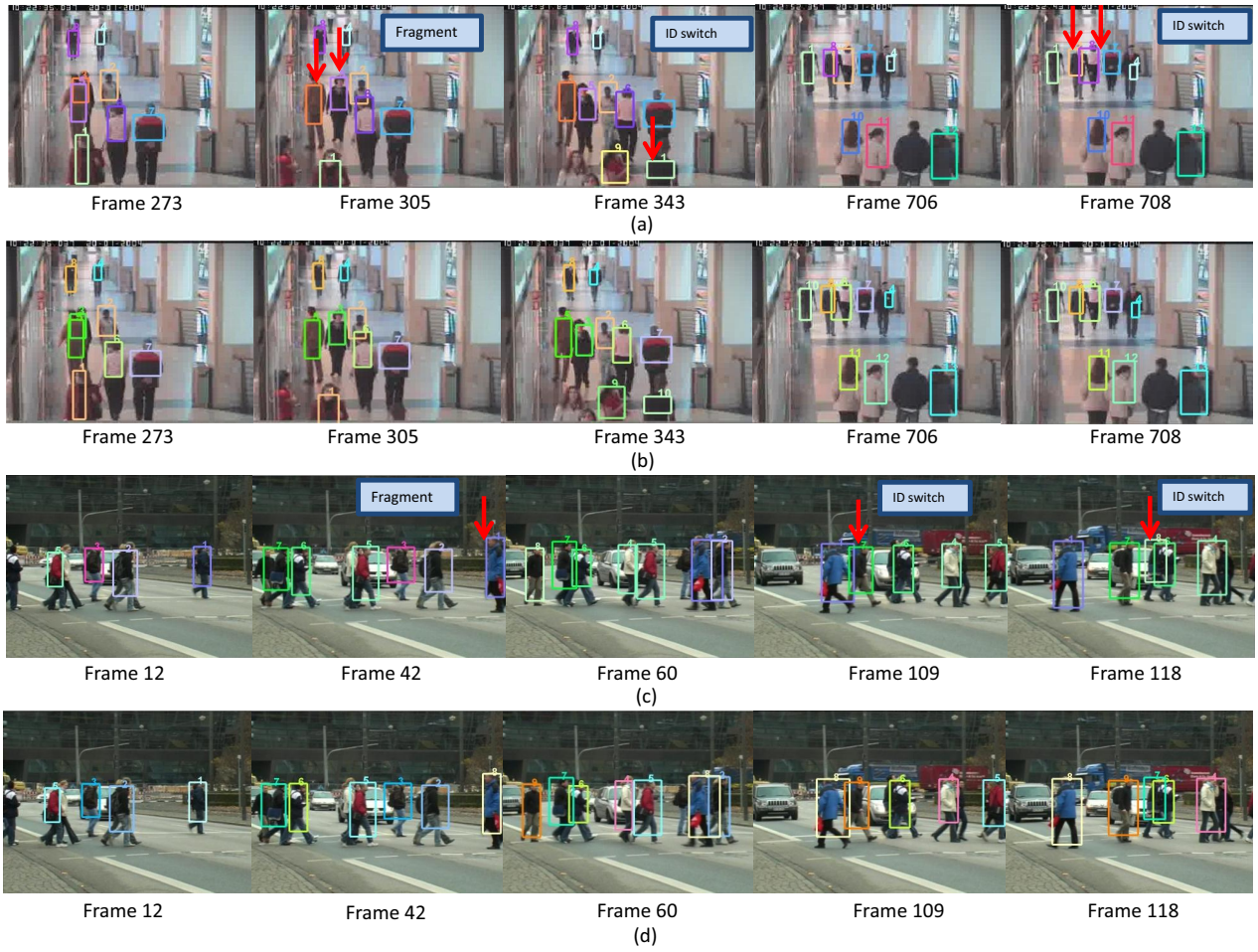


Figure 5. Advantage of using SBM. (a) and (b) are results from one sequence of CAVIAR dataset. (a) shows the result using prediction based affinity model without SBM, which contains several ID switches or fragments. (b) shows the result of our proposed method, with the errors in (a) corrected. (c) and (d) are results from TUD crossing dataset.(d) uses the SBM correcting the wrong association results in (c).

[20] S. Pellegrini, A. Ess, and L. V. Gool. Improving data association by joint modeling of pedestrain trajectories and groupings. In *ECCV*, 2010.

[21] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.

[22] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006.

[23] Z. Qin and C. R. Shelton. Improving multi-target tracking via social grouping. In *CVPR*, 2012.

[24] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, 2011.

[25] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*, 2010.

[26] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and B. T. L. Who are you with and where are you going. In *CVPR*, 2011.

[27] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In *CVPR*, 2011.

[28] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR*, 2012.

[29] M. Yang, W. Ying, and G. Hua. Context-aware visual tracking. *PAMI*, 31(7):1195 – 1209, 2008.

[30] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.