Computer Vision and Image Understanding 134 (2015) 64–73

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Tracking multiple interacting targets in a camera network

Shu Zhang, Yingying Zhu, Amit Roy-Chowdhury *

Dept. of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA

ARTICLE INFO

Article history:
Received 23 January 2014
Accepted 16 January 2015

Keywords:
Multi-camera tracking
Multi-target tracking
Interacting targets
Wide-area camera network
Network flow

ABSTRACT

In this paper we propose a framework for tracking multiple interacting targets in a wide-area camera network consisting of both overlapping and non-overlapping cameras. Our method is motivated from observations that both individuals and groups of targets interact with each other in natural scenes. We associate each raw target trajectory (*i.e.*, a tracklet) with a group state, which indicates if the trajectory belongs to an individual or a group. Structural Support Vector Machine (SSVM) is applied to the group states to decide if merge or split events occur in the scene. Information fusion between multiple overlapping cameras is handled using a homography-based voting scheme. The problem of tracking multiple interacting targets is then converted to a network flow problem, for which the solution can be obtained by the K-shortest paths algorithm. We demonstrate the effectiveness of the proposed algorithm on the challenging VideoWeb dataset in which a large amount of multi-person interaction activities are present. Comparative analysis with state-of-the-art methods is also shown.

1. Introduction

Multi-target tracking in a camera network, although attractive to researchers for a long time [1,2], still remains challenging. In particular, large illumination variations across cameras, cluttered scenarios and a camera network with both overlapping and non-overlapping views pose impediments to traditional tracking algorithms. Moreover, multiple cameras require computing associations between detected targets in different views, which can also be a challenging task. In this paper, we propose a novel camera network tracking scheme, called Switching Network Tracker (SNT), for tracking of multiple interacting targets in a cluttered camera network scene with both overlapping and non-overlapping views.

Our scheme is motivated by natural scenes as shown in Fig. 1, where people interact with each other in a camera network. We see that people often congregate together in a way where it may be difficult to individually detect them, and these congregations might split either within the view of an individual camera, or in the blind areas between non-overlapping cameras. When an individual target cannot be detected, we can still get an estimate of the target's state by tracking the group into which it merged. Therefore, it is necessary to design a tracking scheme for camera networks, which integrates group tracking and individual tracking, and *switches* smoothly between the two to generate robust individual tracks.

Besides, a wide-area camera network usually consists of both overlapping and non-overlapping views. In order for a tracking scheme to obtain long and stable tracks for every target within and across cameras, both overlapping and non-overlapping views should be addressed. For overlapping views, cooperation among cameras is necessary to improve the system performance. For non-overlapping views, different illumination conditions and the unknown behaviors in blind area between cameras should be taken into consideration in order to obtain a stable tracking system.

There are a few works which systematically address the problem of multi-target tracking in camera networks with both overlapping and non-overlapping views. The papers [3,4] extended the recent multi-target tracking framework in [5,6] to a non-overlapping camera network application. Detections of a person are associated to form a short but robust track, the so called tracklet, for this person. However, the problem of tracking targets in a cluttered scene with a large number of interacting activities has not been addressed in these works. The works in [7,8] addressed the tracking problem in overlapping views with a similar tracklet association scheme, but clutter due to people grouping together was not addressed.

The proposed SNT is designed to handle a cluttered scene with multiple interacting targets. Specifically, the SNT can track individuals and groups simultaneously in a camera network. Individuals are tracked in uncluttered scenes where single targets can be clearly detected, whereas the groups are tracked for cluttered environment where individual target detection is inaccurate or

[*] Corresponding author.

http://dx.doi.org/10.1016/j.cviu.2015.01.002
1077-3142/© 2015 Elsevier Inc. All rights reserved.

infeasible with state-of-the-art detectors. We assign each detected region with a group state to define if the region is associated with a group or an individual. In order to identify a group state of a detected region, a Structural Support Vector Machine (SSVM) model is constructed, upon local features of each target and the relationships between targets, to determine the merging and splitting events occurring in the scene. In the operational phase of the system, detections that can be associated with high confidence lead to the formation of tracklets. Each tracklet is assigned with a group state by the learned SSVM based on all associated observations from cameras with overlapping views. Then, a homography based target state estimation methodology is applied to fuse tracklets in overlapping views. With the consistent tracking results on overlapping views, the problem of tracking multiple interacting targets in a camera network is formulated as a network flow problem. We show that such a problem can be converted to a mixed integer programming problem and the K-shortest paths algorithm [9,10] is used to obtain the optimal solution.

In the experiments, we work on the VideoWeb dataset [11], which is a publicly available camera network dataset with both overlapping and non-overlapping camera views. To the best of our knowledge, we deal with a far more complex multi-camera scenario than previous works that have looked into this problem [3,4,7,12], in terms of the number of cameras, targets, their actions, and camera fields of view. We demonstrate the effectiveness of our tracking algorithm with thorough test results.

1.1. Related works

We briefly review the most relevant papers on tracking in camera networks so as to better explain the contribution of the proposed approach. In general, tracking in a camera network can be divided into two parts: tracking in a non-overlapping camera network and tracking in an overlapping camera network.

In the first category, [13] is one of the early works on non-overlapping multi-camera tracking, in which appearance relationships between cameras were used to establish correspondence. The work in [14] learned a camera network topology and path probabilities of objects. Many works focused on spatio-temporal cues to solve the tracking problem. Ref. [15] investigated the unsupervised learning of a model of trajectories based on the activity information. Ref. [16] used a stochastic transition matrix to describe motions between cameras. Similarly, [17,18] proposed new transition distributions based on statistical dependence between observations in different cameras. The work of [1,2,12] learned the brightness transfer functions (BTFs) either online or offline between cameras. Ref. [4] learned an appearance affinity model between two non-overlapping cameras online. Some recent works on tracking in non-overlapping camera views combine both appearance information and spatio-temporal cues together to achieve better results. Refs. [3,19] proposed an optimization framework by combining short term feature correspondences across the cameras with the long-term feature dependency models. Ref. [20] did not use the spatio-temporal cues in multi camera scenarios, but instead investigated directional angles using the spatio-temporal continuity in a single camera field. However, most of these works failed to handle a high clutter scene.

In the second category (overlapping views), most works used systems with calibrated cameras. Ref. [7] projected all the blobs in different cameras onto the ground plane and then performed standard feature association algorithms. Ref. [8] used a similar method but developed a greedy matching algorithm which can achieve results similar to the Hungarian algorithm but with less computation. Ref. [21] determined spatial positions by transforming images based on a ground plane homography. Ref. [22] estimated a ground plane occupancy map to track people by their 2D segmentations in each camera. Ref. [23] exploited both dynamical and geometrical constraints to improve robustness to occlusion.

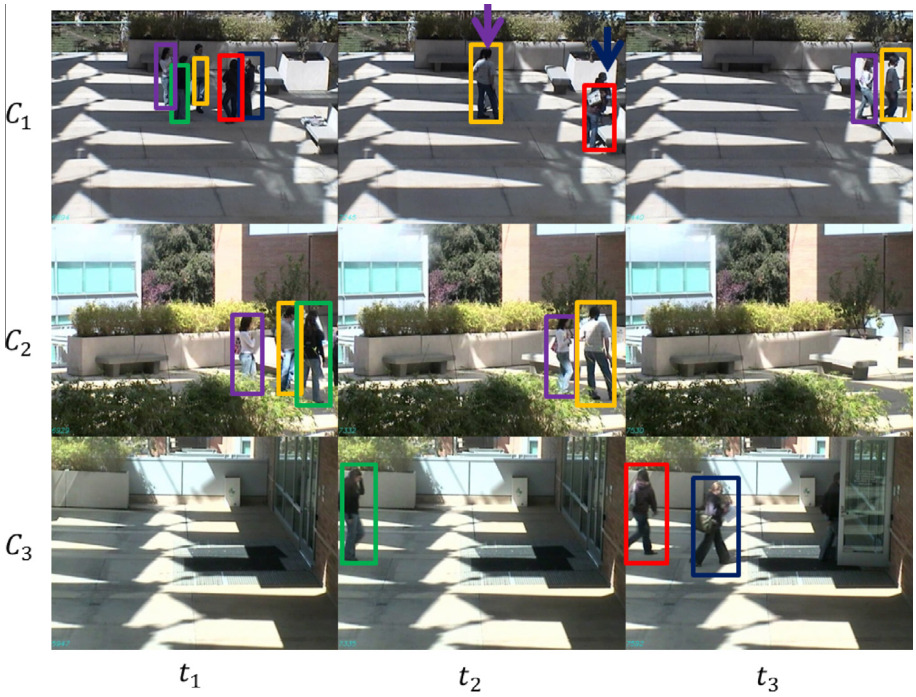

Fig. 1. Tracking challenges in a camera network, where C_1 and C_2 have overlapping views and C_3 is not overlapped with either C_1 or C_2. The horizontal axis represents time and the vertical axis shows three different cameras in each time step. At t_1, five persons stay in a group who are marked with five different colors. C_1 can fully observe these five persons, while C_2 can only observe three of them. At t_2, the five-persons group splits into three parts. C_1 observes the persons in yellow and red, while the persons in purple and blue (illustrated by two arrows) are occluded by them. However, the person in purple can be fully observed in C_2. At time t_3, the person in purple in C_1, who is occluded at t_2, can be fully observed. In C_3, the group where only the person in red is seen at t_2 is recognized as two individual persons, which means that these two persons split in the blind area between C_1 and C_3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Ref. [24] explored a distributed estimation strategy for tracking and data association.

Tracking methods utilizing group information have been studied in single camera tracking schemes. In the work of [25–27], group information worked as a constraint to improve individual tracking performance. The algorithm in [28] jointly modeled individual and group information. Ref. [29,30] proposed the problem of group tracking with a descriptor of appearance features. Ref. [31] exploited the social force between two pedestrians to associate groups of people. However, none of these works used two classes of trackers that could freely track groups and individuals simultaneously to obtain robust tracks for every target. Person re-identification [32] is another method which finds the one to one correspondences between targets in different cameras. However, no group information is used in such an application, and person re-identification datasets are more constrained than what we deal with in this work.

1.2. Contributions

Our work has three **main contributions**:

1. We propose a novel tracking framework – SNT – for camera networks. The SNT is designed to generate robust and long individual tracks, even in cluttered scenes, by tracking both individuals and groups simultaneously depending on the degree of clutter in the scenes.
2. We design an SSVM that integrates spatial and temporal relationships between tracklets to detect group formation and splitting in a camera network. With the merging and splitting events, group states of tracklets can be better determined, which is significant for the smooth switching between individual and groups across cameras.
3. A general tracklet association algorithm is developed for both overlapping and non-overlapping scenes. We introduce group nodes to the standard min-cost network flow problem and modify the problem accordingly to handle multiple interacting targets in a camera network. An approach based on linear programming is proposed to solve the modified network flow problem.

2. Group model using a structural SVM (SSVM)

As interactions between targets may lead to a situation where individuals cannot be detected separately, a group model is needed. We are also interested in the merge and split activities of the group so as to obtain a long track for each target.

In the traditional definition of a group [5,25], a tracklet is seen in a group when there is at least one nearby tracklet within the same time window. Corner features were commonly used to cluster the trajectories into groups [33,34], while [35] performed group detection based on tracklets. We use the tracklets as the inputs of our group detector, while the corner features provided additional cues in case of missing detections. Following the method in [6,36], we use the particle filter to associate detections into tracklets. We train an SVM classifier [37] upon the features of the bounding boxes. Note that the group detector is not the focus of our work and can be replaced by any advanced group detector. Three classification scores (named group states g) are obtained: individual (0), group (1) and others (2). $g = 2$ is needed to deal with situations when it is not clear if an individual or group is detected, which can happen when groups merge or split.

There are three possible group events between two group states: merge, split, and stable. The input of the group model is a set of tracklets in camera C_m; $\mathcal{T}^{C_m} = \{\mathcal{T}_1^{C_m}, \mathcal{T}_2^{C_m}, \ldots, \mathcal{T}_i^{C_m}, \ldots, \mathcal{T}_N^{C_m}\}$, where N is the total number of tracklets in C_m. We consider tracklets $\mathcal{T}_i^{C_m}$ and $\mathcal{T}_j^{C_m}$ from different times, where $\mathcal{T}_j^{C_m}$ starts after $\mathcal{T}_i^{C_m}$ ends and the start time of $\mathcal{T}_j^{C_m}$ minus the end time of $\mathcal{T}_i^{C_m}$ is within a threshold. The group event label y, evaluated over the two time windows, is defined as

$$y\left(\mathcal{T}_i^{C_m}, \mathcal{T}_j^{C_m}\right) = \begin{cases} 0, & \text{if no event}, \\ 1, & \text{if merge events detected}, \\ 2, & \text{if split events detected}. \end{cases} \tag{1}$$

As discussed above, if $g = 2$ for a tracklet, we need to learn if the tracklet is in a merge/split event. We propose a novel group event learning method which can jointly estimate the group event labels of a set of tracklets. A structural SVM model that integrates motion features with various context features is developed for this purpose. In the most of this section, we drop C_m from the notation of a tracklet because the group event detection is performed in every single camera view.

2.1. Motion feature descriptors

To detect the group state of a tracklet, both spatial context features and temporal context features are needed. This is because a tracklet's group state change depends on the spatial relationship between this tracklet and other tracklets. A merge/split event also depends on the temporal relationship between two tracklets, *i.e.*, an individual merges to a group over time. A tracklet i's motion features include the average size, the position on the first and last frame, and the moving speed. A motion feature descriptor of tracklet i is represented as $[W_i, H_i, X_i, Y_i, s_{xi}, s_{y_i}]$. We use W and H to represent the average width and height of a tracklet between its first and last frame, and use X and Y to represent the average horizontal and vertical positions of a tracklet on the image plane between its first and last frame. s_x and s_y denote the average moving speed of the tracklet in horizontal and vertical directions.

Spatial context feature descriptor: Given a time window N_T, the spatial context feature is the spatial relationship between the interested tracklet and the nearby tracklets. $\mathcal{RS}_{ij}(n)$ and $\mathcal{RT}_{ij}(n)$ are the spatial and temporal relationships of $\mathcal{T}_i$ and $\mathcal{T}_j$ at frame n. The spatial relationship between these two tracklets is defined as the normalized histogram $\mathcal{RS}_{ij} = \frac{1}{N_T}\sum_{n=1}^{N_T}\mathcal{RS}_{ij}(n)$, where N_T is the number of frames in the time window. $\mathcal{RS}_{ij}(n)$ represents the distance between $\mathcal{T}_i$ and $\mathcal{T}_j$ at frame n. In practice, $\mathcal{RS}_{ij}(n)$ depends on the motion features of the tracklets. The distance between two tracklets $\mathcal{T}_i$ and $\mathcal{T}_j$ at frame n is $d^n(\mathcal{T}_i, \mathcal{T}_j) = \sqrt{(X_i(n) - X_j(n))^2 + (Y_i(n) - Y_j(n))^2}$. The spatial context feature descriptor between $\mathcal{T}_i$ and $\mathcal{T}_j$ at frame n is defined as

$$\mathcal{RS}_{ij}(n) = \begin{cases} [1\ 0\ 0]^T, & \text{if } d^n(\mathcal{T}_i, \mathcal{T}_j) < \min\{W_i(n), W_j(n), H_i(n), H_j(n)\}, \\ [0\ 0\ 1]^T, & \text{if } d^n(\mathcal{T}_i, \mathcal{T}_j) > 2 \times \min\{W_i(n), W_j(n), H_i(n), H_j(n)\}, \\ [0\ 1\ 0]^T, & \text{otherwise}. \end{cases} \tag{2}$$

Temporal context feature descriptor: The SNT will not switch the tracking mode until a merge or split event is detected. The temporal relationship between merge/split events and the group states g is considered in the tracking system. There are 5 attribute subsets between two tracklets at two consecutive time windows. The attribute subset definitions can be found in Table 1. If any attribute A_i is satisfied, the corresponding attribute is 1 while the others are 0. A_i is determined by the tracklets' group states g and the feature descriptors of the tracklets. Note that the potential merge/split events are determined based on the average size change of the tracklets and become cues for the final label of these group events. The average size change cues include the number of persons as well as the size of the group on the image plane. If the number

Table 1
Different relationships between individuals and groups with $t_1 < t_2$.

Attribute subset	Associated attributes
A_1	t_1: an individual; t_2: a group
A_2	t_1: a small group; t_2: a large group
A_3	t_1: a group; t_2: an individual
A_4	t_1: a large group; t_2: a small group
A_5	Otherwise

of persons does not change while the size of the group on the image plane has a significant change, there is a high possibility that the person detector missed some persons in the crowded scene. The normalized histogram $\mathcal{RT}_{ij}$ is the temporal context feature of tracklet $\mathcal{T}_i$ with respect to tracklet $\mathcal{T}_j$. We define $\mathcal{RT}_{ij}$ as a 5-bin histogram, an example of which is shown in Fig. 2.

2.2. Group merge and split model

Given all the features of tracklets, the goal is to detect the merge or split events. A set of tracklets $\mathcal{T}$ is associated with a label vector $y = \{y_i\},\ i = 1, 2, \ldots, N$, where $y_i \in \{0, 1, 2\}$ is the group event label vector of $\mathcal{T}_i$. We infer the group states of the tracklets set from the combination of various context features discussed above. Define $D_{\mathcal{RS}}$ and $D_{\mathcal{RT}}$ as the dimensions of $\mathcal{RS}_{ij}$ and $\mathcal{RT}_{ij}$. A potential function between features of $\mathcal{T}$ and label y is defined as $F(\mathcal{T}, y)$:

$$F(\mathcal{T},y) = \sum_{i=1,j=1,i\neq j}^{N} w^T_{\mathcal{RS},(y_i,y_j)}\mathcal{RS}_{ij} + \sum_{i=1,j=1,i\neq j}^{N} w^T_{\mathcal{RT},(y_i,y_j)}\mathcal{RT}_{ij} \qquad (3)$$

where $\mathcal{RS}_{ij} \in \mathbb{R}^{D_{\mathcal{RS}}}$ and $\mathcal{RT}_{ij} \in \mathbb{R}^{D_{\mathcal{RT}}}$ are the spatial and temporal context feature descriptors associated with tracklet $\mathcal{T}_i$ and $\mathcal{T}_j$ respectively. $w_{\mathcal{RS},(y_i,y_j)} \in \mathbb{R}^{D_{\mathcal{RS}}}$ and $w_{\mathcal{RT},(y_i,y_j)} \in \mathbb{R}^{D_{\mathcal{RT}}}$ are the weights that capture the spatial and temporal relationships of group event classes y_i and y_j. N is the number of tracklets.

2.3. Model learning and inference

The potential function $F(\mathcal{T}, y)$ can be converted to a linear function with a parameter vector w. Define y_i the label vector of the corresponding tracklet $\mathcal{T}_i$. We first rewrite Eq. (3) as:

$$F(\mathcal{T},y) = w^T_{\mathcal{RS}} \sum_{i=1,j=1,i\neq j}^{N} \psi(\mathcal{RS}_{ij}, y_i, y_j) + w^T_{\mathcal{RT}} \sum_{i=1,j=1,i\neq j}^{N} \phi(\mathcal{RT}_{ij}, y_i, y_j) \qquad (4)$$

where $w_{\mathcal{RS}}$ and $w_{\mathcal{RT}}$ are weight vectors and defined as

$$w_{\mathcal{RS}} = \left[w^T_{\mathcal{RS},(1,1)} \ \cdots w^T_{\mathcal{RS},(1,N)} \ \cdots \ w^T_{\mathcal{RS},(N,N)}\right]^T,$$

$$w_{\mathcal{RT}} = \left[w^T_{\mathcal{RT},(1,1)} \ \cdots w^T_{\mathcal{RT},(1,N)} \ \cdots \ w^T_{\mathcal{RT},(N,N)}\right]^T,$$

and $\psi(\mathcal{RS}_{ij}, y_i, y_j)$ and $\phi(\mathcal{RT}_{ij}, y_i, y_j)$ have non-zero entries at the position corresponding to class pair (y_i, y_j).

We define the joint weight vector w and the joint feature vector $E(\mathcal{T}, y)$ as $w = [w^T_{\mathcal{RS}}, w^T_{\mathcal{RT}}]^T$ and $E(\mathcal{T}, y) = \left[\sum_{i,j,i\neq j}\psi(\mathcal{RS}_{ij}, y_i, y_j),\right.$ $\left.\sum_{i,j,i\neq j}\phi(\mathcal{RT}_{ij}, y_i, y_j)\right]^T$, where $i, j = 1, \ldots, N$. Then Eq. (4) can be expressed as

$$F(\mathcal{T},y) = w^T E(\mathcal{T},y), \qquad (5)$$

The learning algorithm can be written as

$$w^* = \arg\min_{w}\left\{\frac{1}{2}w^T w - C\sum_{i=1}^{M} w^T E(\mathcal{T}_i, y_i) + C\sum_{i=1}^{M}\max_{y_i}\left[w^T E(\mathcal{T}_i, y_i) + \Delta(\mathcal{T}_i, y_i)\right],\right. \qquad (6)$$

where Δ is the number of tracklets that associate with incorrect labels and C controls the tradeoff between the errors in the training model and margin maximization. Eq. (6) can be converted to an unconstrained convex optimization problem and solved by the cutting plane methodology [38].

In the inference part, we adopt the greedy search methodology in [39–41] to find the optimal label vector y_{test}. We first initialize the assignment sets, the label set y, and the instanced tracklet set $\mathcal{T}$ as the null sets, indicating no tracklet state is recognized yet. We augment these sets by iteratively adding the labeled tracklet that increases the potential score the most.

3. Camera network tracklet association

Having the group state and the group event label of every tracklet in each camera, tracklet fusion in cameras with overlapping views is performed to obtain the consistent group states for the overlapping tracklets. Then, with single camera tracking being done, the tracklet association scheme is applied to the camera network. The tracklet association scheme aims to associate tracklets into long, stable tracks. If merge/split events are detected by the SSVM group model in Section 2, a group tracklet is associated with its corresponding individual tracks.

3.1. Group state estimation in overlapping views

The SNT fuses all the tracklets in the overlapping views by a homography transformation between overlapping views similar to [21]. A homography transformation is used to project the ground plane from one camera view to that in another camera view. The group state of one target in different camera views may not be the same because of the differences in the observations. A

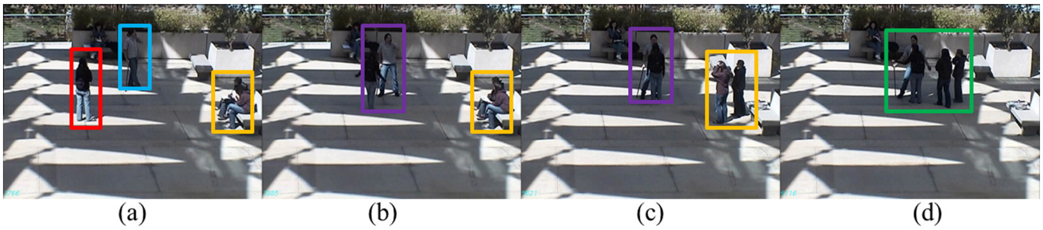

Fig. 2. An example of spatial and temporal context motion features in four consecutive time windows. (a) has two individual tracklets (red and blue rectangles) and one group tracklet (yellow rectangle). The spatial context descriptor between the blue and red individuals at this frame n is $\mathcal{RS}_{br}(n) = [1\ 0\ 0]$, where the subscripts denote the color blue and red. In (b), the purple group is merged by the two individuals in (a). The temporal context feature descriptor between the blue tracklet in (a) and the purple tracklet in (b) is $\mathcal{RT}_{bp} = [1\ 0\ 0\ 0\ 0]$ because it is the event that two individuals merge to a group. From (c) to (d), the purple group and the yellow group merge to a larger group. The temporal context feature descriptor between the purple tracklet and the green tracklet is thus $\mathcal{RT}_{pg} = [0\ 1\ 0\ 0\ 0]$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

weighted voting scheme is used to obtain the consistent group state of the same target across all cameras. Specifically, if there is an observation in the overlapping views between at least two cameras, the final group state is a linear combination of the group state from each camera. The group state from the camera with the widest view is assigned the highest weight while the ones from other cameras are assigned low weights. Thresholding is applied then to decide a consistent group state for the target.

3.1.1. Single camera tracking

We formulate the multi-target tracking task in a single camera as a network flow problem. We assume that there are l time intervals in a video sequence in every camera view. A graph $G = (V, E)$ is built as in Fig. 3. Every tracklet is seen as a node in G, where the group states of it in all the overlapping view cameras are consistent. We also add two virtual nodes: the source node v_{start} and the sink node v_{end} representing the starting and ending nodes. The edges correspond the admissible association between two nodes. The vertex set V consists of a start node v_{start}, a sink node v_{end}, and nodes from the time interval 1 to l. Each edge is assigned a cost $c_{i,j}$ which is based on the feature similarity between two nodes i and j. We define f_{ij} as a binary indicator variable that is 1 when there is an association between the nodes i and j and 0 otherwise. Thus,

$$f_{start,i} = \begin{cases} 1, & \text{if } i \text{ is the starting node of a track } \mathcal{X}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

$$f_{i,end} = \begin{cases} 1, & \text{if } i \text{ is the ending node of a track } \mathcal{X}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$$f_{i,j} = \begin{cases} 1, & \text{if there is an admissible associationbetween the nodes } i \text{ and } j \\ & \text{in two consecutive time intervals,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

where $\mathcal{X}_k$ represents a long, associated track.

For every node, the sum of flows arriving at a node j equals to the sum of outgoing flows from the node j. If there is an association between two tracklets, no other associations are allowed between either of these two tracklets. Thus the following constraint on the variable f must be satisfied,

$$f_{start,j} + \sum_i f_{i,j} = \sum_k f_{j,k} + f_{j,end}, \tag{10}$$

While a node can represent either a group or an individual, such a constraint in Eq. (10) cannot be always satisfied because a group can be merged or split into multiple individuals. This might cause a many-to-one matching problem. To avoid such a case, we use the group state of a node as well as the merge/split label obtained from Section 2 to temporarily change the number of nodes. Specifically, if there is a merge/split event before/after a group node, we add virtual group nodes to make the total number of group nodes including the virtual nodes equal to the number of individual nodes before/after the split/merge event. An example is shown in Fig. 3, in which the blue node with solid lines is a group node and the blue node with dotted lines is the virtual node. Thus,

$$N_t = \begin{cases} N_{t-1} + \widetilde{N}^G_{t-1}, & \text{if there is a merge event before } t, \\ N_{t+1} + \widetilde{N}^G_{t+1}, & \text{if there is a split event after } t, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

where N_t is the number of nodes in time interval t, and $\widetilde{N}^G_{t-1}$ is the number of the added virtual nodes in time interval $t-1$.

We define the cost between two nodes i and j in a graph G as the negative logarithm of the feature similarity between two nodes, which is denoted by $c_{i,j}$, *i.e.*,

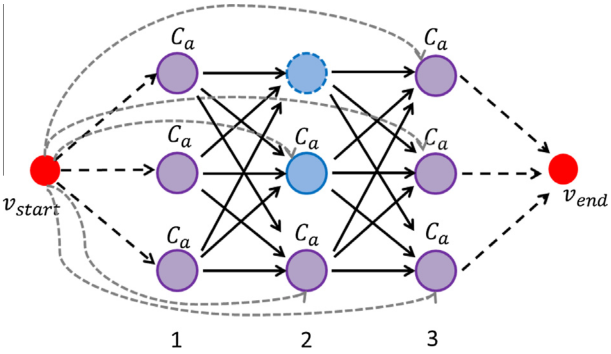

Fig. 3. A network flow framework for a simple graph in a single camera view. The purple nodes with solid lines represent the single target, the blue nodes with solid lines represent the group target, and the blue nodes with dotted lines are virtual nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$c_{i,j} = -\log \mathcal{S}(\mathcal{T}_i, \mathcal{T}_j), \tag{12}$$

where $\mathcal{S}$ is the similarity function.

The tracking problem in a single camera view is then formulated as

$$\text{Minimize} \quad \sum_j c_{start,j} f_{start,j} + \sum_{i,j} c_{i,j} f_{i,j} + \sum_j c_{j,end} f_{j,end}, \tag{13}$$

We also need to ensure that all flows from the source node v_{start} eventually end up in the sink node v_{end}, *i.e.*,

$$\sum_i f_{start,i} = \sum_k f_{k,end}. \tag{14}$$

3.1.2. Camera network tracking

In a camera network, the problem formulation is similar to the method above. However, some significant differences should be noticed. We assume that an edge only exists between two consecutive time steps excluding the source and sink nodes in a single camera. This is because a target's motion can be seen continuously. However, such an assumption is not valid for non-overlapping views because of the blind area between camera views. Thus the definition of f_{ij} in Eq. (9) is redefined as

$$f_{i,j} = \begin{cases} 1, & \text{if there is an admissible association between} \\ & \text{the nodes } i \text{ and } j \text{ within a time window;} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

In the network flow in a single camera, a target is seen as a node in every camera. Since the tracklet fusion can be done as stated above, all the observations of the same target are seen as one node in the new network flow framework. Such a simplification can guarantee that Eq. (10) is satisfied. An example of the network flow framework of a camera network can be seen in Fig. 4. The link from the second node at time 1 to the first node of time 3 is because of the blind area between C_a and C_b. Note that a node in camera a can have an edge with another node in the same camera since a target might leave the camera view and return back to the same camera.

The similarity between two observations is defined as $S\left(O_j^{C_b}, O_i^{C_a}\right)$, where $O_i^{C_a}$ denotes the observation of tracklet i in camera a. C_b can be any camera including the camera C_a.

The overall problem for tracking multiple interacting targets in a camera network can now be rewritten as a linear programming one.

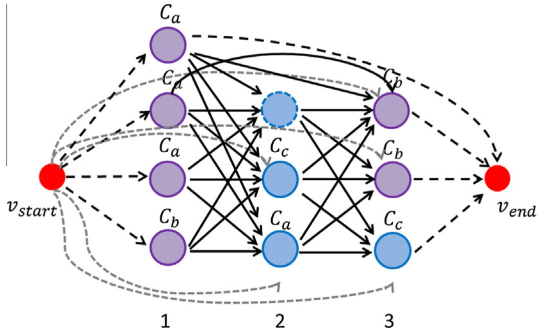

Fig. 4. A network flow framework for a simple graph in a camera network. The purple nodes with solid lines represent the single target, the blue nodes with solid lines represent the group target, and the blue nodes with dotted lines are virtual nodes. To keep the graph clean, we only show one example of two non-consecutive nodes from camera *a* in time interval 1 to camera *c* in time interval 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\begin{aligned}\text{Minimize}\ &\textstyle\sum_j -\log\left\{S_{start}(O_i^{C_a})\right\}f_{start,j} + \sum_{i,j} -\log\left\{S\left(O_j^{C_b},O_i^{C_a}\right)\right\}f_{i,j}\\ &+\sum_j -\log\left\{S_{end}(O_i^{C_c})\right\}f_{j,end}\\ \text{s.t.}\ & f_{i,j}\geqslant 0, f_{start,i}\geqslant 0, f_{i,end}\geqslant 0\quad \forall(i,j)\in E,\\ & f_{i,j}\leqslant 1, f_{start,i}\leqslant 1, f_{i,end}\leqslant 1\quad \forall(i,j)\in E,\\ & \textstyle\sum_k f_{j,k} + f_{j,end} - \left(f_{start,j}+\sum_i f_{i,j}\right)\leqslant 0,\\ & \textstyle\sum_k f_{k,end} - \sum_i f_{start,i}\leqslant 0\end{aligned}\tag{16}$$

where $S_{start}\left(O_i^{C_a}\right)$ represents the probability that the observation i in camera a is the starting point of a track and $S_{end}\left(O_i^{C_c}\right)$ representing this observation in camera a is the last observation of a track.

3.2. Solution of the linear programming problem

The problem in Eq. (16) is similar to a linear programming formulation with discrete variables. A similar problem has been studied in [42–45]. However, ours is the first work that addresses the problem of tracking multiple interacting targets, *i.e.*, both individuals and groups, into the network flow graph, and makes the problem different from existing works. Moreover, we work on a camera network with both overlapping and non-overlapping views while handling the relationships between individuals and groups. We realize that though the integer program (IP) can be solved by any generic IP solver, the size of the NP-complete problem makes the solution impractical. Such a problem is also known as a mixed integer programming problem [46]. Though the branch and bound algorithm has been proved to effectively solve such a problem, recent studies [43] have shown that the complexity of this problem can be reduced by reformulating the problem to a similar problem that approximates the original. The relaxed problem is called the k-shortest node-disjoint paths (KSP) problem on a directed acyclic graph (DAG).

Let $\mathcal{H}$ denote the feasible solutions of the problem in Eq. (16). Thus the optimal solution $\mathbf{f}^*$ is rewritten as

$$\operatorname{argmin}_{f\in\mathcal{H}} \sum_{a,b,i,j} c\left(e_{i,j}^{a,b}\right) f_{i,j} \tag{17}$$

where
$c\left(e_{i,j}^{a,b}\right) = -\log\left\{S\left(O_j^{C_b},O_i^{C_a}\right)\right\} - \log\left\{S_{start}\left(O_i^{C_a}\right)\right\} - \log\left\{S_{end}\left(O_i^{C_a}\right)\right\}.$

The optimal solution in Eq. (17) can be obtained as in [43]. We first run KSP on a single camera assuming the group state of each target is known. Then we add the shape and appearance features and the time gap constraints into the KSP in the camera network tracking scheme. The KSP is rerun and the optimal solution Eq. (17) is obtained. A detailed implementation will be provided in the experimental section.

4. Experiment

4.1. Dataset

We perform experiments on the public VideoWeb dataset [11] to assess the effectiveness of our tracking system. It contains both overlapping and non-overlapping views and complex target interactions. In VideoWeb dataset, each scenario has 8 cameras. There are totally $\binom{8}{2} = 28$ camera pairs per scenario. Among them, 12 pairs have overlapping views and the rest 16 pairs do not share overlapping views with each other. Each scenario has at least 8 persons walking into different camera views. Each video lasts for 4–6 min. 17 challenging scenarios are selected to test our algorithm. We use 7 scenarios for training and the remaining for testing. So totally 80 video sequences are used for testing our algorithm where the total video duration is around 350 min and the number of persons is 91. Each video sequence records real-world scenes with complex human activities that are present most of time. The dataset has many challenging scenarios, *e.g.*, people interact with each other while merging to a group and then splitting; people leave a camera view and then come back after a long time; people stay in a cluttered scenario with heavy occlusions. We use 8 cameras to test our tracking algorithm on a camera network while [1,2,4] used 5, 4, 2 cameras respectively.

4.2. Feature similarity

There are three equations, (12), (16) and (17), where calculation of feature similarity is involved. In the tracking scheme, the optimal solution $\mathbf{f}^*$ depends on the flow cost in Eq. (17), where the flow cost is defined as a negative logarithm of a feature similarity score S. In Eq. (12), the similarity score is defined as the motion affinity between the features of two tracklets in a single camera. In Eq. (16), feature similarity is necessary because the motion affinities between two tracklets are not reliable. The view and pose of a tracklet can change significantly in different cameras. It is possible that a merge/split event may happen in the blind area between the two camera views. Since the SNT can decide the group state of a tracklet in any camera, in general, two cases need to be considered when calculating feature similarities: two observations are recognized as individuals and at least one observation is recognized as a group.

Individual tracklet association. If two observations are recognized as two individuals, we use a linear combination of appearance in HSV space, histogram of oriented gradients (HoG) and pyramid of histograms of orientation gradients (PHoG) as the features of each observation. Firstly, a brightness transfer function (BTF) is applied to transform the appearance of a target from a camera to another. We adopt the method of [2] which can incrementally learn the BTF across cameras. With the transformed color features and the shape features (HoG and PHoG), the feature distance between individuals tracklets can be calculated by Bhattachayya distance which is denoted by $B(.)$. Thus, the feature similarity between two tracklets is $S\left(O_j^{C_b},O_i^{C_a}\right) \propto p_{tran}\cdot exp\left\{-B\left(O_j^{C_b},O_i^{C_a}\right)\right\}$, where p_{tran} denotes the transition probabilities between two cameras.

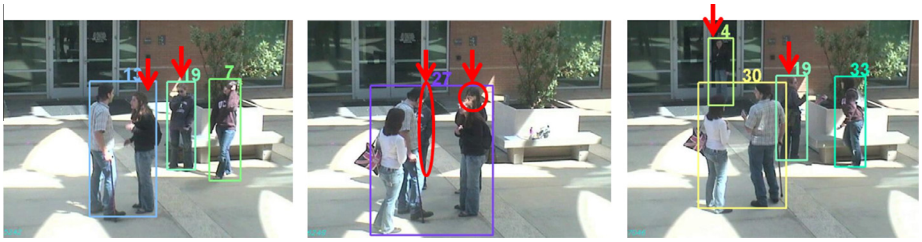

Fig. 5. Tracking recovering through clutter using individual-group switching mechanism. (a)–(c) are sorted by time. In (b), two persons who are marked by red cannot be detected because of the occlusion. However, their identities can be found before the group merging in (a) and after the group splitting in (c).

Table 2
Single camera tracking results.

	PR (%)	TF	IDC
Cam 16	67.0	3	1
Cam 17	89.2	8	4
Cam 20	93.0	2	1
Cam 21	74.6	9	9
Cam 27	85.5	7	6
Cam 31	81.4	15	8
Cam 36	89.6	7	4
Cam 37	83.1	10	5

Table 3
Multi-camera tracking results with both overlapping and non-overlapping views.

	TL (%)	XFG	XIDS
Scene 1	79.6	4	3
Scene 3	80.0	6	4
Scene 4	81.5	4	4
Scene 5	77.3	5	3
Scene 6	79.0	5	4
Scene 7	78.7	5	3
Scene 22	75.6	7	6
Scene 23	76.3	8	7
Scene 24	79.9	6	6
Scene 25	77.1	8	7

Group tracklet association. If at least one of the two observations in two cameras is a group, two sub-cases should be considered. The first sub-case is that there is no merge/split event in the blind area between two cameras. This means that the same group appears in these two linked cameras. The second sub-case is that a merge/split event occurs in the blind area which makes the targets in these two cameras different.

To associate group observations G_i and G_j, feature similarity between two group regions should be calculated. Similar to [29,30], the appearance and statistical properties of the group region are represented by a covariance descriptor. Given the feature points $\{x_i\}_{i=1,\ldots,N_{PI}}$ where N_{PI} is the number of pixels in the group region, the covariance descriptor is

$$V_G = \frac{1}{N_{PI}-1}\sum_{i=1}^{N_{PI}}(f_i-\mu_G)(f_i-\mu_G)^T, \tag{18}$$

where μ_G is the mean feature vector of these N_{PI} feature vectors. The feature similarity between two groups in cameras a and b is computed based on their covariance descriptors V^{G_i} and V^{G_j} between two cameras a and b, *i.e.*,

$$S\left(G_i^{C_a}, G_j^{C_b}\right) = 1 - \sqrt{\sum\nolimits_i ln^2\lambda_i\left(V_{G_i}^{C_a}, V_{G_j}^{C_b}\right)}, \tag{19}$$

where $\{\lambda_i\}$ are the generalized eigenvalues of the two group covariance matrices $C_{G_i}^{C_a}$ and $C_{G_j}^{C_b}$.

If $S\left(G_i^{C_a}, G_j^{C_b}\right)$ is larger than a given threshold, we consider $G_j^{C_b}$ is a good candidate match to $G_i^{C_a}$. Otherwise, the two groups are recognized as different. If no group matches $G_i^{C_a}$, it is highly possible that a merge/split event occurred in the blind area. The number of individuals in two groups are estimated according to the detections in the group. The algorithm to associate group tracklets is listed in Algorithm 1. Given a group tracklet $\mathcal{G}_i$ in C_a, the goal of this algorithm is to find a correspondence $\mathcal{T}_j^{C_b}$ which is the best match to $\mathcal{G}_i^{C_a}$. Note that we use $\mathcal{T}_j$ to represent a candidate tracklet because the candidate can be an individual tracklet. Even if $\mathcal{T}_j$ is an individual tracklet, the same method can be applied.

Algorithm 1. Group tracklet association.

```
Input: Tracklets $\mathcal{T}_i^{C_a}$ whose group state is 1.
if at least one good group candidate is found to match $\mathcal{G}_i^{C_a}$ in all entry/exit
zones linked to the zone of $\mathcal{T}_i^{C_a}$ then
    Apply Eq. 19;
else
    Search all candidates from valid linked entry/exit zones in other
    cameras;
    Divide the region of $\mathcal{G}_i^{C_a}$ and $\mathcal{T}_j^{C_b}$ into a set of small parts where each
    part has a size of a standard individual;
    Find the feature distance between each small part of the two tracklets
    and select the best one;
    if all feature similarities are smaller than a given threshold then
        Recognize the tracklet $\mathcal{G}_i^{C_a}$ as a birth/death tracklet;
    else
        Find the tracklet $\mathcal{T}_j^{C_b}$ with the largest similarity to $\mathcal{G}_i^{C_a}$ and assign
        them the same ID;
    end
end
Output: The best match to $\mathcal{G}_i^{C_a}$;
```

4.3. Group detection evaluation

VideoWeb dataset is very rich in interaction activities. Every video sequence has around 8 merge events and 7 split events on average. Our training samples are video frames of crowded scenes, which contain both individuals and groups. The group and individual labels of the bounding boxes are manually annotated according to the ground truth for training. During the training, SVM is trained upon the pedestrians' distance features developed based on their bounding boxes. Every camera view is trained separately because the average sizes of pedestrians are different in different camera views. For every camera view, we selected 4 video sequences for training. Each video sequence contains individuals and pedestrian groups where the ground truth is available. In the testing phase, among 80 video sequences, we obtain 3188 groups totally. After tracklet association, 1513 groups are obtained. This means that 1675 groups are associated using the optimization framework in Section 3. The number of people detected per group

Table 4
Comparison of the proposed SNT algorithm with some existing methods.

	TL (%)	XFG	XIDS
[2]	63.0	150	111
[3]	65.0	135	129
Proposed method	78.5	58	47

varies from 2 to 8 depending upon the scenario. In Fig. 5, an example shows that though individuals cannot be recognized in the high clutter, their identities can be recovered after the group splits into individuals.

4.4. Tracking performance evaluation

There are no standard evaluation metrics for a multi-camera tracking scheme, though single camera tracking evaluation metrics have been studied for years [47]. We adopt the evaluation metrics in [48] to fairly evaluate our results because it introduces both single camera and multi-camera evaluation metrics. In the single-camera evaluation, PR (precision), IDC (ID change) and TF (track fragmentation) are adopted. Three metrics are used for the multi-camera tracking evaluation: TL (trajectory length), XFrag (crossing fragments in two different cameras) and XIDS (crossing ID-switches in two cameras).

To the best of our knowledge, there is no existing tracking work based on this public dataset. Comparisons with other multi-camera tracking methods like [1,4] are not feasible as the goals of the algorithms presented there were very different from ours and, therefore, cannot be applied to the Videoweb dataset. Thus, we evaluate our results with respect to the ground truth. We also show comparisons with [2,3]. The average results on all the ten testing scenarios are provided in Table 4.

We first run a particle filter to associate detections into tracklets. After tracklet fusion across overlapping views, the SNT is run to obtain single camera tracking results, the average of which are shown in Table 2. According to the results, our tracker is able to obtain a good precision while the number of track fragmentation and ID change are small. The precision of camera 16 is low as there are many misdetections in this camera view.

The multiple interacting targets tracking results in a camera network are shown in Table 3. We present the results on every scenario we worked on to better represent the performance of our camera network tracker on different scenes. A high trajectory

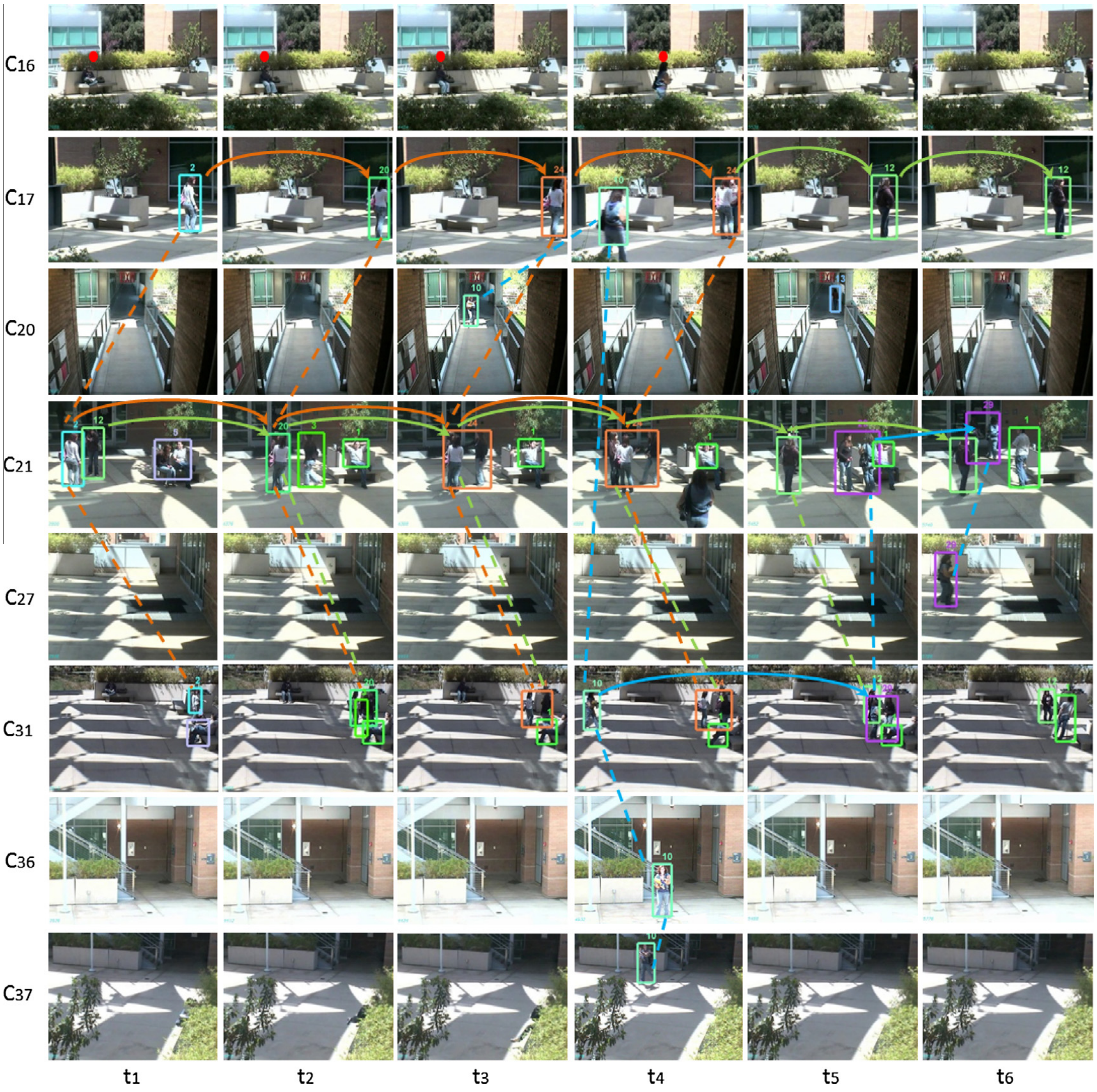

Fig. 6. Representative multiple interacting targets tracking results in a camera network. The horizontal axis represents the time step while the vertical axis illustrates 8 different cameras. Different colors of lines with arrows represent how a target moves over time, while straight dotted lines show the same target observed by different cameras.

length with low crossing fragments and crossing ID switches in each scene shows the advantage of the proposed algorithm.

Representative tracking results are provided in Fig. 6. Among the 6 time steps, there are totally 7 persons appearing in the scene. Typical scenarios are listed below.

- t_1: Person 2 and 12, who can be observed by C_{21}, stay individually.
- t_2: Person 2 and 12 interact with each other and person 12 cannot be observed by any camera. The SNT outputs the ID of these two persons as a group 20. The person who stays within the group 5 at t_1 leaves the group at t_2 with the ID 3.
- t_3: Person 3 interacts with the group 20, and finally merges to a new group 24. The track of person 10 starts in C_{20}.
- t_4: Group 24 can be observed by C_{17}, C_{21}, C_{31} at t_3 and t_4. Person 10 walks into the view of $C_{16}, C_{17}, C_{21}, C_{31}, C_{36}$ and C_{37} at t_4, in which full body detections in C_{17}, C_{31}, C_{36} and C_{37} are obtained by a person detector.
- t_5 and t_6: Person 10 and person 3 merge into group 29 at t_5, and walk into the view of C_{27} at t_6.

The results show that the track IDs of the same target in overlapping camera are the same, *e.g.*, person 10 and person 12. Another observation is that though only parts of a group (an individual) are observed in some camera, the weighted voting scheme is able to find the correct group state. For instance, at t_3, the group 24 can be fully observed in C_{21} and C_{31}. However, only one person can be observed by C_{17} while a group ID is assigned to this person. A similar example is illustrated in group 29 at t_6. A failure case is represented by red dots where missing detections lead to tracks loss.

5. Conclusion

We have addressed the problem of tracking in a camera network where there are individuals and groups interacting. A structural SVM model was proposed to discriminate between individuals and groups. Observations in overlapping cameras were fused, and associations between those in a camera network were calculated. Formulating the problem of the camera network tracking as a network flow model, a standard linear program problem is obtained. An efficient K-shortest paths algorithm is used to perform robust multi-object tracking. Experimental results on a very challenging public dataset show the robust performance of our tracking system.

Acknowledgment

This work has been partially supported by ONR United States Grant N00014-09-C-0388.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cviu.2015.01.002.

References

[1] K. Chen, C. Lai, Y. Hung, C. Chen, An adaptive learning method for target tracking across multiple cameras, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2008.
[2] A. Gilbert, R. Bowden, Tracking objects across cameras by incrementally learning inter-camera color calibration and patterns of activity, in: Proc. European Conf. Computer Vision, 2006.
[3] B. Song, A.K. Roy-Chowdhury, Robust tracking in a camera network: a multi-objective optimization framework, IEEE J. Sel. Top. Signal Process. 2 (4) (2008) 582–596.
[4] C.-H. Kuo, C. Huang, R. Nevatia, Inter-camera association of multi-target tracks by on-line learned appearance affinity models, in: Proc. European Conf. Computer Vision, 2010.
[5] B. Yang, R. Nevatia, Multi-target tracking by online learning of non-linear motion patterns and robust appearance models, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2012.
[6] B. Song, T.-Y. Jeng, E. Staudt, A.K. Roy-Chowdhury, A stochastic graph evolution framework for robust multi-target tracking, in: Proc. European Conf. Computer Vision, 2010.
[7] M. Liem, D.M. Gavrila, Multi-person tracking with overlapping cameras in complex, dynamic environments, in: Proc. British Machine Vision Conf., 2009.
[8] M. Bredereck, X. Jiang, M. Korner, J. Denzler, Data association for multi-object tracking-by-detection in multi-camera networks, in: Proc. Int'l Conf. Distributed Smart Cameras, 2012.
[9] J.W. Suurballe, Disjoint paths in a network, Networks 4 (2) (1974) 125–145.
[10] J. Hershberger, M. Maxel, S. Suri, Finding the k shortest simple paths: a new algorithm and its implementation, ACM Trans. Algor. 3 (4) (2007) 45:1–45:19.
[11] G. Denina, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, B. Varda, VideoWeb dataset for multi-camera activities and non-verbal communication, in: Distributed Video Sensor Networks, Springer, 2010.
[12] O. Javed, K. Shafique, Z. Rasheed, M. Shah, Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views, Int. J. Comput. Vis. 109 (2) (2008) 146–162.
[13] V. Kettnaker, R. Zabih, Bayesian multi-camera surveillance, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 1999.
[14] O. Javed, Z. Rasheed, K. Shafique, M. Shah, Tracking across multiple cameras with disjoint views, in: Proc. Int'l Conf. Computer Vision, 2003.
[15] D. Makris, T. Ellis, J. Black, Bridging the gaps between cameras, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2004.
[16] A.R. Dick, M.J. Brooks, A stochastic approach to tracking objects across multiple cameras, in: Proc. Australian Conference on Artificial Intelligence, 2004.
[17] K. Tieu, G. Dalley, W. Grimson, Inference of non-overlapping camera network topology by measuring statistical dependence, in: Proc. Int'l Conf. Computer Vision, 2005.
[18] X. Wang, K. Tieu, W. Grimson, Correspondence-free activity analysis and scene modeling in multiple camera views, IEEE Trans. Pattern Anal. Mach. Intell. 1 (1) (2009) 1–17.
[19] S. Zhang, E. Staudt, T. Faltemier, A. Roy-Chowdhury, A camera network tracking (CamNeT) dataset and performance baseline, in: Proc, IEE, Winter Conf. Applications of Computer Vision, 2015.
[20] X. Chen, K. Huang, T. Tan, Direction-based stochastic matching for pedestrian recognition in non-overlapping cameras, in: Proc. IEEE Int'l Conf. on Image Processing, 2011.
[21] S. Khan, M. Shah, A multiview approach to tracking people in crowded scenes using a planar homography constraint, in: Proc. European Conf. Computer Vision, 2006.
[22] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multicamera people tracking with a probabilistic occupancy map, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 267–282.
[23] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, O.I. Camps, Dynamic subspace-based coordinated multicamera tracking, in: Proc. Int'l Conf. Computer Vision, 2011.
[24] A. Kamal, J.A. Farrel, A.K. Roy-Chowdhury, Information consensus for distributed multi-target tracking, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2013.
[25] S. Pellegrini, A. Ess, L.V. Gool, Improving data association by joint modeling of pedestrian trajectories and groupings, in: Proc. European Conf. Computer Vision, 2010.
[26] Z. Qin, C.R. Shelton, Improving multi-target tracking via social grouping, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2012.
[27] X. Chen, Z. Qin, L. An, B. Bhanu, An online elementary grouping model for multi-target tracking, in: Proc, IEE, Int'l Conf. Computer Vision and Pattern Recognition, 2014.
[28] L. Bazzani, M. Cristani, V. Murino, Decentralized particle filter for joint individual-group tracking, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2012.
[29] W.S. Zheng, S. Gong, T. Xiang, Associating groups of people, in: Proc. British Machine Vision Conf., 2009.
[30] Y. Cai, V. Takala, M. Pietikainen, Matching groups of people by covariance descriptor, in: Proc. Int'l. Conf. Pattern Recognition, 2010.
[31] R. Mazzon, F. Poiesi, A. Cavallaro, Detection and tracking of groups in crowd, in: Proc. IEEE Conf. Advanced Video and Signal Based Surveillance, 2013.
[32] R. Vezzani, D. Baltieri, R. Cucchiara, People reidentification in surveillance and forensics: a survey, ACM Comput. Surv. 46 (2) (2013) 280–293.
[33] V. Rabaud, S. Belongie, Counting crowded moving objects, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2006.
[34] G.J. Brostow, R. Cipolla, Unsupervised Bayesian detection of independent motion in crowds, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2006.
[35] W. Ge, R.T. Collins, R.B. Ruback, Vision-based analysis of small groups in pedestrian crowds, IEEE Trans. Pattern Anal. Mach. Intell. 34 (5) (2012) 1003–1016.
[36] S. Zhang, A. Das, C. Ding, A. Roy-Chowdhury, Online social behavior modeling for multi-target tracking, in: Proc, IEE, Int'l Conf. Computer Vision and Pattern Recognition Workshops, 2013.

[37] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27:1–27:27.

[38] T. Joachims, T. Finley, C.J. Yu, Cutting-plane training of structural SVMs, Mach. Learn. 77 (1) (2009) 27–59.

[39] C. Desai, D. Ramanan, C. Fowlkes, Discriminative models for multi-class object layout, in: Proc. Int'l Conf. Computer Vision, 2009.

[40] Y. Zhu, N. Nayak, A. Roy-Chowdhury, Context-aware modeling and recognition of activities in video, in: Proc, IEE, Int'l Conf. Computer Vision and Pattern Recognition, 2013.

[41] Y. Zhu, N. Nayak, A. Roy-Chowdhury, Context-aware activity recognition and anomaly detection in video, IEEE J. Sel. Top. Signal Process 7 (1) (2013) 91–101.

[42] L. Zhang, Y. Li, R. Nevatia, Global data association for multi-object tracking using network flows, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2008.

[43] J. Berclaz, F. Fleuret, E. Türetken, P. Fua, Multiple object tracking using K-shortest paths optimization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (9) (2011) 1806–1819.

[44] A.A. Butt, R.T. Collins, Multi-target tracking by Lagrangian relaxation to min-cost network flow, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2013.

[45] H. Pirsiavash, D. Ramanan, C.C. Fowlkes, Globally-optimal greedy algorithms for tracking a variable number of objects, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2011.

[46] Global Optimization Techniques, 2014. <http://www.mat.univie.ac.at/~neum/glopt/techniques.html>.

[47] T. Nawaz, F. Poiesi, A. Cavallaro, Measures of effective video tracking, IEEE Trans. Image Process. 23 (1) (2014) 376–388.

[48] ICPR 2012 Contest on People Tracking in Wide Baseline Camera Networks, 2012. <http://www.wide-baseline-camera-network-contest.org/?page_id=50>.