Multi-Target Multi-Camera Tracking by Tracklet-to-Target Assignment

Yuhang He, Xing Wei, Xiaopeng Hong, *Member, IEEE*, Weiwei Shi, and Yihong Gong, *Fellow, IEEE*

***Abstract*— This paper focuses on the Multi-Target Multi-Camera Tracking task (MTMCT), which aims at tracking multiple targets within a multi-camera network. As the trajectory of each target is inherently split into multiple sub-trajectories (namely local tracklets) in a multi-camera network, a major challenge of MTMCT is how to accurately match the local tracklets generated within each camera across different cameras and generate a complete global trajectory for each target, *i.e.*, the *cross-camera tracklet matching* problem. We solve the cross-camera tracklet matching problem by TRACklet-to-Target Assignment (TRACTA), and propose the Restricted Non-negative Matrix Factorization (RNMF) algorithm to compute the optimal assignment solution that meets a set of constraints, which should be in force in practice. TRACTA can correct the tracking errors caused by occlusions and missed detections in local tracklets, and produce a complete global trajectory for each target across all the cameras. Moreover, we also develop an analytical way of estimating the total number of targets in the camera network, which plays an important role to compute the tracklet-to-target assignment. Experimental evaluations and ablation studies on four MTMCT benchmark datasets show the superiority of the proposed TRACTA method.**

***Index Terms*— Multi-camera racking, multi-target tracking, non-negative matrix factorization, tracklet association.**

I. Introduction

MULTI-TARGET Multi-Camera Tracking (MTMCT) aims at inferring a complete, cross-camera trajectory for each target in a multi-camera network. It has a wide range of applications in pedestrian monitoring [1], [2], video surveillance [3]–[5], in-store customer behavior analysis [6], city traffic control [7], and crowd behavior analysis [8], [9].

To date, most MTMCT methods are comprised of the following two phases: the local tracklet generation phase that tracks each detected target, and generates its local trajectory within a single camera; and the cross-camera tracklet matching phase that matches local tracklets across all the cameras to generate a complete trajectory for each target within the entire multi-camera network. Despite years of efforts, MTMCT remains a largely unsolved problem due to the following reasons: 1) background clutter and object occlusions cause erroneous like incomplete local tracking results under a single camera [10], [11]; 2) Dramatic variations in visual appearance and ambient environment caused by different viewpoints from different cameras make the cross-camera local tracklet matching extremely difficult [12], [13]; 3) The number of cameras in which each target appears, and the number of targets within the entire multi-camera network are both unknown, and thus the inference of the global trajectory of each target becomes even more challenging [14]. The problem 3) in the above list is further compounded when a tracker mistakenly generates plural local tracklets for the same target within a single camera.

In the past decades, a great number of methods have been developed to track multiple targets under a single camera [15]–[19]. In this paper, we mainly focus on the cross-camera tracklet matching problem. Most existing methods tackle this problem using the tracklet-to-tracklet matching approach. Methods in [20], [21] match tracklets between every two adjacent cameras until tracklets across all the cameras are matched. Some other methods [22]–[24] iteratively match all the local tracklets across cameras using greedy matching or hierarchical clustering. To reduce the search space and improve the matching efficiency, additional mechanisms such as candidate pruning using camera topology [23] and adaptive attribute selection [24] are developed in the matching process. There are also research works [25]–[28] that attempt to find a global solution for tracklet matching using the Bayesian formulation [25], [29] or the graph models [2], [26], [27], [30]. The global trajectory of each target is obtained by maximizing a posterior probability or finding a network flow from the source node to the sink one.

There are two problems associated with such tracklet-to-tracklet matching scheme. First, since different targets appear in a different number of cameras, the number of local tracklets associated with each target is different and unknown. Therefore, it is hard to determine how many tracklets should be matched and grouped into a global trajectory. Second, to make the matching result viable in practice, the matching of local tracklets should satisfy the *matching consistency principle*, which ensures that the matched tracklets are fully connected (e.g., if tracklet A is matched to tracklet B and tracklet B is matched to tracklet C, then tracklet A should be matched to tracklet C), and there is no connection between different tracklet sets (e.g., if certain tracklets are grouped together to form a global trajectory for a specific target, then different sets

Manuscript received July 26, 2019; revised December 21, 2019; accepted February 25, 2020. Date of current version March 23, 2020. This work was supported in part by the National Major Project of China under Grant 2017YFC0803905 and in part by the National Key Research and Development Program under Grand 2019YFB1312000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shuicheng Yan. *(Yuhang He and Xing Wei are co-first authors.) (Corresponding author: Xiaopeng Hong.)*

Yuhang He, Xing Wei, Xiaopeng Hong, and Yihong Gong are with the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: hyh1379478@stu.xjtu.edu.cn; xingxjtu@gmail.com; hongxiaopeng@mail.xjtu.edu.cn; ygong@mail.xjtu.edu.cn).

Weiwei Shi is with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China (e-mail: wshi@xaut.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.2980070

of grouped tracklets should be mutually exclusive). With the tracklet-to-tracklet matching scheme, it is difficult to impose such matching consistency principle in a systematic way. There have been research efforts in the literature that attempt to impose various constraints such as sparsity constraint, time conflict constraint, etc [27], [31]–[33], none of them has accomplished the matching consistency principle in an elegant way.

To address the above problems, we propose a novel MTMCT method that formulates the cross-camera tracklet matching problem as a TRACklet-to-Target Assignment (TRACTA) problem, where each tracklet is assigned to a unique target and the optimal assignment is computed using the Restricted Non-negative Matrix Factorization (RNMF) algorithm. More specifically, we first generate a local tracklet set $\mathcal{T}_i^L$ for each camera i, $i = 1, \ldots, M$, and then compute the pairwise tracklets similarity matrix $\mathbf{S} \in [0, 1]^{N \times N}$, where N is the number of tracklets and each element $\mathbf{S}(u, v)$ represents the similarity score between tracklets u and v. Moreover, we denote by $\mathbf{A} \in \{0, 1\}^{N \times K}$ the tracklet-Target ID (TID) assignment matrix from N tracklets to K targets, where each element $\mathbf{A}(u, v)$ takes binary values of 1 or 0, which correspond to the assignment and non-assignment of TID v to tracklet u, respectively. In TRACTA, each tracklet is imposed to be associated with one and only one target, *i.e.*, $\forall u, \sum \mathbf{A}(u, :) = 1$, therefore the assignment from tracklets to targets should be a mapping (the *mapping assignment constraint*). In the Appendix, we will detail the proof of how the mapping assignment constraint meets the matching consistency principle. On this basic, the optimal tracklet-TID assignment matrix $\mathbf{A}^*$ is computed using the proposed RNMF algorithm that guarantees the solution to meet the matching consistency principle. Using $\mathbf{A}^*$, and by integrating information from all the local tracklet sets $\mathcal{T}_i^L$, we can correct the tracking errors caused by occlusions and missed object detections in $\mathcal{T}_i^L$ and generate a complete, more accurate trajectory for each target across multiple cameras. We also develop a method to analytically estimate the number of targets K in the entire camera network. Experimental evaluations and comparative studies using four benchmark datasets show the superiority of the proposed TRACTA method.

In summary, the main contributions of this paper include:

- We provide a novel MTMCT solution, which formulates the cross-camera tracklet matching problem as a tracklet-to-target assignment problem.
- We propose the restricted non-negative matrix factorization algorithm to compute the optimal tracklet-TID assignment matrix $\mathbf{A}^*$.
- We propose an analytical way to estimate the total number of targets K across all the cameras.

The remaining of the paper is organized as follows. Section II introduces the related works. Section III provides an overview of the proposed method. Section IV elaborates the main components of the proposed method, including the restricted non-negative matrix factorization algorithm for cross-camera tracklet matching, the algorithm for estimating the total number of targets K across all the cameras, and the global trajectory generation algorithm. In Section V, we evaluate the proposed method in four public available datasets and compare it with state-of-the-art methods. Section VI concludes this paper.

II. Related Work

There is a large literature on visual tracking [18], [34]–[41]. In this section, we discuss the most relevant research works to the MTMCT task.

A large number of research works for the MTMCT task assume overlapping Fields of Views (FOV) between cameras. Fleuret *et al.* [14] project targets into a probabilistic occupancy map (POM), and combine occupancy probabilities with color and motion attributes in the tracking process. Berclaz *et al.* [42] formulate the problem of tracking targets in the probabilistic occupancy map as an integer programming problem, and compute the optimal solution by using the k-shortest paths (KSP) algorithm.

There are also research studies that tackle the MTMCT problem by using graph models. Hofmann *et al.* [26] and Shitrit *et al.* [43] find the detection correspondences across all the views using a constrained min-cost flow graph. Leal *et al.* [27] formulate the MTMCT problem as a multi-commodity network flow problem, where each node in the network denotes a detection and edges denote their connections, and use the branch-and-price (B&P) algorithm to link detections into trajectories.

In recent years, we have seen increasing efforts to solve the MTMCT problem using the two-step approach: 1) generating local tracklets of all the targets within each camera; 2) matching local tracklets that belong to the same target across all the cameras. For the first step, the generation of local tracklets within a single camera is referred to as the single camera multi-target tracking task [18], [44]–[47], which has been intensively studied in computer vision and pattern recognition communities [3], [48]–[50]. Due to the impressive progress of object detection techniques [51]–[53], tracking-by-detection [16], [54]–[57] has become the mainstream approach for multi-target tracking in recent years. For the second step, various cross-view data association methods have been proposed to match local tracklets across different cameras. Hu *et al.* [58] and Eshel and Moses [59] utilize epipolar geometry to find the correspondences across cameras. They project lines or points of targets into a reference plane and consider those targets with intersections in the reference plane as belonging to the same identity. Xu *et al.* [24] propose a Hierarchical Composition of Tracklet (HCT) framework to match local tracklets by utilizing multiple cues of targets such as appearances and their ground plane locations. Bredereck *et al.* [20] propose a Greedy Matching Association (GMA) method, which iteratively matches local tracklets obtained from different cameras one by one. Xu *et al.* [25] solve the tracklet matching problem using a Bayesian formulation, and propose the Spatio-Temporal Parsing (STP) structure to prune matching candidates by exploiting semantic attributes of targets.

There is another category of research works that aim at the MTMCT task for non-overlapping FOVs (namely

wide-area MTMCT). Research studies in this category attempt to match local tracklets across different views by exploiting different information such as appearance cues [60], [61], motion pattern [26], camera topology [62], and human semantic attribute [30]. Chen *et al*. [29] estimate affinities of targets in different views by using the Piecewise Major Color Spectrum Histogram Representation (PMCSHR), which generates a major color histogram for each target using the online k-means clustering method. Caiand and Medioni [61] propose a Relative Appearance Context (RAC) to distinguish targets moving in proximity. There are research works [63], [64] attempt to match local tracklets between every two adjacent cameras, and the method in HFUTDSP [21] uses the Hungarian algorithm match tracklets across all the cameras by iterative pairwise matching. In the Ristani and Tomasi Tracking (RTT) work [12], the matching of local tracklets in different views is formulated as a binary integer program problem, and the connection of tracklets is obtained by correlation clustering. Zhang *et al*. [22] produce a robust feature representation for each target using Convolutional Neural Networks (CNNs) and introduce a Feature Re-Ranking mechanism (FRR) to find correspondences among tracklets. Chen *et al*. [2] find the matching of local tracklets using Equalized Graph Models (EGM), where each node in the graph denotes a tracklet and edges are the connections of tracklets. The trajectory of each target is obtained by finding a min-cost flow from the source node to the sink one. Lee *et al*. [62] propose an Incremental Camera Link Model (ICLM) to match local tracklets using different cues of targets adaptively, and target representations are online updated to relieve the visual variations in different views. There are also research works [30] integrating social grouping in MTMCT, which associate local tracklets using a Conditional Random Field (CRF) model, and find the matching of local tracklet by minimizing the unary and pairwise energy costs in the model. To reduce the search space and generate more accurate matching results, the EGM, ICLM and CRF methods prune infeasible matching candidates according to camera topology [23] and motion continuity. Moreover, a series of constraints such as the coupling and non-overlap constraints [26] and the spatial-temporal constraint [33] are proposed to meet the matching consistency principle.

The proposed TRACTA method is applicable to cameras with or without overlapping FOVs. The most related research work to TRACTA is the Multi-Dimensional Assignment (MDA) method [28], [65]. However, the difference between MDA and TRACTA is clear, and can be summarized as follows. In MDA, all the tracklet matching hypotheses are enumerated, and each hypothesis is a tracklet set. The goal of MDA is to find the most likely hypothesis that meets a series of constraints, and tracklets in the same hypothesis are assigned to the same target. To the contrary, in TRACTA, we directly assign each tracklet to a unique target and compute the optimal assignment using the RNMF algorithm.

It is worth mentioning that there are research studies using non-negative matrix factorization (NMF) for single object tracking [66], [67] and multi-target tracking [68]. Nonetheless, in these methods, NMF is a feature extraction module to generate a robust representation for each target, and the inputs of the NMF algorithm are feature vectors. In TRACTA, however, the proposed RNMF algorithm is used as a solver to find the optimal assignment from tracklets to targets, and the input of the RNMF algorithm is the similarity matrix of tracklets. To the best of our knowledge, this is the first work that formulates the cross-camera tracklet matching problem by tracklet-to-target assignment, and uses the RNMF algorithm to obtain the optimal assignment.

III. Overview of the Proposed Method

We use a calligraphic capital letter to denote a dataset, a bold capital letter to denote a matrix, and a regular capital or lowercase letter to denote a variable or a constant.

The proposed TRACTA method consists of four major modules: 1) local tracklet generation module, 2) tracklet similarity measurement module, 3) cross-camera tracklet matching module, and 4) global trajectory generation module. Figure 1 depicts the proposed framework. The input to the framework is M video clips from M cameras. Module 1 takes each input video i, $i = 1, \ldots, M$, and applies the tracking-by-detection method in [45] to generate all local tracklets for the video clip. We denote by $\mathcal{T}^L$ the entire local tracklet set:

$$\mathcal{T}^L = \{\mathcal{T}_1^L, \mathcal{T}_2^L, \ldots, \mathcal{T}_M^L\},$$

where $\mathcal{T}_i^L$ denotes the local tracklet set for the i-th camera.

Module 2 takes $\mathcal{T}^L$ as the input to compute the similarity score between every two local tracklets using their appearance, motion, location information and camera topology whenever they are available. It then outputs a tracklet similarity matrix $\mathbf{S}$ that is composed of sub-matrices $\mathbf{S}_{ij}$. Each element in $\mathbf{S}_{ij}$ is the similarity score of a pair of tracklets from $\mathcal{T}_i^L$ and $\mathcal{T}_j^L$.

The third module in our framework takes the tracklet similarity matrix $\mathbf{S}$ as the input, and applies the proposed RNMF algorithm to assign a unique TID $k \in 1, \ldots, K$ to each tracklet in $\mathcal{T}_i^L \in \mathcal{T}^L$, $i = 1, \ldots, M$. RNMF computes the optimal solution for the TID assignment to the local tracklets. Here, K is the total number of targets among all the M video clips. In Section IV-B, we will describe how to infer the value of K, and in Section V-E, we will reveal how different K values will affect the tracking accuracy and running speed of the TRACTA method.

Using the output from Module 3, Module 4 generates the global trajectory for each of the M cameras. We use $\mathcal{T}^G$ to denote the entire global trajectory set:

$$\mathcal{T}^G = \{\mathcal{T}_1^G, \mathcal{T}_2^G, \ldots, \mathcal{T}_M^G\},$$

where $\mathcal{T}_i^G$ denotes the global trajectory set for the i-th camera. There are two differences between $\mathcal{T}_i^L$ and $\mathcal{T}_i^G$. First, tracklets in $\mathcal{T}_i^L \in \mathcal{T}^L$ are independent of tracklets in $\mathcal{T}_j^L \in \mathcal{T}^L$, $i, j = 1, \ldots, M, i \neq j$, meaning that the k-th tracklets from $\mathcal{T}_i^L$ and $\mathcal{T}_j^L$, respectively, do not necessarily correspond to the same target. In contrast, in $\mathcal{T}^G$, the k-th tracklets from each of $\mathcal{T}_i^G$, $i = 1, \ldots, M$, all correspond to the same target. Second, in $\mathcal{T}^L$, the number of tracklets in each $\mathcal{T}_i^L$ is different. Some $\mathcal{T}_i^L \in \mathcal{T}^L$ may contain less than K tracklets due to the fact that some targets may never appear in camera i, or because of missed detections of certain targets from the camera. It is

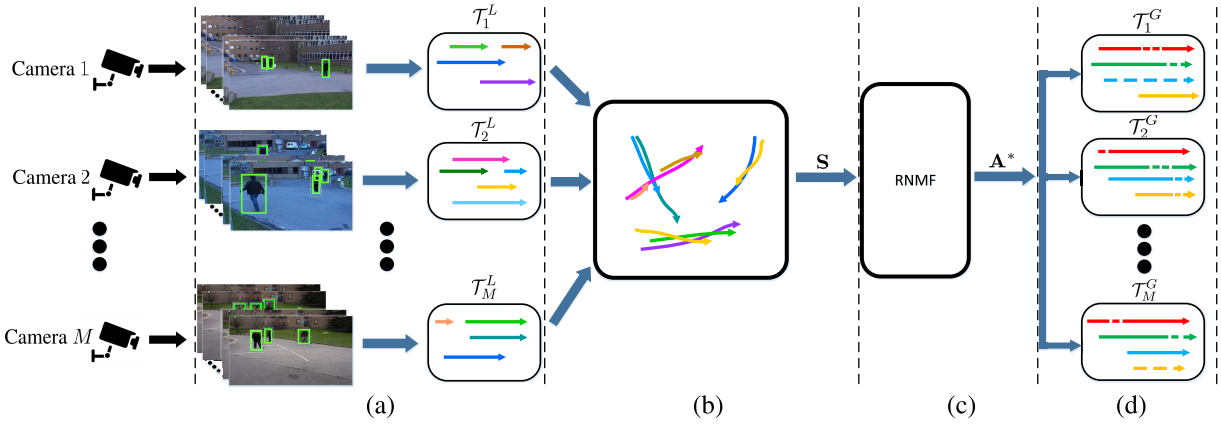


Fig. 1. Overview of the proposed TRACTA framework. The input to the framework is M video clips from M cameras. (a) The local tracklet generation module. (b) The tracklet similarity measurement module. (c) The cross-camera tracklet matching module. (d) The global trajectory generation module.

also probable that some $\mathcal{T}_i^L \in \mathcal{T}^L$ may contain more than K tracklets because some targets were occluded for a while and reappeared later, causing the object tracker to generate multiple tracklets for the same target. By contrast, in $\mathcal{T}^G$, every $\mathcal{T}_i^G$ contains the same number of K tracklets. Using the output from Module 3, the missed tracklets in certain $\mathcal{T}_i^L \in \mathcal{T}^L$ are recovered, and the multiple tracklets belonging to the same target are reconnected and repaired to a complete tracklet. By discovering the correspondences between, and integrating the useful information of all the tracklets from different cameras, our proposed TRACTA method strives to generate a complete, more robust trajectory for every target captured by the multiple cameras.

IV. Description of Key Algorithms

The entire algorithm of the TRACTA is shown in Algorithm 1. The key techniques of this paper lie in the steps 5-11, *i.e.*, the RNMF algorithm for calculating the optimal tracklet-TID assignment matrix $\mathbf{A}^*$ (described in Section IV-A), a method for estimating the total number of targets K (described in Section IV-B), and the global trajectory generation algorithm (described in Section IV-C). In Section IV-D, we also provide implement details of computing similarity scores between local tracklets. We discuss the differences between the TRACTA and the tracklet-to-tracklet matching methods in Section IV-E.

Algorithm 1 Multi-Target Multi-Camera Tracking by Tracklet-to-Target Assignment

Input: Video set $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_M\}$ collected from M cameras.
Output: Global trajectory set $\mathcal{T}^G$.
1: **for** each video $\mathcal{D}_i \in \mathcal{D}$ **do**
2: Generate a local tracklet set $\mathcal{T}_i^L$ using single camera multi-object tracking methods, such as [41].
3: **end for**
4: Calculate a tracklet similarity matrix $\mathbf{S} \in [0,1]^{N\times N}$ using Eq. (18)-Eq. (20), where N is the total number of local tracklets.
5: Estimate the total number of targets K using Eq. (11)
6: Initialize tracklet-TID assignment matrix $\mathbf{A}' \in \mathbb{R}_+^{N\times K}$ with random positive values.
7: **repeat**
8: Update $\mathbf{A}'$ according to Eq.(8)
9: **until** converge
10: Obtain $\mathbf{A}^*$ by setting the maximum element of each row of $\mathbf{A}$ to 1, and the other elements in the row to 0.
11: Generate the global trajectory set $\mathcal{T}^G$ using Eq. (13)-Eq. (16)

A. Cross-Camera Tracklet Matching Using Restricted Non-Negative Matrix Factorization

The proposed RNMF algorithm aims to assign a unique TID to each tracklet in $\mathcal{T}_i^L \in \mathcal{T}^L$, $i = 1, \ldots, M$. Let N_i be the number of tracklets in $\mathcal{T}_i^L$, and $N = \sum_{i=1}^{M} N_i$ is the total number of tracklets. We define the pairwise tracklets similarity matrix $\mathbf{S}_{ij}$ of dimension $N_i \times N_j$ between $\mathcal{T}_i^L$ and $\mathcal{T}_j^L$, where each element $\mathbf{S}_{ij}(u, v)$ represents the similarity score between tracklet u in $\mathcal{T}_i^L$ and tracklet v in $\mathcal{T}_j^L$. $\mathbf{S}_{ij}(u, v)$ takes a real value in [0, 1], and Section IV-D provides a brief description of the similarity score computation. Using $\mathbf{S}_{ij}$ as the building block, we form the full tracklets similarity matrix $\mathbf{S}$ of dimension $N \times N$:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \cdots & \mathbf{S}_{1M} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{M1} & \cdots & \mathbf{S}_{MM} \end{bmatrix}. \quad (1)$$

Furthermore, for each $\mathcal{T}_i^L \in \mathcal{T}^L$, we define the tracklet-TID assignment matrix $\mathbf{A}_i$ of dimension $N_i \times K$, where K is the total number of targets in all the M video clips, and each element $\mathbf{A}_i(u, v)$ takes binary values of 1 or 0, which correspond to the assignment, non-assignment of TID v to tracklet u in $\mathcal{T}_i^L$, respectively. The value of K is estimated by the proposed method, which is described in Section IV-B. Similarly, using $\mathbf{A}_i$ as the building block, the full tracklet-TID

assignment matrix $\mathbf{A}$ of dimension $N \times K$ is formed as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_M \end{bmatrix}. \tag{2}$$

Matrix $\mathbf{A}$ has several properties. First, $\mathbf{A}$ is a binary matrix. Second, each row $\mathbf{A}(u,:)$ can have only one non-zero element, because each tracklet can only belong to one target, and the assignment from tracklets to TIDs is a mapping, *i.e.*, the mapping assignment constraint. Third, each column $\mathbf{A}(:,v)$ can have multiple non-zero elements, because multiple tracklets from either the same camera or different cameras can belong to the same target. Finally, if tracklets u and v have a high similarity score, *i.e.*, $\mathbf{S}(u,v) \to 1$, meaning that u and v might belong to the same target, then we should have $\mathbf{A}(u,:)\mathbf{A}(v,:)^T = 1$. On the contrary, if $\mathbf{S}(u,v) \to 0$, we expect $\mathbf{A}(u,:)\mathbf{A}(v,:)^T = 0$.

The last property of $\mathbf{A}$ implies that the two matrices $\mathbf{S}$ and $\mathbf{A}\mathbf{A}^T$ have a strong linkage. In other words, if $\mathbf{S}(u,v)$ is large, then $\mathbf{A}\mathbf{A}^T(u,v)$ should also be large, and vice versa. This is not a surprise by looking into the semantic nature of the matrix $\mathbf{A}$. In fact, each row $\mathbf{A}(u,:)$ of $\mathbf{A}$ can be considered as a feature vector that indicates the probabilities of tracklet u belonging to each of the K targets. Therefore, $\mathbf{A}\mathbf{A}^T$ can be interpreted as another similarity matrix indicating pairwise similarities among the tracklets in $\mathcal{T}^L$.

The above observation suggests that $\mathbf{A}\mathbf{A}^T$ has the same semantic meaning as $\mathbf{S}$, *i.e.*, $\mathbf{A}\mathbf{A}^T \to \mathbf{S}$. This allows us to formulate the tracklet matching problem as the following math problem:

Given the full tracklet similarity matrix $\mathbf{S}$*, compute the full tracklet-TID assignment matrix* $\mathbf{A}$ *that minimizes the L-2 distance* $\|\mathbf{S} - \mathbf{A}\mathbf{A}^T\|^2$ *and satisfies all the properties described above.*

Translating this statement into equations, we have

$$\mathbf{A}^* = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{S} - \mathbf{A}\mathbf{A}^T\|^2, \tag{3}$$

$$\text{s.t} \quad \mathbf{A} \in \{0,1\}^{N \times K}, \tag{4}$$

$$\mathbf{A}\mathbf{1}_K = \mathbf{1}_N, \tag{5}$$

where $\mathbf{1}_K$ and $\mathbf{1}_N$ denote all-ones matrices of size $K \times 1$ and $N \times 1$, respectively. Eq. (4) ensures that $\mathbf{A}$ is binary, and Eq. (5) enforces that the assignment from tracklets to targets is a mapping ($\forall u, \sum \mathbf{A}(u,:) = 1$).

Under the binary constraint on $\mathbf{A}$, the objective function in Eq. (3) is hard to optimize. To make the problem tractable, we relax all the elements in $\mathbf{A}$ to take non-negative real values, and denote the relaxed matrix by $\mathbf{A}'$. Then, the problem of optimizing Eq. (3) turns into solving the following optimization problem:

$$\mathbf{A}'^* = \underset{\mathbf{A}' \geq 0}{\operatorname{argmin}} \ \|\mathbf{S} - \mathbf{A}'\mathbf{A}'^T\|^2 + \alpha\|\mathbf{A}'\mathbf{1}_1 - \mathbf{1}_2\|^2, \tag{6}$$

where α is the penalty weight.

We denote by J the objective function in Eq. (6), and the gradient of J with respect to $\mathbf{A}'$ is:

$$\frac{\partial J}{\partial \mathbf{A}'} = -4\mathbf{S}\mathbf{A}' + 4\mathbf{A}'\mathbf{A}'^T\mathbf{A}' + 2\alpha\mathbf{A}'\mathbf{1}_1\mathbf{1}_1^T - 2\alpha\mathbf{1}_2\mathbf{1}_1^T. \tag{7}$$

Using the gradient in Eq. (7), we can use the following updating rule to optimize $\mathbf{A}'$ according to [69]:

$$\mathbf{A}' \leftarrow \mathbf{A}' \odot \text{sqrt}([4\mathbf{S}\mathbf{A}' + 2\alpha\mathbf{1}_2\mathbf{1}_1^T] \oslash [4\mathbf{A}'\mathbf{A}'^T\mathbf{A}' + 2\alpha\mathbf{A}'\mathbf{1}_1\mathbf{1}_1^T]), \tag{8}$$

where $\text{sqrt}(\cdot)$ calculates the square root of every element of a matrix, $\odot$ and $\oslash$ denote the element-wise multiplication and division, respectively. Similar to [69], objective function J can be proven to be non-increasing under this updating rule, and the convergence of the iteration is guaranteed.

The optimal solution $\mathbf{A}'^*$ to Eq. (6) is a non-negative matrix. We can compute its binary version by simply setting the maximum element of each row in $\mathbf{A}'^*$ to 1, and the other elements in the row to 0. Obviously, the binarized version of $\mathbf{A}'^*$ satisfies both the constraints of Eq. (4) and Eq. (5), and hence can be used as the optimal solution to Eq. (3).

B. Estimation of K

Consider the binary tracklets similarity matrix $\mathbf{S}^* \in \{0,1\}^{N \times N}$ for N tracklets, where $\mathbf{S}^*(u,v) = 1$ if tracklets u and v belong to the same TID, and $\mathbf{S}^*(u,v) = 0$ otherwise. According to the derivations in Section IV-A, $\mathbf{S}^*$ can be decomposed as $\mathbf{S}^* = \mathbf{Q}\mathbf{Q}^T$, where $\mathbf{Q} \in \{0,1\}^{N \times D}$ is the binary tracklet-TID assignment matrix between N tracklets and D TIDs, and each row of $\mathbf{Q}$ has only one non-zero element. The Singular Value Decomposition (SVD) of $\mathbf{Q}$ is $\mathbf{Q} = \mathbf{U}\Sigma\mathbf{V}^T$, where Σ is an $N \times D$ rectangular diagonal matrix. The Eigen Value Decomposition (EVD) of the following symmetric square matrices can be written as:

$$\mathbf{Q}\mathbf{Q}^T = \mathbf{U}\Sigma\Sigma^T\mathbf{U}^T \in \{0,1\}^{N \times N}, \tag{9}$$

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{V}\Sigma^T\Sigma\mathbf{V}^T \in \{0,1\}^{D \times D}. \tag{10}$$

It can be easily proven that $\mathbf{Q}\mathbf{Q}^T$ and $\mathbf{Q}^T\mathbf{Q}$ have the same set of non-zero eigenvalues. Because each column $\mathbf{Q}(:,v)$ is a binary vector indicating the affiliations between TID v and N tracklets, and the TID$\leftrightarrow$tracklet affiliations are mutually exclusive, meaning that columns of $\mathbf{Q}$ are mutually perpendicular, $\mathbf{Q}^T\mathbf{Q}$ must be diagonal, and its eigenvalues are the diagonal elements of $\mathbf{Q}^T\mathbf{Q}$. Furthermore, each diagonal element $\mathbf{Q}^T\mathbf{Q}(v,v)$ is a non-negative integer that counts how many tracklets in N belong to TID v. Therefore, the number of real targets can be estimated by counting the number of eigenvalues of $\mathbf{Q}^T\mathbf{Q}$ (or equivalently $\mathbf{Q}\mathbf{Q}^T$) that are larger than 1.

Based on the above derivations, we estimate K using the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ of $\mathbf{S}$ as follows:

$$K = \sum_{i=1}^{N} \delta(\lambda_i \geq \beta), \tag{11}$$

where $\delta(cond) = 1$ if $cond$ is true, and equals zero otherwise. The summing operation in Eq. (11) counts the number of eigenvalues larger than β. The influence of β is analyzed in Section V-E, based on which we set $\beta = 0.9$ for a good trade-off between the accuracy and the speed.

C. Global Trajectory Generation

Let $\mathcal{T}_i^L(u)$ and $\mathcal{T}_j^G(v)$ denote the tracklet u in $\mathcal{T}_i^L$ and the tracklet v in $\mathcal{T}_j^G$, respectively. A tracklet $\mathcal{T}_i^L(u) \in \mathcal{T}_i^L$ is composed of a series of tuples over a period of time:

$$\mathcal{T}_i^L(u) = \{(a_{iu}^L(t), b_{iu}^L(t), l_{iu}^L(t)) | t \in \pi_i(u)\}, \tag{12}$$

where $a_{iu}^L(t) \in \mathbb{R}^{800\times 1}$ denotes an appearance feature, $b_{iu}^L(t) \in \mathbb{R}^{4\times 1}$ denotes a bounding box, $l_{iu}^L(t) \in \mathbb{R}^{2\times 1}$ denotes a reference location of $\mathcal{T}_i^L(u)$ at time t, respectively, and $\pi_i(u)$ is the time index set of $\mathcal{T}_i^L(u)$.

For a particular target k, there exist four possible scenarios in a local tracklet set $\mathcal{T}_i^L$ from camera i, $i = 1, \ldots, M$:

1) $\sum \mathbf{A}_i(:, k) = 0$. When there is no tracklet assigned to the target k in camera i, *i.e.*, $\sum \mathbf{A}_i(:, k) = 0$, we consider the target k never appears in camera i, *i.e.*, the **Scenario 1**.

2) $\sum \mathbf{A}_i(:, k) = 1$ & $(\pi_i(u) = \bigcup \pi_j(v), \forall j, v$ if $\mathbf{A}_j(v, k) = 1)$. When there is only one tracklet assigned to the target k in camera i (assume the tracklet is $\mathcal{T}_i^L(u)$) and the time index set of tracklet $\mathcal{T}_i^L(u)$ is equal to the total existence time index set of the target k, *i.e.*, $\pi_i(u) = \bigcup \pi_j(v), \forall j, v$ if $\mathbf{A}_j(v, k) = 1$, we consider the target k is perfectly tracked by tracklet $\mathcal{T}_i^L(u)$ in camera i, *i.e.*, the **Scenario 2**.

3) $\sum \mathbf{A}_i(:, k) = 1$ & $(\pi_i(u) \subset \bigcup \pi_j(v), \forall j, v$ if $\mathbf{A}_j(v, k) = 1)$. When there is only one tracklet assigned to target k in camera i (assume the tracklet is $\mathcal{T}_i^L(u)$) and the time index set of tracklet $\mathcal{T}_i^L(u)$ is a subset of the total existence time index set of the target k, *i.e.*, $\pi_i(u) \subset \bigcup \pi_j(v), \forall j, v$ if $\mathbf{A}_j(v, k) = 1$, we consider tracklet $\mathcal{T}_i^L(u)$ is a fragment tracklet of target k in camera i, *i.e.*, the **Scenario 3**.

4) $\sum \mathbf{A}_i(:, k) > 1$. When there are multiple tracklets assigned to target k in camera i, *i.e.*, $\sum \mathbf{A}_i(:, k) > 1$, the target k in camera i is in the **Scenario 4**.

For the above four scenarios, we generate the global trajectory $\mathcal{T}_i^G(k)$ for target k under camera i as follows. For case (1), we simply set $\mathcal{T}_i^G(k) = \emptyset$. For case (2), as target k has been correctly detected and tracked at every time instance $t \in \pi$ under camera i, we can obtain $\mathcal{T}_i^G(k)$ by copying the corresponding local tracklet $\mathcal{T}_i^L(u)$:

$$\mathcal{T}_i^G(k) \doteq \{a_{iu}^L(t), b_{iu}^L(t), l_{iu}^L(t) | t \in \pi_i(u)\}.$$

As for case (3), it is clear that there are missed detections of target k for certain time instances under camera i. Nonetheless, we can still recover such missing detections under camera i by using the information of the tracklets from the other cameras that correspond to target k. Assume that $\mathcal{T}_i^L(u)$ is the tracklet assigned to target k in camera i. Let $\tau_i(k)$ be the set of time instances when target k is lost in camera i, and $\mathcal{A}_k^L(t)$, $\mathcal{B}_k^L(t)$, and $\mathcal{L}_k^L(t)$ be the sets of appearance features, bounding boxes, and locations at time t, respectively, collected from tracklets that correspond to target k under all the other cameras except camera i. Then, we have:

$$\mathcal{T}_i^G(k) = \{a_{ik}^G(t), b_{ik}^G(t), l_{ik}^G(t) | t \in \pi_i(k)\}, \tag{13}$$

$$a_{ik}^G(t) = \begin{cases} \text{avg}(\mathcal{A}_k^L(t)), & \text{if } t \in \tau_i(k), \\ a_{iu}^L(t), & \text{otherwise.} \end{cases} \tag{14}$$

$$b_{ik}^G(t) = \begin{cases} \phi(\mathcal{B}_k^L(t)), & \text{if } t \in \tau_i(k), \\ b_{iu}^L(t), & \text{otherwise.} \end{cases} \tag{15}$$

$$l_{ik}^G(t) = \begin{cases} \text{avg}(\mathcal{L}_k^L(t)), & \text{if } t \in \tau_i(k), \\ l_{iu}^L(t), & \text{otherwise.} \end{cases} \tag{16}$$

where $\text{avg}(\cdot)$ is the function that averages the values in the input set, and $\phi(\cdot)$ is the function that estimates the bounding box based on the input set using the algorithm in [70].

For case (4), we take the union of the tracklets in $\mathcal{T}_i^L$ that correspond to target k.

$$\mathcal{T}_i^G(k) = \bigcup_{u \in \omega_i(k)} \mathcal{T}_i^L(u), \tag{17}$$

where $\omega_i(k)$ is the index set of tracklets in $\mathcal{T}_i^L$ that correspond to target k. If there are holes in $\mathcal{T}_i^G(k)$, meaning that missed detection and tracking occurs at certain time instances $t \in \pi$, then we fill these holes using Eq. (13) to Eq. (16) developed for case (3) to make $\mathcal{T}_i^G(k)$ a complete one.

D. Similarity Measure of Local Tracklets

For a local tracklet $\mathcal{T}_i^L(u)$, we obtain its reference plane projection by projecting foot point of each bounding box into the global coordinate system of the reference plane, where the reference plane locations of the foot points are calculated by using the homography correspondence between camera i and the reference plane [71].

We compute the similarity score between two local tracklets $\mathcal{T}_i^L(u)$ and $\mathcal{T}_j^L(v)$ using appearance, motion, location and camera topology whenever they are available. Let $\mathcal{A}_{iu}^L$, $\mathcal{B}_{iu}^L$ and $\mathcal{L}_{iu}^L$ be the sets of appearance features, bounding boxes, and locations (in the coordinate system of the reference plane) of $\mathcal{T}_i^L(u)$, respectively.

When two local tracklets u and v are from the same camera (*i.e.*, $i = j$), we calculate their similarity score according to their appearance and motion similarities in the image plane, which can be written as:

$$\mathbf{S}_{ii}(u, v) = \eta_1 \cdot \psi_{app}(\mathcal{A}_{iu}^L, \mathcal{A}_{iv}^L) + (1 - \eta_1) \cdot \psi_{mot}(\mathcal{B}_{iu}^L, \mathcal{B}_{iv}^L), \tag{18}$$

where η_1 is a weight parameter, $\psi_{app}(\cdot, \cdot) \in [0, 1]$ and $\psi_{mot}(\cdot, \cdot) \in [0, 1]$ are the similarity measurement functions that calculate the appearance similarity and motion consistency in the image plane based on the input sets using algorithms in [72], respectively. Moreover, guided by the intuition that any target will not be observed multiple times in one image frame, we set the similarity between two different tracklets that overlap in time to 0.

When two local tracklets u and v are from different cameras (*i.e.*, $i \neq j$) with overlapping FOVs, we compute their similarity score according to their appearance similarity and location proximity in the reference plane (Eq. (19)). Note that if u and v belong to the same target, they are usually close to each other in the reference plane.

$$\mathbf{S}_{ij}(u, v) = \eta_2 \cdot \psi_{app}(\mathcal{A}_{iu}^L, \mathcal{A}_{jv}^L) + (1 - \eta_2) \cdot \psi_{loc}(\mathcal{L}_{iu}^L, \mathcal{L}_{jv}^L), \tag{19}$$

where η_2 is a weight parameter, and $\psi_{loc}(\cdot,\cdot) \in [0,1]$ is the function that calculates the location proximity in the reference plane using the algorithm in [24].

When tracklets are from cameras with non-overlapping FOVs, we calculate their similarity according to appearance similarity and camera topology as follows:

$$\mathbf{S}_{ij}(u,v) = \psi_{app}(\mathcal{A}_{iu}^{L}, \mathcal{A}_{jv}^{L}) \cdot \psi_{top}(\mathcal{T}_{i}^{L}(u), \mathcal{T}_{j}^{L}(v)), \quad (20)$$

where $\psi_{top}(\mathcal{T}_{i}^{L}(u), \mathcal{T}_{j}^{L}(v)) = 1$ if the matching is feasible according to camera topology [23], and 0 for otherwise.

E. Differences Between TRACTA and Tracklet-to-Tracklet Matching

The existing tracklet-to-tracklet matching methods attempt to directly determine whether tracklets should be matched together or not, while the TRACTA infers the connection of tracklets according to the assignment from tracklets to targets, where the tracklets assigned to the same target are matched together and the others are not. More concretely, let a symmetric matrix $\mathbf{X} \in \{0,1\}^{N\times N}$ denote the matching of N local tracklets across all the cameras, where $\mathbf{X}(u,v) = 1$ if the tracklets u and v are matched and $\mathbf{X}(u,v) = 0$ for otherwise, the differences between TRACTA and the tracklet-to-tracklet matching methods come from the following aspects: 1) The TRACTA is targeted at finding an optimal tracklet-TID assignment matrix $\mathbf{A}$, rather than finding a solution on $\mathbf{X}$. 2) The matching of local tracklets can be obtained by $\mathbf{X} = \mathbf{A}\mathbf{A}^T$, and since $K \ll N$ when there are multiple local tracklets of the same target, finding a solution in $\mathbf{A}^{N\times K}$ is more efficient than $\mathbf{X}^{N\times N}$. 3) As the number of tracklets for different targets is different and unknown, it is uncertain that how many tracklets should be matched with a specific tracklet, and the norm of the rows of $\mathbf{X}$ is varying. By contrast, in TRACTA, each tracklet should be assigned to a unique target, and the norm of the rows of $\mathbf{A}$ is a constant of 1. 4) The matching consistency principle can be naturally satisfied in TRACTA on the basis of tracklet-target assignment, while it is quite difficult to impose such principle in the tracklet-to-tracklet matching scheme. In the Appendix, we will detail the proof of how the matching consistency principle is satisfied in TRACTA.

V. Experiments

In this section, we evaluate the proposed TRACTA method using four publicly available benchmark datasets and compare it with representative MTMCT methods. Note that many MTMCT methods included in the comparative study require that the input videos be filmed by cameras with the overlapping FOVs, while our TRACTA method is applicable to input videos with or without overlapping FOVs. Moreover, as the proposed RNMF algorithm plays a central role in cross-camera tracklet matching, we design experiments to compare it to other tracklet matching algorithms. We also conduct experiments to reveal how different values of the meta-parameter β influence the tracking accuracy and running speed of the proposed method.

We evaluate all the methods using the widely adopted CLEAR metrics [73] to evaluate all the methods under consideration, including MOTA ↑, MOTP ↑, MODA ↑, MODP ↑, as well as six additional metrics MT ↑ (the mostly tracked), ML ↓ (the mostly lost), IDS ↓ (ID switches), Precision ↑, Recall ↑ and MCTA ↑ [2]. We show these metrics for each method in comparison whenever they are reported in the references. Here, ↑ indicates the higher score is better and ↓ is the opposite. The parameters used to calculate the CLEAR metrics are set according to the *strict criteria* in [24], [25]. In our experiments, these metrics are computed using the *devkit* toolkit provided in [74].

A. Preprocessing and Overall Settings

In the local tracklet generation module, we use the Faster-RCNN algorithm [52] to obtain detections of targets, where the NMS and confidence threshold are set to 0.3 and 0.8, respectively; Then we extract appearance feature of each detection using the CNN model in [75]. The reference plane location of each bounding box is obtained by projecting the foot point of the bounding box into the reference plane using the homography correspondence in [71]. Detections between different frames are linked into local tracklets using the tracking algorithm in [45], where the minimum detection height, maximum missing age and tracklet buffer length are set to 40, 30 and 100, respectively.

For our RNMF algorithm, we initialize $\alpha = 10$ and gradually decrease it to 1 along with iterations, and set $\beta = 0.9$ according to the experimental result in Section V-E. η_1 and η_2 are chosen on validation sequence, and fixed to 0.5 and 0.3 for all the experiments, respectively.

B. Datasets

We use four public MTMCT benchmark datasets,[1] PETS09-S2L1, CAMPUS, EPFL, MCT, for performance evaluations. Among these four datasets, the first three are filmed by cameras with overlapping FOVs, while the last one is filmed without overlapping FOVs. We use the ground truth annotations of the PETS09-S2L1, CAMPUS, EPFL datasets provided by [24] [2] and use the annotations of MCT provided by [2]. A brief description of the four datasets is provided as follows:

PETS09-S2L1 [8]: It films about 10 targets entering and passing through a footpath using 7 cameras, and we use the videos captured by 4 cameras with overlapping FOVs. The videos are shot at 7 fps with a resolution of 720 × 576, and there are 795 frames in each view.

CAMPUS [24]: This dataset consists of 4 subsets of videos shooting four different locations, *i.e.*, Garden1, Garden2, Auditorium and Parkinglot. There are 15 ∼ 25 pedestrians in the four video subsets, respectively. Each subset records 3 ∼ 4 minutes of videos using 4 cameras with overlapping

[1]DukeMTMC [1] dataset is not included because it is no longer available since April 2019, according to the owner of DukeMTMC via email communication.

[2]https://bitbucket.org/merayxu/multiview-object-tracking-dataset

TABLE I
EVALUATION RESULTS (%) ON PETS09-S2L1 DATASET

Method	MOTA↑	MOTP↑	MODA↑	MODP↑
KSP [38]	80	57	81	58
B&P [25]	72	53	74	55
GMA [18]	73	78	74	78
HCT [22]	**89**	73	**90**	74
Ours	87.5	**79.2**	88.2	**79.7**

TABLE II
EVALUATION RESULTS (%) ON EPFL DATASET

Sequence	Method	MOTA↑	MOTP↑	MODA↑	MODP↑
Terrace1	POM [12]	58	63	60	64
	KSP [38]	67	58	68	59
	HCT [22]	72	71	74	72
	Ours	**81.0**	**79.5**	**84.2**	**79.6**
Passage-way	POM [12]	32	62	32	64
	KSP [38]	40	57	40	59
	HCT [22]	44	71	45	72
	Ours	**52.1**	**77.5**	**52.8**	**77.7**
Basket-ball	POM [12]	45	56	47	57
	KSP [38]	56	54	58	57
	HCT [22]	60	68	61	68
	Ours	**64.3**	**72.5**	**65.3**	**72.7**

FOVs, and each video is shot at 30 fps at a resolution of 1920×1080.

EPFL [14]: We use the 3 video subsets, Terrace1, Passageway and Basketball, that are captured by 4 cameras with overlapping FOVs. There are about 10 pedestrians in the three subsets, respectively. Each video was filmed at head height with 25 fps frame rate and 360×288 resolution.

MCT [2]: This dataset consists of 4 subsets, *i.e.*, Dataset1, Dataset2, Dataset3, Dataset4, that are all filmed without overlapping FOVs. There are 3,3,4,5 cameras capturing 235, 255, 14, 49 targets in the four subsets, respectively. All the videos are shot at 25 fps at a resolution of 320×240, and there are about 30,000 frames for each subset.

C. Comparisons With State of the Arts

1) Results on Datasets With Overlapping FOVs: We first compare the proposed TRACTA method with the representative MTMCT methods that require overlapping FOVs among the input videos, including K-Shortest Path (KSP) [42], Probabilistic Occupancy Map (POM) [14], Hierarchical Composition of Tracklet (HCT) [24], Brand-and-Price (B&P) [27], Spatio-Temporal Parsing (STP) [25], and Greedy Matching Association (GMA) [20]. Evaluations of these methods are conducted using the three benchmark datasets PETS09-S2L1, CAMPUS, EPFL that all consist of videos with overlapping FOVs.

As shown in Table I, the proposed TRACTA method significantly outperforms KSP, B&P, GMA on all the four metrics by up to 26.2%. Compared to the state-of-the-art HCT method, it is slightly lower on MOTA and MODA (1.5% and 1.8%, respectively), but significantly higher on MOTP and MODP (6.2% and 5.7%, respectively). Table II and III reveal the evaluation results on the EPFL and CAMPUS datasets, respectively. From Table II, it is clear that on the EPFL dataset, TRACTA outperforms all the representative methods on all the metrics, with up to 10% improvement on certain metrics compared to the state-of-the-art method HCT. Similarly, on the CAMPUS dataset, TRACTA outperforms all the comparative methods on most of the metrics, and is the top performer in terms of the overall performance accuracies.

2) Results on the Dataset With Non-Overlapping FOVs: Here, we use the MCT dataset to compare TRACTA with the representative MTMCT methods that do not require overlapping FOVs among the input videos but use the visual appearance and camera topology information, including RAC [61], EGM [2], HFUTDSP [21], PMCSHR [29], and ICLM [62].

Table IV shows that TRACTA significantly outperforms all the comparative methods with a large margin in terms of all the metrics. Moreover, in the challenging subsets Dataset3 and Dataset4, which contain over four cameras in complex scenarios, TRACTA leads to an significant improvement by at least 5.3% and 17.1% in terms of MOTA, and 16.6% and 17.2% in terms of MCTA on these two subsets, respectively.

In summary, the proposed method can track multiple targets using cameras with both overlapping and non-overlapping FOVs in a unified manner, and achieves the state-of-the-art performance.

D. Comprehensive Comparisons to Modern Solvers

To investigate the effectiveness of the RNMF algorithm in matching local tracklets and the potential of $\mathcal{T}^G$ in improving the MTMCT performance, we compare RNMF with several tracklet matching solvers. Again, the comparisons are conducted for both the overlapping FOV and the non-overlapping FOV settings using the EPFL and MTC datasets, respectively. Moreover, we conduct additional experiments on these two datasets to study the influence of using different target information in the RNMF algorithm.

Table V shows the comparison results for the overlapping FOV setting on the EPFL dataset, where the proposed RNMF algorithm is compared with five modern tracklet matching solvers as described in Section II, *i.e.*, GMA [20], Eshel and Moses [59], Hu *et al*. [58], HCT [24] and STP [25]. The FPS ↑ column shows the processing speed in terms of frame per second.[3] The *baseline* method outputs the local tracklet set $\mathcal{T}^L$. The methods marked with '*' output the global trajectory sets which are obtained by substituting the RNMF algorithm with the respective cross-camera tracklet matching solvers in the proposed pipeline (Module (c) in Figure 1), while keeping all the other parts unchanged. The method *Ours-M(A)+A(L)* outputs the global trajectory sets $\mathcal{T}^G$ by using ψ_{mot} (or ψ_{app}) in Eq.(18) and ψ_{app} (or ψ_{loc}) in Eq.(19), respectively, while the method *Ours-full* generates $\mathcal{T}^G$ using all the tracklet attributes.

We make the following important observations from Table V:

- By reconnecting the split local tracklets and repairing the missed detections in $\mathcal{T}^L$ (*baseline*), the generated $\mathcal{T}^G$ (*Ours-full*) achieves significant improvements in term of MOTA (by 11.3%, 9.8%, 11.5% on these three subsets, respectively).

[3]Tested on a PC with an Nvidia Titan X GPU and an Intel Core I7 CPU.

TABLE III
EVALUATION RESULTS ON CAMPUS DATASET

Sequence	Method	MOTA (%) ↑	MOTP (%) ↑	MODA (%) ↑	MODP (%) ↑	MT (%) ↑	ML (%) ↓
Garden1	POM [12]	22.4	64.2	24.5	64.3	0	43.7
	KSP [38]	28.1	62.0	30.5	62.1	6.3	25.1
	HCT [22]	49.0	71.9	49.3	72.0	**31.3**	6.3
	STP [23]	57	**75**	–	–	–	–
	Ours	**58.5**	74.3	**59.2**	**74.4**	30.6	**1.6**
Garden2	POM [12]	14.0	63.8	16.5	63.9	14.3	7.1
	KSP [38]	21.9	61.6	24.4	61.8	14.3	**0**
	HCT [22]	25.8	71.6	27.8	71.7	**33.3**	11.1
	STP [23]	30	75	–	–	–	–
	Ours	**35.5**	**75.3**	**37.1**	**75.4**	16.9	11.3
Auditorium	POM [12]	16.2	61.0	17.9	61.2	16.7	16.6
	KSP [38]	17.6	59.3	19.5	59.5	22.2	16.7
	HCT [22]	20.6	69.2	20.8	69.3	33.3	**11.1**
	STP [23]	24	72	–	–	–	–
	Ours	**33.7**	**73.1**	**36.2**	**73.4**	**37.3**	20.9
Parkinglot	POM [12]	11.0	60.0	11.7	69.3	0	53.3
	KSP [38]	14.0	58.4	14.7	58.5	0	46.7
	HCT [22]	24.1	66.2	24.5	66.4	6.7	26.6
	STP [23]	28	68	–	–	–	–
	Ours	**39.4**	**74.9**	**42.1**	**75.0**	**15.5**	**10.3**

TABLE IV
EVALUATION RESULTS ON MCT DATASET

Subset	Method	MCTA (%) ↑	MOTA (%) ↑	MOTP (%) ↑	Precision (%) ↑	Recall (%) ↑	IDS ↓
Dataset1	HFUTDSP [19]	28.1	57.6	46.6	71.1	65.3	5243
	EGM [2]	41.2	59.4	68.0	79.7	59.2	1888
	PMCSHR [27]	12.5	70.7	17.6	14.9	21.5	406
	RAC [57]	59.5	92.6	64.6	69.2	60.6	154
	ICLM [58]	61.2	87.3	68.1	77.2	60.9	112
	Ours	**70.8**	**94.9**	**85.2**	**92.7**	**92.6**	**71**
Dataset2	HFUTDSP [19]	28.2	54.7	49.2	74.6	36.7	5655
	EGM [2]	47.9	67.2	70.6	79.8	63.3	1985
	PMCSHR [27]	10.8	74.6	16.5	14.3	19.3	564
	RAC [57]	62.6	86.8	73.7	69.5	78.4	171
	ICLM [58]	67.7	88.3	76.6	83.3	70.9	123
	Ours	**83.7**	**93.4**	**85.9**	**95.5**	**95.4**	**60**
Dataset3	HFUTDSP [19]	3.6	21.1	15.2	33.4	9.9	644
	EGM [2]	18.6	27.0	64.7	82.1	53.5	525
	PMCSHR [27]	1.1	8.6	10.0	0.8	12.1	605
	RAC [57]	5.6	9.2	55.3	47.5	66.2	666
	ICLM [58]	37.2	53.2	69.1	66.0	72.6	228
	Ours	**53.8**	**58.5**	**75.4**	**75.2**	**91.3**	**144**
Dataset4	HFUTDSP [19]	0.06	28.1	20.1	77.2	12.1	732
	EGM [2]	28.4	35.8	71.1	83.6	61.9	3111
	PMCSHR [27]	2.1	27.1	7.4	6.1	9.4	1153
	RAC [57]	34.0	53.9	63.0	52.2	79.4	329
	ICLM [58]	54.3	62.5	86.8	**87.6**	86.0	189
	Ours	**71.5**	**79.6**	**90.0**	86.3	**96.0**	**70**

- With a high processing speed of about 20-45 FPS, RNMF significantly outperforms the other solvers, by at least 3.6% and 3.9% improvements in terms of MOTA and MODA, respectively, and achieves highly competitive results on MOTP and MODP.
- Using ψ_{app} in Eq. (18) and ψ_{loc} in Eq. (19) are more effective than the combinations of *Ours-M+A*, *Ours-M+L* and *Ours-A+A*, and using all the attributes (*Ours-full*) achieves the best performance.

To get more insights into the differences in using different information,[4] we illustrate tracking examples of the Basketball subset in Figure 2. There are several main observations. 1) Only using the ψ_{app} in Eq. (19) generates more ID switches due to similar appearance (*e.g.*, tracklets #4 and #7 in camera 1 switch their ID labels in *Ours-M+A* and *Ours-A+A*). 2) Only using ψ_{mot} in Eq. (18) leads to tracking failure due to motion blur (*e.g.*, under the camera 4 in *Ours-M+A* and *Ours-M+L*, tracklet #13 is failed to be assigned to the target #4 and the TRACTA generates a redundant bounding box for target #4). 3) Only using ψ_{loc} in Eq. (19) may leads to mismatches due to projection errors (*e.g.*, under the camera 1 in *Ours-M+L* and *Ours-A+L*, the tracklet #9 is failed to be assigned to the target #11 and the TRACTA generate a redundant bounding box for target #11). 4) By exploiting all the different information, *Ours-full* corrects these errors and achieves the best tracking performance.

Table VI shows the comparison results for the non-overlapping FOV setting on the MCT dataset, where no overlapping FOVs between cameras is available to the tracklet

[4]More illustrations of tracking results are available at https://github.com/GehenHe/TRACTA

TABLE V
COMPARISONS OF CROSS-CAMERA TRACKLET MATCHING SOLVERS ON EPFL DATASET

Subest	Method	MOTA(%) ↑	MOTP(%) ↑	MODA(%) ↑	MODP(%) ↑	FPS ↑
Terrace1	baseline	69.7	**80.7**	75.0	**80.9**	—
	GMA* [18]	46.2	78.5	50.7	79.8	**210**
	Eshel *et al.** [55]	65.5	80.3	73.1	80.5	0.2
	Hu *et al.** [54]	71.7	80.2	76.8	80.4	40
	HCT* [22]	74.2	79.7	81.8	79.8	5
	STP* [23]	77.0	79.6	82.0	79.8	5
	ours-M+A	72.0	79.6	77.7	79.8	30
	ours-M+L	75.9	79.4	80.1	79.7	30
	ours-A+A	74.6	79.5	78.7	79.7	30
	ours-A+L	77.2	79.5	81.5	79.7	30
	ours-full	**81.0**	79.5	**84.2**	79.6	30
Passageway	baseline	42.3	**78.6**	45.4	**78.7**	—
	GMA* [18]	26.0	78.5	33.1	78.6	**420**
	Eshel *et al.** [55]	34.8	78.5	43.0	78.6	0.2
	Hu *et al.** [54]	39.9	78.3	43.1	78.5	65
	HCT* [22]	47.5	77.4	48.2	77.5	15
	STP* [23]	48.5	78.4	48.9	78.4	10
	ours-M+A	43.5	77.5	44.9	78.1	45
	ours-M+L	46.1	78.1	46.5	78.3	45
	ours-A+A	44.9	77.7	46.8	77.7	45
	ours-A+L	49.2	77.5	52.5	77.9	45
	ours-full	**52.1**	77.5	**52.8**	77.7	45
Basketball	baseline	52.8	**73.4**	54.3	**73.5**	—
	GMA* [18]	35.1	73.1	36.5	73.2	**430**
	Eshel *et al.** [55]	47.5	72.9	50.3	73.1	0.2
	Hu *et al.** [54]	44.8	73.0	48.3	73.1	15
	HCT* [22]	54.4	72.9	55.8	73.1	0.5
	STP* [23]	55.5	72.9	57.9	73.2	0.3
	ours-M+A	57.3	73.2	58.7	73.3	20
	ours-M+L	60.6	72.9	62.0	73.1	20
	ours-A+A	57.6	72.4	59.1	72.9	20
	ours-A+L	62.3	72.7	64.1	73.1	20
	ours-full	**64.3**	72.5	**65.3**	72.7	20

TABLE VI
COMPARISONS OF CROSS-CAMERA TRACKLET MATCHING SOLVERS ON MCT DATASET

Method	Evaluation Metric	Dataset1	Dataset2	Dataset3	Dataset4	Average
baseline	IDS ↓	101	157	139	213	152.5
	MCTA ↑	69.7	61.5	8.6	16.8	39.2
HFUTDSP [19]	IDS ↓	86	141	40	155	105.5
	MCTA ↑	74.3	65.4	73.7	39.5	63.2
PMCSHR [27]	IDS ↓	112	167	44	110	108.3
	MCTA ↑	66.2	59.1	71.1	63.3	63.3
EGM [2]	IDS ↓	55	121	39	157	93
	MCTA ↑	83.5	70.3	74.2	38.5	66.6
CRF [28]	IDS ↓	54	81	51	70	64
	MCTA ↑	83.8	80.2	66.5	72.7	75.8
RAC [57]	IDS ↑	27	34	70	72	58.8
	MCTA ↑	91.5	91.3	51.6	70.5	76.3
RTT [10]	IDS ↓	80	127	77	137	105.3
	MCTA ↑	76.0	69.4	49.3	46.5	60.3
FRR [20]	IDS ↓	128	85	48	43	76
	MCTA ↑	55.3	76.1	64.1	82.1	69.4
ICLM [58]	IDS ↓	**13**	30	32	62	34.3
	MCTA ↑	96.1	92.7	79.0	75.8	85.9
ours-A	IDS ↓	18	29	24	27	24.5
	MCTA ↑	95.5	92.9	83.9	89.8	90.5
ours-A+T	IDS ↓	15	**20**	**18**	**22**	**18.8**
	MCTA ↑	**97.6**	**95.1**	**88.1**	**91.4**	**93.1**

matching solvers. The method *Ours-A* outputs the matching results using ψ_{app} in Eq. (20), and *Ours-A+T* outputs results using appearance together with camera topology. The results in Table VI show that the proposed RNMF algorithm achieves the top performance on MCTA in all the four subsets. Moreover, RNMF significantly decreases the number of IDS, and outperforms all the state-of-the-art solvers with a large margin in terms of MCTA (at least 9.3% and 7.2%, in the challenging subsets Dataset3 and Dataset4, respectively).

E. Influence of β

To study the influence of the meta-parameter β, we conduct experiments using the validation video subset from EPFL. The β value affects the estimation of K value, *i.e.*, the total

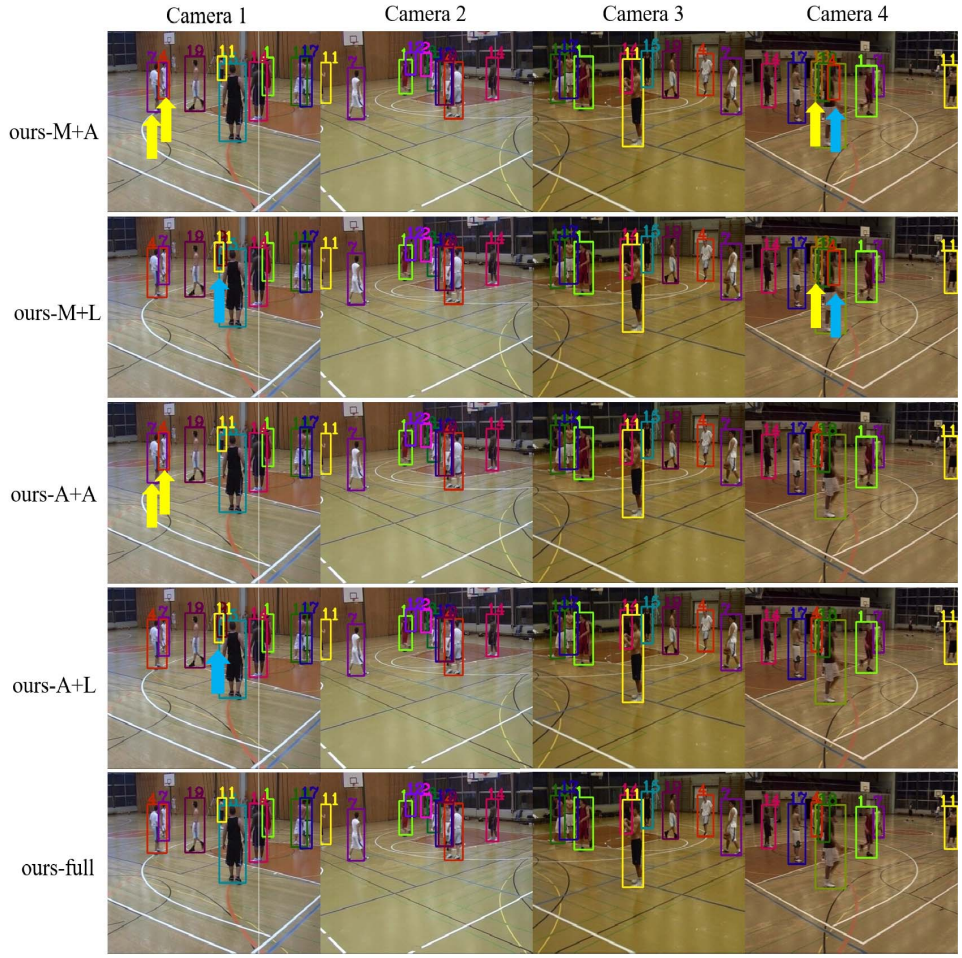


Fig. 2. Tracking examples of using different tracklet information. The number denotes the ID number of a tracklet. The yellow and blue arrows point to the ID switch and false positive detection, respectively. In *Ours-M+A*, tracklets #7 and #4 in camera 1 are mismatched due to the similar appearance, and in camera 4, the tracklet #13 failed to be assigned to target #4 and the TRACTA generates a redundant bounding box for target #4. In *Ours-M+L*, the tracklet #4 and #7 in camera 1 are correctly identified by utilizing the reference location, but tracklet #9 is failed to be assigned to target #11 due to the reference plane projection error and the TRACTA generates a redundant bounding box for target #11. In *Ours-A+A* and *Ours-A+L*, the mismatched tracklet and redundant bounding box caused by motion blur are fixed, while the ID switches caused by similar appearance and reference plane projection error still remained. *our-full* corrects these errors and achieve the best performance by exploiting all information.

number of TIDs in the input video clip set. The smaller the β is, the larger is the estimated K. As shown in Figure 3, when β increases from 0.5, the tracking accuracy keeps steady, whereas the processing speed keeps rising. When β exceeds 1, despite the continuous improvement in the processing speed, the tracking accuracy starts to deteriorate drastically. This is because the estimated K value starts to underestimate the genuine TID number in the input video clip set, which results in more and more incorrect tracklet-TID assignments in the global trajectory set $\mathcal{T}^G$.

On the other hand, when β becomes smaller than 1, the estimated K value starts to overestimate the genuine TID number, resulting in a larger tracklet-TID assignment matrix $\mathbf{A}$ and a slower processing speed. However, since the redundant columns in $\mathbf{A}$ will not be assigned any tracklets by the proposed RNMF algorithm, it makes no harms to the tracking accuracies.

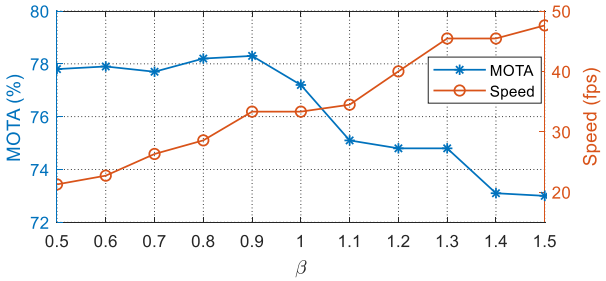


Fig. 3. Influence of β on tracking accuracy and speed.

F. Illustrated Differences Between $\mathcal{T}^L$ and $\mathcal{T}^G$

To get more insights of the differences between $\mathcal{T}^L$ and $\mathcal{T}^G$, and the effectiveness of $\mathcal{T}^G$ in generating a complete and accurate trajectory for each pedestrian across all the cameras, we show parts of the local tracklet sets $\mathcal{T}_1^L$ and $\mathcal{T}_2^L$ and the global trajectory sets $\mathcal{T}_1^G$ and $\mathcal{T}_2^G$ in the Terrace1 subset in Figure 4, which is challenging due to frequent occlusions. We can see that, (i) in $\mathcal{T}^G$, tracklets from different cameras that are belonging to the same pedestrian are matched together and are assigned with the same ID number. In $\mathcal{T}^L$, tracklets with the same ID number, *i.e.*, tracklets #1-#6 in $\mathcal{T}_1^L$ and

Fig. 4. Tracking examples of the local tracklet sets $\mathcal{T}_1^L$ and $\mathcal{T}_2^L$ and the global trajectory sets $\mathcal{T}_1^G$ and $\mathcal{T}_2^G$. The number in each bounding box denotes the tracklet's ID number. The red arrows and the yellow arrows point to missed detections and split tracklets belonging to the same pedestrian in the local tracklet sets, respectively. In $\mathcal{T}_1^L$ and $\mathcal{T}_2^L$, tracklets with the same ID number, *e.g.*, tracklet #1 in $\mathcal{T}_1^L$ and tracklet #1 in $\mathcal{T}_2^L$, are not necessarily corresponding to the same pedestrian, and there are missed detections (pointed by the red arrows) and split tracklets (pointed by the yellow arrows) due to the object occlusions. In $\mathcal{T}_1^G$ and $\mathcal{T}_2^G$, tracklets with the same ID number, *i.e.*, tracklets #1-#6 in $\mathcal{T}_1^G$ and $\mathcal{T}_2^G$, are corresponding to the same pedestrian with no exception. The tracking errors in $\mathcal{T}_1^L$ and $\mathcal{T}_2^L$ are corrected by reconnecting the split tracklets (*i.e.*, tracklets #2 and #3 in $\mathcal{T}_1^L$ are reconnected by tracklet #2 in $\mathcal{T}_1^G$, tracklets #1 and #7 and tracklets #2 and #9 in $\mathcal{T}_2^L$ are reconnected by global trajectories #3 and #4 in $\mathcal{T}_2^G$, respectively), and repairing the holes in local tracklets by estimating bounding boxes for the missed detections (*i.e.*, bounding box #4 in frame 1825 in $\mathcal{T}_1^G$, bounding box #3 in frame 1825 in $\mathcal{T}_2^G$, bounding boxes #2 in frames 1805 and 1825 in $\mathcal{T}_2^G$, and bounding boxes #4 in frames 1825, 1845 and 1865 in $\mathcal{T}_2^G$).

tracklets #1-#9 in $\mathcal{T}_2^L$, respectively, are not necessarily corresponding to the same pedestrian. In contrast, by using the RNMF algorithm to match local tracklets across views, tracklets in $\mathcal{T}_1^G$ and $\mathcal{T}_2^G$ that with the same ID number are all correspond to the same pedestrian. (ii) Tracking errors in $\mathcal{T}^L$ are corrected in $\mathcal{T}^G$ by reconnecting the split local tracklets and estimating the bounding boxes for the missed detections. More specifically, in $\mathcal{T}_1^L$, due to the occlusion in frame 1825, there is a missed detection of tracklet #3 in this frame (pointed by the red arrow), and the local tracklet #3 switches to #2 when it reappears in frame 1845, which results in multiple split tracklets in $\mathcal{T}_1^L$ corresponding to the same pedestrian (pointed by the yellow arrows). In $\mathcal{T}_2^L$, there are missed detections due to object interaction (pointed by the red arrows), *i.e.*, tracklet #1 is occluded by tracklet #3 in frame 1825, tracklet #2 is occluded by tracklets #3 and #7 in frames 1825, 1845 and 1865, and tracklet #6 is occluded by tracklet #5 in frames 1805 and 1825, and this leads to multiple split tracklets in $\mathcal{T}_2^L$ corresponding to the same pedestrian (pointed by the yellow arrows), *i.e.*, tracklet #1 switches to tracklet #7 in frame 1845 and tracklet #2 switches to tracklet #9 in frame 1865. In $\mathcal{T}_1^G$ and $\mathcal{T}_2^G$, these tracking errors are corrected by reconnecting the split tracklets and estimating the bounding boxes for the missed detections, *i.e.*, the split local tracklets #2 and #3 in $\mathcal{T}_1^L$ are reconnected by global trajectory #2 in $\mathcal{T}_1^G$, the local tracklets #1 and #7 and local tracklets #2 and #9 in $\mathcal{T}_2^L$ are reconnected by global trjectories #3 and #4, respectively, and the missed detections are recovered by the estimated bounding boxes in $\mathcal{T}_1^G$ and $\mathcal{T}_2^G$, *i.e.*, bounding box #4 in frame 1825 in $\mathcal{T}_1^G$, bounding box #3 in frame 1825, bounding boxes #2 in frames 1805 and 1825, and bounding boxes #4 in frames 1825, 1845 and 1865.

This is a strong evidence that the global trajectory set $\mathcal{T}^G$ is very effective for correcting errors in local tracklets, and the RNMF algorithm is efficient to matching local tracklets for MTMCT.

Finally, we discuss the remained challenges of the proposed method. For targets that share similar features, such as similar appearances and near locations, TRACTA may mismatch them due to the confusing features. To further improve the tracking

performance, more discriminative features are desired to be involved.

VI. Conclusion

In this paper, we propose a novel multi-target multi-camera tracking framework TRACTA that matches the local tracklets by tracklet-to-target assignment. To obtain the optimal assignment, we provide a solution using the restricted non-negative matrix factorization algorithm. Moreover, we also derive a method to estimate the total number of target K. TRACTA can correct the tracking errors in local tracklets and generate a unique and complete global cross-camera trajectory for each target. Experimental evaluations and comparison studies using four benchmark datasets show the superiority of the proposed method.

Appendix

A. The Solution to TRACTA Satisfies the Matching Consistency Principle

Theorem 1: Given the symmetric local tracklet matching matrix $\mathbf{X} \in \{0,1\}^{N\times N}$ of N tracklets, where $\mathbf{X}(u,v)=1$ if the tracklets u and v are matched and $\mathbf{X}(u,v)=0$ for otherwise. We denote by $\mathbf{A} \in \{0,1\}^{N\times K}$ the assignment matrix from N tracklets to K TIDs, where each element $\mathbf{A}(u,v)$ takes binary values of 1 or 0, which correspond to the assignment and non-assignment of TID v to tracklet u, respectively, and $\mathbf{X}=\mathbf{A}\mathbf{A}^T$. The matching consistency principle on $\mathbf{X}$ (constraints in Eq. 21 and Eq. 22)

$$\forall u,v,\quad \mathbf{X}(u,v)=1 \Longleftrightarrow \mathbf{X}(u,:)=\mathbf{X}(v,:), \tag{21}$$

$$\forall u,v,\quad \mathbf{X}(u,v)=0 \Longleftrightarrow \mathbf{X}(u,:)\mathbf{X}(v,:)^T=0. \tag{22}$$

can be naturally satisfied in TRACTA when the assignment from tracklets to targets is a mapping, *i.e.*, $\forall u, \sum \mathbf{A}(u,:)=1$.

PROOF. The constraint in Eq. 21 is meet that

$$\mathbf{X}(u,v)=1 \Longleftrightarrow \mathbf{A}(u,:)\mathbf{A}(v,:)^T=1.$$

When $\mathbf{A}$ is binary and $\forall u, \sum \mathbf{A}(u,:)=1$, there is only one non-zero element in each row of $\mathbf{A}$. Therefore,

$$\begin{aligned}\mathbf{A}(u,:)\mathbf{A}(v,:)^T=1 &\Longleftrightarrow \mathbf{A}(u,:)=\mathbf{A}(v,:),\\ &\Longleftrightarrow \mathbf{A}(u,:)\mathbf{A}^T=\mathbf{A}(v,:)\mathbf{A}^T,\\ &\Longleftrightarrow \mathbf{X}(u,:)=\mathbf{X}(v,:).\end{aligned}$$

For the constraint in Eq. 22, we have

$$\begin{aligned}\mathbf{X}(u,v)=0 &\Longleftrightarrow \mathbf{A}(u,:)\mathbf{A}(v,:)^T=0,\\ &\Longleftrightarrow tr(\mathbf{A}(u,:)\mathbf{A}(v,:)^T)=0,\\ &\Longleftrightarrow tr(\mathbf{A}(v,:)^T\mathbf{A}(u,:))=0.\end{aligned}$$

Since $\mathbf{A}$ is non-negative and the *trace* of $\mathbf{A}(v,:)^T\mathbf{A}(u,:)$ is zero, the diagonal elements of $\mathbf{A}(v,:)^T\mathbf{A}(u,:)$ are all zero. Moreover, $\Sigma=\mathbf{A}^T\mathbf{A}$ is diagonal and its diagonal elements are non-zero. As a result,

$$tr(\mathbf{A}(v,:)^T\mathbf{A}(u,:))=0 \Longleftrightarrow tr(\mathbf{A}(v,:)^T\mathbf{A}(u,:)\Sigma)=0.$$

As the trace remains invariant under cyclic permutations,

$$\begin{aligned}tr(\mathbf{A}(v,:)^T\mathbf{A}(u,:)\Sigma)=0 &\Longleftrightarrow tr(\mathbf{A}(u,:)\Sigma\mathbf{A}(v,:)^T)=0,\\ &\Longleftrightarrow \mathbf{A}(u,:)\Sigma\mathbf{A}(v,:)^T=0,\\ &\Longleftrightarrow \mathbf{A}(u,:)\mathbf{A}^T\mathbf{A}\mathbf{A}(v,:)^T=0,\\ &\Longleftrightarrow \mathbf{X}(u,:)\mathbf{X}(v,:)^T=0.\end{aligned}$$

This concludes the proof of **Theorem 1**.

References

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

[2] W. Chen, L. Cao, X. Chen, and K. Huang, "An equalized global graph model-based approach for multicamera object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2367–2381, Nov. 2017.

[3] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.

[4] J. Fan, X. Shen, and Y. Wu, "What are we tracking: A unified approach of tracking and recognition," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 549–560, Feb. 2013.

[5] D. Li, X. Wei, X. Hong, and Y. Gong, "Infrared-visible cross-modal person re-identification with an X modality," in *Proc. AAAI Conf. Artif. Intell.*, 2020.

[6] D. Liciotti, M. Contigiani, E. Frontoni, A. Mancini, P. Zingaretti, and V. Placidi, "Shopper analytics: A customer activity recognition system using a distributed RGB-D camera network," in *Proc. Int. Workshop Video Anal. Audience Meas. Retail Digit. Signage*. Cham, Switzerland: Springer, 2014, pp. 146–157.

[7] Z. Tang *et al.*, "CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," 2019, *arXiv:1903.09254*. [Online]. Available: http://arxiv.org/abs/1903.09254

[8] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *Proc. 12th IEEE Int. Workshop Perform. Eval. Tracking Surveill.*, Dec. 2009, pp. 1–6.

[9] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6142–6151.

[10] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," 2017, *arXiv:1701.01909*. [Online]. Available: http://arxiv.org/abs/1701.01909

[11] K. Otsuka and N. Mukawa, "Multiview occlusion analysis for tracking densely populated objects based on 2-D visual angles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2004, p. 1.

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

[13] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, "Modeling inter-camera space–time and appearance relationships for tracking across non-overlapping views," *Comput. Vis. Image Understand.*, vol. 109, no. 2, pp. 146–162, Feb. 2008.

[14] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 267–282, Feb. 2008.

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

[16] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 36–42.

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

[18] B. Zhong, B. Bai, J. Li, Y. Zhang, and Y. Fu, "Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2331–2341, May 2019.

[19] C. Wang, Y. Wang, Y. Wang, C.-T. Wu, and G. Yu, "muSSP: Efficient min-cost flow algorithm for multi-object tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 423–432.

[20] M. Bredereck, X. Jiang, M. Körner, and J. Denzler, "Data association for multi-object tracking-by-detection in multi-camera networks," in *Proc. 6th Int. Conf. Distrib. Smart Cameras (ICDSC)*, 2012, pp. 1–6.

[21] W. Chen, L. Cao, X. Chen, and K. Huang. *Multi-Camera Object Tracking Challenge*. Accessed: Feb. 1, 2014. [Online]. Available: http://mct.idealtest.org

[22] Z. Zhang, J. Wu, X. Zhang, and C. Zhang, "Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on DukeMTMC project," 2017, *arXiv:1712.09531*. [Online]. Available: http://arxiv.org/abs/1712.09531

[23] N. Jiang, S. Bai, Y. Xu, C. Xing, Z. Zhou, and W. Wu, "Online inter-camera trajectory association exploiting person re-identification and camera topology," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1457–1465.

[24] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, "Multi-view people tracking via hierarchical trajectory composition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4256–4265.

[25] Y. Xu, X. Liu, L. Qin, and S.-C. Zhu, "Cross-view people tracking by scene-centered spatio-temporal parsing," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4299–4305.

[26] M. Hofmann, D. Wolf, and G. Rigoll, "Hypergraphs for joint multi-view reconstruction and multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3650–3657.

[27] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn, "Branch-and-price global optimization for multi-view multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1987–1994.

[28] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke, "Tracking a large number of objects from multiple views," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1546–1553.

[29] W. Chen, L. Cao, X. Chen, and K. Huang, "A novel solution for multi-camera object tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2329–2333.

[30] X. Chen and B. Bhanu, "Integrating social grouping for multitarget tracking across cameras in a CRF model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2382–2394, Nov. 2017.

[31] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst, "Sparsity driven people localization with a heterogeneous network of cameras," *J. Math. Imag. Vis.*, vol. 41, nos. 1–2, pp. 39–58, Sep. 2011.

[32] Y. T. Tesfaye, E. Zemene, A. Prati, M. Pelillo, and M. Shah, "Multi-target tracking in multiple non-overlapping cameras using fast-constrained dominant sets," *Int. J. Comput. Vis.*, vol. 127, no. 9, pp. 1303–1320, Sep. 2019.

[33] K. A. S. Kumar, K. R. Ramakrishnan, and G. N. Rathna, "Distributed person of interest tracking in camera networks," in *Proc. 11th Int. Conf. Distrib. Smart Cameras (ICDSC)*, 2017, pp. 131–137.

[34] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[35] K. Li, Y. Kong, and Y. Fu, "Multi-stream deep similarity learning networks for visual tracking," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1–7.

[36] Z. Cui, Y. Cai, W. Zheng, C. Xu, and J. Yang, "Spectral filter tracking," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2479–2489, May 2019.

[37] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1794–1801.

[38] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1449–1458.

[39] A. Li and S. Yan, "Object tracking with only background cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 11, pp. 1911–1919, Nov. 2014.

[40] X. Hong, H. Chang, S. Shan, X. Chen, and W. Gao, "Sigma set: A small second order statistical region descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1802–1809.

[41] X. Hong, H. Chang, S. Shan, B. Zhong, X. Chen, and W. Gao, "Sigma set based implicit online learning for object tracking," *IEEE Signal Process. Lett.*, vol. 17, no. 9, pp. 807–810, Sep. 2010.

[42] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[43] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1614–1627, Aug. 2014.

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

[45] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

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

[47] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2365–2374.

[48] X. Zhao, Y. Fu, and Y. Liu, "Human motion tracking by temporal-spatial local Gaussian process experts," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1141–1151, Apr. 2011.

[49] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.

[50] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, "Learning common and feature-specific patterns: A novel multiple-sparse-representation-based tracker," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2022–2037, Apr. 2018.

[51] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[53] P. Baque, F. Fleuret, and P. Fua, "Deep occlusion reasoning for multi-camera multi-target detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2017, pp. 271–279.

[54] H. Jiang, J. Wang, Y. Gong, N. Rong, Z. Chai, and N. Zheng, "Online multi-target tracking with unified handling of complex scenarios," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3464–3477, Nov. 2015.

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

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

[57] C. Liu, R. Yao, S. H. Rezatofighi, I. Reid, and Q. Shi, "Model-free tracker for multiple objects using joint appearance and motion inference," *IEEE Trans. Image Process.*, vol. 29, pp. 277–288, Jul. 2020.

[58] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, "Principal axis-based correspondence between multiple cameras for people tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 663–671, Apr. 2006.

[59] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[60] Y. Tariku Tesfaye, E. Zemene, A. Prati, M. Pelillo, and M. Shah, "Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets," 2017, *arXiv:1706.06196*. [Online]. Available: http://arxiv.org/abs/1706.06196

[61] Y. Cai and G. Medioni, "Exploring context information for inter-camera multiple target tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 761–768.

[62] Y.-G. Lee, Z. Tang, and J.-N. Hwang, "Online-learning-based human tracking across non-overlapping cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2870–2883, Oct. 2018.

[63] D. Cheng, Y. Gong, J. Wang, Q. Hou, and N. Zheng, "Part-aware trajectories association across non-overlapping uncalibrated cameras," *Neurocomputing*, vol. 230, pp. 30–39, Mar. 2017.

[64] W. Nie *et al.*, "Single/cross-camera multiple-person tracking by graph matching," *Neurocomputing*, vol. 139, pp. 220–232, Sep. 2014.

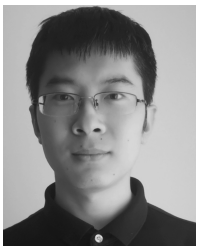
[65] A. B. Poore, "Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking," *Comput. Optim. Appl.*, vol. 3, no. 1, pp. 27–57, Mar. 1994.

[66] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 374–383, Mar. 2014.

[67] H. Zhang, S. Hu, X. Zhang, and L. Luo, "Visual tracking via constrained incremental non-negative matrix factorization," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1350–1353, Sep. 2015.

[68] A. M. Cheriyadat and R. J. Radke, "Non-negative matrix factorization of partial track data for motion segmentation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 865–872.

[69] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.
[70] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 505–519, Mar. 2009.
[71] P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang, "Robust multiple cameras pedestrian detection with multi-view Bayesian network," *Pattern Recognit.*, vol. 48, no. 5, pp. 1760–1772, May 2015.
[72] S. Zhang, J. Wang, Z. Wang, Y. Gong, and Y. Liu, "Multi-target tracking by learning local-to-global trajectory models," *Pattern Recognit.*, vol. 48, no. 2, pp. 580–590, Feb. 2015.
[73] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Dec. 2008.
[74] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*. [Online]. Available: http://arxiv.org/abs/1603.00831
[75] S. Zhou, J. Wang, J. Wang, Y. Gong, and N. Zheng, "Point to set similarity based deep feature learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 6, Jul. 2017, pp. 3741–3750.

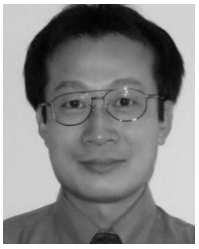


Yuhang He received the B.S. degree in control science and engineering from Xi'an Jiaotong University, Shaanxi, China, in 2016, where he is currently pursuing the Ph.D. degree with the College of Artificial Intelligence (CAI). His research interests include computer vision, pattern recognition, and machine learning, specifically in the areas of video surveillance, image classification, and object recognition.

Xing Wei received the B.E. degree in automation and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Shaanxi, China, in 2013 and 2019, respectively. His research interests include computer vision, pattern recognition, and machine learning, specifically in the areas of visual scene analysis, image retrieval, object recognition, and low-level vision.

Xiaopeng Hong (Member, IEEE) received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, China, in 2010. He was a Docent with the Center for Machine Vision and Signal Analysis, University of Oulu, Finland, where he was a Senior Researcher from 2011 to 2019. He is currently a Distinguished Research Fellow with Xi'an Jiaotong University, China. He was a PI of several projects, such as the National Key Research and Development Program Projects, China, and Infotech Oulu Postdoctoral funding project. He has coauthored over 50 articles in top-tier journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, THE IEEE TRANSACTIONS ON IMAGE PROCESSING, CVPR, ICCV, and AAAI. His current research interests include visual surveillance, continual learning, and micro-expression analysis. His research about micro-expression analysis has been reported by International media, such as MIT Technology Review and Daily Mail. He has been an Area Chair of ACM MM20 and served as a reviewer for a few top-tier journals and conferences. He has been a co-organizer of five international workshops and also served as a Guest Editor of special issues in several journals.

Weiwei Shi received the M.S. degree in applied mathematics and the Ph.D. degree in pattern recognition and intelligent system from Xi'an Jiaotong University, Xi'an, China, in 2012 and 2019, respectively. He is currently an Assistant Professor with the Xi'an University of Technology. His current research interests include multimedia analysis, machine learning, and pattern recognition, specifically in the areas of deep learning, image classification, and image retrieval.



Yihong Gong (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from The University of Tokyo, Japan, in 1987, 1989, and 1992, respectively. In 1992, he joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as an Assistant Professor. From 1996 to 1998, he was a Project Scientist with the Robotics Institute, Carnegie Mellon University, USA. Since 1999, he has been with the Silicon Valley Branch, NEC Labs America, as a Group Leader, a Department Head, and a Branch Manager. In 2012, he joined Xian Jiaotong University, China, as a Distinguished Professor. His research interests include image and video analysis, multimedia database systems, and machine learning.