Multi-Target Tracking in Multiple Non-Overlapping Cameras using Fast-Constrained Dominant Sets

IJCV manuscript No.
(will be inserted by the editor)

Multi-Target Tracking in Multiple Non-Overlapping Cameras using Fast-Constrained Dominant Sets

Yonatan Tariku Tesfaye · Eyasu Zemene · Andrea Prati · Marcello Pelillo · Mubarak Shah

Received: date / Accepted: date

Abstract In this paper, a unified three-layer hierarchical approach for solving tracking problem in a multiple non-overlapping cameras setting is proposed. Given a video and a set of detections (obtained by any person detector), we first solve *within-camera tracking* employing the first two layers of our framework and then, in the third layer, we solve *across-camera tracking* by associating tracks of the same person in all cameras simultaneously. To best serve our purpose, we propose Fast-Constrained Dominant Set Clustering (FCDSC), a novel method which is several orders of magnitude faster (close to real time) than existing methods. FCDSC is a parameterized family of quadratic programs that generalizes the standard quadratic optimization problem.

In our method, we first build a graph where nodes of the graph represent short-tracklets, tracklets and tracks in the first, second and third layer of the framework, respectively. The edge weights reflect the similarity between nodes. FCDSC takes as input a constrained set, a subset of nodes from the graph which need to be included in the extracted cluster. Given a constrained set, FCDSC generates compact clusters by selecting nodes from the graph which are highly similar to each other and with elements in the constrained set.

We have tested this approach on a very large and challenging dataset (namely, MOTchallenge DukeMTMC) and show that the proposed framework outperforms the state-of-the-art approaches. Even though the main focus of this paper is on multi-target tracking in non-overlapping cameras, the proposed approach can also be applied to solve *video-based person re-identification* problem. We show that when the re-identification problem is formulated as a clustering problem, FCDSC can be used in conjunction with state-of-the-art video-based re-identification algorithms, to increase their already good performances. Our experiments demonstrate the general applicability of the proposed framework for multi-target multi-camera tracking and person re-identification tasks.

Keywords Multi-target multi-camera tracking · Constrained Dominant Sets · Standard Quadratic optimization

Y. T. Tesfaye
Center for Research in Computer Vision (CRCV) at UCF and IUAV University of Venice, Italy
E-mail: yonitare@gmail.com
yonatantariku@knights.ucf.edu

E. Zemene
Center for Research in Computer Vision (CRCV) at UCF and Ca' Foscari University of Venice, Italy
E-mail: eyasu201011@gmail.com

M. Pelillo
DAIS / ECLT
Ca' Foscari University of Venice, Italy
E-mail: pelillo@unive.it

A. Prati
Department of Engineering and Architecture
University of Parma, Italy
E-mail: andrea.prati@unipr.it

M. Shah
Center for Research in Computer Vision (CRCV)
University of Central Florida, USA
E-mail: shah@eecs.ucf.edu

1 Introduction

As the need for visual surveillance grows, a large number of cameras are being deployed to cover large and wide areas like airports, shopping malls, city blocks, etc. Since the fields of views of single cameras are limited,

in most wide-area surveillance scenarios, multiple cameras are required to cover larger areas. Using multiple cameras with overlapping fields of view is costly from both economical and computational aspects. Therefore, camera networks with non-overlapping fields of view are preferred and widely adopted in real-world applications.

In this work, our goal is to track multiple targets and maintain their identities as they move from one camera to another camera with non-overlapping fields of views. In this context, two problems need to be solved, that is, within-camera data association (or tracking) and across-cameras data association by employing the tracks obtained from within-camera tracking. Although there has been significant progress in both problems separately, tracking multiple targets jointly in both within and across non-overlapping cameras remains a poorly explored topic.

In this paper, we first determine tracks within each camera (Figure 1(a)), by solving data association, and later we associate tracks of the same target in different cameras in a unified approach (Figure 1(b)), hence solving the across-camera tracking. Since appearance and motion cues of a target tend to be consistent in a short temporal window in a single camera tracking, solving tracking problems in a hierarchical manner is common: tracklets are generated within short temporal windows first and later they are linked or merged to form full tracks (or trajectories).

We cast the tracking problem as finding a cluster of nodes in a graph. Though graph-based approaches are efficient in solving tracking problems, they have limitations with the size of the graph. In this work, we propose a novel fast-constrained dominant sets clustering (FCDSC) technique, a parametrized version of standard quadratic optimization, to solve both within- and across-camera tracking tasks. Typical graph-based methods use the whole graph to solve the clustering problem. In our approach, instead of employing the whole graph, we consider a sub-graph that is much smaller than the original graph. This allows our approach to handle arbitrarily-large graphs and also to be an order of magnitude faster.

Given a constrained set and a graph, FCDSC generates a compact cluster, that is, it selects a subset of nodes from the given graph, which form compact and coherent cluster. Since the nodes in graphs in the first, second and third layers, respectively, represent short-tracklets, tracklets and tracks, corresponding clusters essentially solve the data association among them and define tracklets, tracks and across camera correspondences. The proposed within-camera tracker can robustly handle long-term occlusions, does not change the scale of original problem, as it does not remove nodes from the graph during the extraction of compact clusters, and is several orders of magnitude faster (close to real time) than existing methods.

The proposed across-camera tracking method has several other advantages. More specifically, FCDSC not only considers the affinity (relationship) between tracks, observed in different cameras, but also takes into account the affinity among tracks from the same camera. As a consequence, the proposed approach not only accurately associates tracks from different cameras, but also makes it possible to link multiple short broken tracks obtained during within-camera tracking, which may belong to a single target track. For instance, in Figure 1(a) track T_1^3 (third track from camera 1) and T_1^4 (fourth track from camera 1) are tracks of the same person, which were mistakenly broken from a single track. However, during the third layer, as they are highly similar to tracks in camera 2 (T_2^3) and camera 3 (T_3^3), they form a cluster, as shown in Figure 1(b). Such across-camera formulation is able to associate these broken tracks with the rest of tracks from different cameras, represented with the green cluster in Figure 1(b).

The contributions of this paper are summarized as follows:

- We propose a novel fast-constrained dominant set clustering approach, which is highly scalable and 2000x faster than previous Constraint Dominant Set Clustering (CDSC) method.
- We formulate multi-target tracking in multiple non-overlapping cameras as finding compact cluster from a graph and simultaneously solving within- and across-camera tracking.
- We propose a refinement step which is a principled way to decide between ambiguous tracks and which overall improves the results.
- Experiments are performed on MOTchallenge DukeMTMC and MARS datasets, which show improved effectiveness of our method with respect to the state of the art.

The rest of the paper is organized as follows. In Section 2, we review relevant previous works. Following that, a brief background on Constrained Dominant Set clustering (CDSC) is present in section 3. The proposed approach for within- and across-cameras tracking modules is summarized in section 4. In particular, in subsection 4.1, we present the proposed fast-constrained dominant set clustering method, while sections 5 and 6 provide details of within- and across-cameras tracking. Next, section 7 discusses our track refinement step. Experimental results are presented in Section 8. Finally, section 9 concludes the paper.

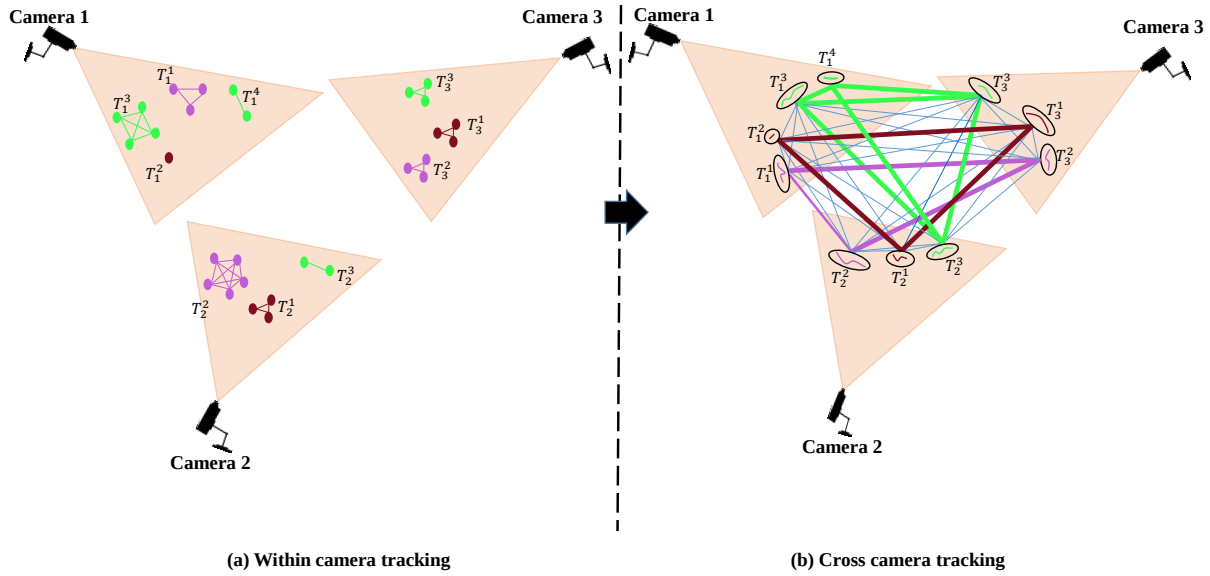

Fig. 1: An overview of the proposed framework. A field of View (FOV) of each camera is shown by a triangle and tracklets and tracks of each camera are shown within each triangle. (a) First, tracks are determined within each camera, then (b) tracks of the same person from different non-overlapping cameras are associated, solving the across-camera tracking. Nodes in (a) represent tracklets and nodes in (b) represent tracks. T_j^i is a set of tracklets that form a cluster, resulting in the i^{th} track of camera j. Nodes in (b) represent tracks form different cameras. Each cluster shown in color represents the tracks of the same person in non-overlapping cameras. Similar color cluster represent the same person. (Best viewed in color)

2 Related work

Visual tracking is a very active area of research and has rich literature. It is very difficult to cover all different approaches. For the sake of this, we only cover a few relevant works in the context of this paper and refer readers to some excellent surveys on tracking [20, 36, 43]. We divide our review into the three following parts.

Single camera tracking: Single camera tracking associates target detections across frames in a video sequence in order to generate the target motion trajectory over time. Zamir *et al.* [47] formulate tracking problem as generalized maximum clique problem (GMCP), where the relationships between all detections in each frame of a temporal window are considered. In [47], a cost is assigned to each clique, which is a complete graph such that one node is selected from each frame, results in a compact cluster corresponding to a track. Then, a clique which maximizes their score function is selected. Nonetheless, the approach is prone to local optima as it uses greedy local neighbourhood search. Deghan *et al.* [6] cast tracking as a generalized maximum multi clique problem (GMMCP) and follow a joint optimization for all the tracks simultaneously. In order to handle outliers and weak-detections associations, they introduce dummy nodes. However, this solution is computationally expensive. In addition, the hard constraint in their optimization makes the approach impractical for large graphs. Authors in [19] simultaneously optimized detection and tracking by coupling them into a Quadratic Boolean Problem. In [1], they apply a maximum-weight independent set algorithm to hierarchically merge small tracklets into long tracks. In [40] they formalize tracking as solving min-cost network flow problem by first grouping detections into tracklets and then combining those into tracks. Authors of [13] revisit the JPDA (Joint Probabilistic Data Association) formulation and address the issue of its complexity by leveraging the latest developments in finding the m-best solutions of an integer linear program. The authors in [46] propose an identity-aware multi-object tracker based on the solution path algorithm. The tracker is formulated as a quadratic optimization problem with l_0 norm constraints, which they solve with the solution path algorithm. The authors in [26] formulate ball tracking problem in sport events and related physical constraints in terms of mixed integer programming. However, most of the above approaches suffer computationally when the number of target increases, while our approach can handle arbitrarily-large graphs, as it operates on selected sub-graphs of the bigger graph.

Multi camera tracking: Across-camera tracking is a challenging problem due to the illumination and pose changes across cameras, or track discontinuities due to the blind areas or miss detections. Existing across-camera tracking methods try to deal with the above problems using appearance cues. The variation in the appearance due to illumination changes has been dealt with using different techniques, such as Brightness Transfer Functions (BTFs) [15]. Authors of [11] used an incremental learning method to model the color variations, and [31] proposed a Cumulative BTF, which is a better use of the available color information from a very sparse training set. Performance comparison of different variations of BTFs can be found in [7]. Authors in [39] tried to achieve color consistency using colorimetric principles, where the image analysis system is modelled as an observer and camera-specific transformations are determined, so that images of the same target appear similar to this observer.

Obviously, learning BTFs or color correction models require large amount of training data and they may not be robust against drastic illumination changes across different cameras. Therefore, recent approaches have combined them with spatio-temporal cues, which improve multi-target tracking performance [2, 3, 4, 10, 18, 50]. Chen *et al.* [4] utilized human part configurations for every target track from different cameras to describe the across-camera spatio-temporal constraints for across-camera track association, which is formulated as a multi-class classification problem via Markov Random Fields (MRFs). In [2] spatio-temporal context is used for collecting samples for discriminative appearance learning. Authors in [3] learn across-camera transfer models including both spatio-temporal and appearance cues.

However, using only low-level information (appearance and spatio-temporal information) is unreliable for tracking across non-overlapping cameras. Therefore, in this work, for associating tracks across different cameras, besides using their pairwise similarity computed using appearance and spatio-temporal cues, we employ their relative similarity, enforcing that their corresponding FCDSC clusters should also be similar. Also, most of the approaches mentioned above assume within-camera tracking results for all cameras are given. Conversely, the work proposed in this paper addresses a more realistic problem by solving both within- and across-camera tracking in one joint framework.

Recently, the problem of target tracking across multiple non-overlapping cameras has been also tackled in [34], by extending their previous single camera tracking method [33], where they formulate the tracking task as a graph partitioning problem. However, their approach gets impractical as their graph gets larger, while we propose a highly-scalable approach, which can handle arbitrarily-large graphs. In [27], authors impose global consistency of trajectories by using behavioral patterns to guide the tracking algorithm. They showed that when their approach is used together with the existing state-of-the-art tracking algorithms, it further improves their performance. However, in this approach the initial trajectories are assumed to be given which is a big limitation. In [32], authors showed that learning high-quality appearance features lead to good clustering solutions and proposed adaptive weighted triplet loss to learn better feature embeddings. Authors in [44] used track-hypothesis trees to solve tracking in multiple cameras. Within-camera tracking is performed simultaneously with the tree formation by manipulating a status of each track hypothesis. However, their approach suffers in handling crowded scenes.

Person Re-Identification: Another recent popular research topic, video-based person re-identification (ReID) [5, 8, 21, 25, 28, 41, 42, 45, 52], is closely related to across-camera multi-target tracking. Both problems aim to match tracks of the same persons across non-overlapping cameras. However, across-camera tracking aims at 1-1 correspondence association between tracks of different cameras. Moreover, person ReID approaches mainly focus on building a strong appearance model to match the same person observed in different views or learning a distance metric [17, 22] to maximize the differences between different people. To this end, ReID approaches have made impressive advances. However, most video-based ReID approaches exploit only pairwise affinities between the probes and gallery to get final sorting. By employing FCDSC in conjunction with state-of-the-art video-based ReID algorithms, and by formulating the problem as a constrained quadratic optimization problem, we show that performance of ReID methods can be further increased.

3 Background on Constrained Dominant Set Clustering

In this section, we briefly introduce the basic definitions and formulations of constrained dominant set clustering framework.

As introduced in [48], constrained dominant set clustering, a constrained quadratic optimization program, is an efficient approach that has been originally applied to interactive image segmentation. The approach generalizes the dominant set clustering framework [29], which is a well-known generalization of the maximal

clique problem to edge weighted graphs. Given an edge-weighted graph $G(V, E, w)$ and a constraint set $\mathcal{Q} \subseteq V$, where V, E and w, respectively, denote the set of nodes (of cardinality n), edges and edge weights. The objective is to find for sub-graphs that contain all or some of elements of the constraint set, which form a compact cluster.

Consider a graph, G, with n vertices (set V), and its weighted adjacency matrix $\mathtt{A}$. Given a parameter $\alpha > 0$, let us define the following parametrized quadratic program:

$$\begin{aligned} &\text{maximize} \quad f^{\alpha}_{\mathcal{Q}}(\mathbf{x}) = \mathbf{x}^{\top}(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x} \\ &\text{subject to } \mathbf{x} \in \Delta \end{aligned} \tag{1}$$

where $\Delta = \{\mathbf{x} \in \mathbb{R}^n : \sum_i x_i = 1, \text{ and } x_i \geq 0 \text{ for all } i = 1 \ldots n\}$. From now onwards we will be calling $\mathbf{x}$ state, distribution (since elements sum to one, it can be called distribution) and membership score vector interchangeably. $I_{\mathcal{Q}}$ is the $n \times n$ diagonal matrix whose diagonal elements are set to 1 in correspondence to the vertices contained in $V \setminus \mathcal{Q}$ (a set V without the elements in $\mathcal{Q}$) and to zero otherwise.

Let $\mathcal{Q} \subseteq V$, with $\mathcal{Q} \neq \emptyset$ and let $\alpha > \lambda_{\max}(\mathtt{A}_{V\setminus\mathcal{Q}})$, where $\lambda_{\max}(\mathtt{A}_{V\setminus\mathcal{Q}})$ is the largest eigenvalue of the principal submatrix of $\mathtt{A}$ indexed by the elements of $V \setminus \mathcal{Q}$. If $\mathbf{x}$ is a local maximizer of $f^{\alpha}_{\mathcal{Q}}$ in Δ, then $\sigma(\mathbf{x}) \cap \mathcal{Q} \neq \emptyset$, where, $\sigma(\mathbf{x}) = \{i \in V \ : \ x_i > 0\}$ [48].

The above result provides us with a simple technique to determine clusters containing user-specified query vertices, $\mathcal{Q}$. Indeed, if $\mathcal{Q}$ is a vertex selected by the user, by setting

$$\alpha > \lambda_{\max}(\mathtt{A}_{V\setminus\mathcal{Q}}) \tag{2}$$

we are guaranteed that all local solutions of (1) will have a support that necessarily contains elements of $\mathcal{Q}$.

Standard quadratic program (StQP) solvers: The above Standard Quadratic Program (1) can be solved using dynamics from evolutionary game theory. In [48], replicator dynamics, a well-known family of algorithms from evolutionary game theory inspired by Darwinian selection processes [37], is employed to solve standard quadratic program. Despite their effectiveness in finding good solutions in a variety of applications, however, replicator dynamics suffer from being computationally expensive, as they require a number of operations per step which grows quadratically with the dimensionality of the problem being solved, $\mathcal{O}(n^2)$, which makes it inefficient for large scale problems. Efficient out-of-sample [30], an extension of dominant set methods, is used to reduce the computational cost by sampling nodes of the graph using a given sampling rate, which affects the efficiency of the framework. Liu *et al.* [23] proposed an iterative clustering algorithm, which operates in two steps: Shrink and Expansion. These steps are used to reduce the run time of replicator dynamics, however they require the whole graph, which makes it slow. The approach has many limitations, such as its preference of sparse graphs with several small clusters and the results are sensitive to several additional parameters.

All the above formulations, with their limitations, try to minimize the computational complexity of StQP using standard game dynamics, whose complexity is $\mathcal{O}(n^2)$ for each iteration. Rota Bulò *et al.* [35] proposed a new class of evolutionary game dynamics, called Infection and Immunization Dynamics (InfImDyn). It simulates the infection and immunization process. The process which finds a distribution able to infect the population and the process which finds the immune state are all linear processes. Therefore, InfImDyn solves the problem in linear time $\mathcal{O}(n)$. However, it needs the whole affinity matrix to extract a cluster.

In our approach, we further reduce the computational time by running InfImDyn on a very small sub-graph selected out from the original graph. We propose a principled technique to select sub-graph which contains all possible solutions. We also show that the solution from the sub-graph is a valid solution in the larger graph, which makes the dynamics even more faster, $\mathcal{O}(r)$, where $r << n$, by significantly reducing the search space. Note that, the proposed approach can be used with all other solvers discussed above to improve their efficiency as it helps reducing the search space.

4 Overall Approach

In our formulation, in the first layer each node in a graph represents a short-tracklet along a temporal window (typically 15 frames) (Figure 2(a)). We apply fast-constrained dominant set clustering to determine clusters in this graph, which correspond to *tracklets*. Likewise, each node in a graph in the second layer represents a tracklet (Figure 2(b)), obtained from the first layer, and FCDSC is applied here to determine clusters, which correspond to *tracks* (Figure 2(c)). Finally, in the third layer, nodes in a graph correspond to tracks from different non-overlapping cameras, obtained from the second layer, and FCDSC is applied to determine clusters, which relate tracks of the same person across non-overlapping cameras.

In this section, first we present our proposed fast-constrained dominant set clustering approach. This is followed by formulation of within- and across-camera tracking.

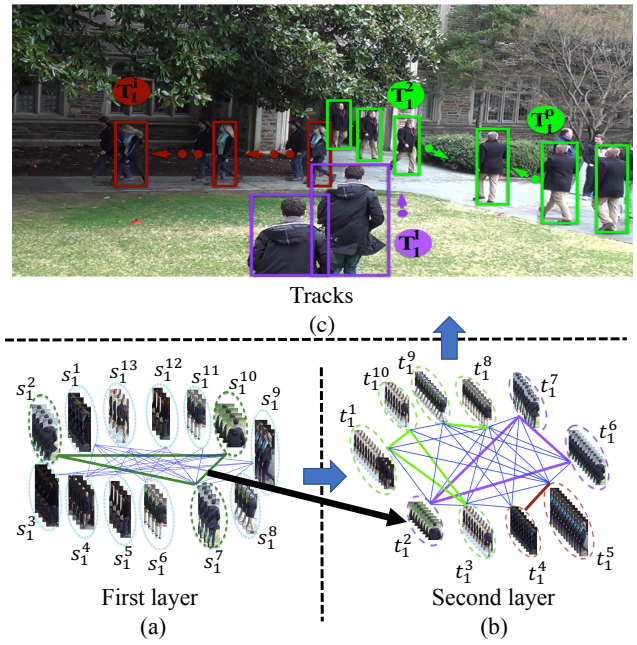

Fig. 2: The figure shows within-camera tracking where short-tracklets (s) from different segments are used as input to our first layer of tracking. The resulting tracklets (t) from the first layer are inputs to the second layer, which determine tracks (T) for each person. The three dark green short-tracklets (s_1^2, s_1^{10}, s_1^7), shown by dotted ellipse in the first layer, form a cluster resulting in a tracklet (t_1^2) in the second layer, as shown with the black arrow. In the second layer, each cluster, shown in purple, green and dark red colors, form tracks of different targets, as can be seen on the top row, tracklets and tracks with the same color indicate same target. The two green clusters (with two tracklets and three tracklets) represent tracks of the person going in and out of the building (tracks T_1^p and T_1^2 respectively)

4.1 Fast-Constrained Dominant Set Clustering

In this paper, we propose an algorithm that reduces the search space using the Karush-Kuhn-Tucker (KKT) condition of the constrained quadratic optimization, effectively enforcing the user constraints. In the constrained optimization framework [48], the algorithm computes the eigenvalue of the submatrix for every extraction of the compact cluster, which contains the user constraint set. Computing eigenvalues for large graphs is computationally intensive, which makes the whole algorithm inefficient. In our approach, instead of operating over the whole graph, we localize it on the submatrix, selected using the dominant distribution, that is much smaller than the original one. To alleviate the issue with the eigenvalues, we utilize the properties of eigenvalues; a good approximation for the parameter α is to use the maximum degree of the graph, which, of course, is larger than the eigenvalue of corresponding matrix. The computational complexity, apart from eigenvalue computation, is reduced to $\mathcal{O}(r)$ where r is the size of the sub-matrix which is employed, which is much smaller than the dimension of the original affinity matrix.

Let us summarize the KKT conditions for quadratic program reported in eq. (1). By adding Lagrangian multipliers, n non-negative constants $\mu_1, ..., \mu_n$ and a real number λ, its Lagrangian function is defined as follows:

$$\mathcal{L}(\mathbf{x}, \mu, \lambda) = f_{\mathcal{Q}}^{\alpha}(\mathbf{x}) + \lambda \left(1 - \sum_{i=1}^{n} x_i\right) + \sum_{i=1}^{n} \mu_i x_i.$$

For a distribution $\mathbf{x}$, $\mathbf{x} \in \Delta$, to be a KKT-point, that is, in order to satisfy the first-order necessary conditions for local optimality [24], it should satisfy the following two conditions: the derivative of the Lagrangian $\mathcal{L}$ should be zero

$$2 * [(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}]_i - \lambda + \mu_i = 0,$$

for all $i = 1 \ldots n$, and

$$\sum_{i=1}^{n} x_i \mu_i = 0 \ .$$

Since both the x_i and the μ_i are nonnegative, the latter condition is equivalent to saying that $i \in \sigma(\mathbf{x})$ which implies that $\mu_i = 0$, from which we obtain:

$$[(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}]_i \begin{cases} = \lambda/2, & \text{if } i \in \sigma(\mathbf{x}) \\ \leq \lambda/2, & \text{if } i \notin \sigma(\mathbf{x}) \end{cases} \tag{3}$$

We now need to define a
dominant distribution."

Definition 1 *A distribution* $\mathbf{y} \in \Delta$ *is said to be a* ***dominant distribution*** *for* $\mathbf{x} \in \Delta$ *if*

$$\left\{ \sum_{i,j=1}^{n} x_i y_j a_{ij} - \alpha \sum_{i=1}^{n} x_i y_i \right\} > \left\{ \sum_{i,j=1}^{n} x_i x_j a_{ij} - \alpha \sum_{i=1}^{n} x_i^2 \right\}, \tag{4}$$

where a_{ij} is the similarity between node i and j.

Let the "support" be $\sigma(\mathbf{x}) = \{i \in V \ : \ x_i > 0\}$ and e_i the i^{th} unit vector (a zero vector whose i^{th} element is one).

Proposition 1 *Given an affinity* $\mathtt{A}$ *and a distribution* $\mathbf{x} \in \Delta$, *if* $(\mathtt{A}\mathbf{x})_i > \mathbf{x}'\mathtt{A}\mathbf{x} - \alpha \mathbf{x}'_{\mathcal{Q}} \mathbf{x}_{\mathcal{Q}}$, *for* $i \notin \sigma(\mathbf{x})$,

1. $\mathbf{x}$ *is not the maximizer of the parametrized quadratic program of (1)*
2. e_i *is a* ***dominant distribution*** *for* $\mathbf{x}$

Proof . To show that the first condition holds, let us assume $\mathbf{x}$ is a KKT point

$$\mathbf{x}^\top(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x} = \sum_{i=1}^{n} x_i[(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x}]_i$$

Since $\mathbf{x}$ is a KKT point

$$\mathbf{x}^\top(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x} = \sum_{i=1}^{n} x_i * \lambda/2 = \lambda/2.$$

From the second condition, we have:

$$[(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x}]_i \leq \lambda/2 = \mathbf{x}^\top(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x}$$

Since $i \notin \sigma(\mathbf{x})$

$$(\mathtt{A}\mathbf{x})_i \leq \mathbf{x}^\top(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x}.$$

Which concludes the proof showing that the inequality does not hold.

As for the second condition, if e_i is a **dominant distribution** for $\mathbf{x}$, it should satisfy the inequality

$$\left\{e_i^\top(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x}\right\} > \left\{\mathbf{x}^\top(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x}\right\}.$$

Since $i \notin \sigma(\mathbf{x})$ following should hold:

$$(\mathtt{A}\mathbf{x})_i > \left\{\mathbf{x}^\top(\mathtt{A} - \alpha I_\mathcal{Q})\mathbf{x}\right\}.$$

Which concludes the proof.

The proposition provides us with an easy-to-compute dominant distribution.

We summarize the details of our proposed algorithm in Algorithm (1). Given a subset of vertices $\mathcal{Q} \subseteq V$, the face of Δ corresponding to $\mathcal{Q}$ is given by: $\Delta_\mathcal{Q} = \{\mathbf{x} \in \Delta : \sigma(\mathbf{x}) \subseteq \mathcal{Q}\}$. Here the function $\mathcal{G}(\mathtt{A}, \mathcal{Q})$ returns the local maximizer of program (1). $\mathcal{S}(\mathtt{A}, \mathbf{x})$ returns a dominant distribution, $\mathbf{x}_d$, for distribution, $\mathbf{x}$. $\mathbf{x}_d$ is a distribution whose support contains all possible indices $i \in V$, which satisfy the second condition from the proposition 1. Then, $\mathcal{H}$ is computed by taking the union of the support of $\mathbf{x}_d$ with the set of indices which are considered as constraint set $\mathcal{Q}$, which gives us the final sub-graph nodes on which we run the dynamics.

The selected dominant distribution always increases the value of the objective function. Moreover, the objective function is bounded, which guarantees the convergence of the algorithm.

Algorithm 1: FCDSC

INPUT: Affinity $\mathtt{A}$, Constraint set $\mathcal{Q}$;
Initialize $\mathbf{x}$ to the barycenter of $\Delta_\mathcal{Q}$;
$\mathbf{x}_d \leftarrow \mathbf{x}$, initialize *dominant distribution* ;
while *true* **do**
  $\mathbf{x}_d \leftarrow \mathcal{S}(\mathtt{A}, \mathbf{x})$, Find dominant distribution for $\mathbf{x}$;
  if $\sigma(\mathbf{x}_d) = \emptyset$ break ;
  $\mathcal{H} \leftarrow \sigma(\mathbf{x}_d) \cup \mathcal{Q}$, subgraph nodes;
  $\mathtt{B} \leftarrow \mathtt{A}_\mathcal{H}$, extract sub matrix from $\mathtt{A}$ corresponding to indices in $\mathcal{H}$;
  $\mathbf{x}_l \leftarrow \mathcal{G}(\mathtt{B}, \mathcal{Q})$;
  $\mathbf{x} \leftarrow \mathbf{x}$*0;
  $\mathbf{x}(\mathcal{H}) \leftarrow \mathbf{x}_l$;
end
OUTPUT: $\{\mathbf{x}\}$

5 Within-Camera Tracking

Figure 2 shows the proposed within-camera tracking framework. First, we divide a video into multiple short segments, each segment contains f frames (typically 15), and generate short-tracklets, where human detection bounding boxes in two consecutive frames with large (typically 70%) overlap, are connected [6]. Then, short-tracklets from $\mathcal{S}$ (typically 10) different non-overlapping segments are used as input to our first layer of tracking. Here the nodes are short-tracklets (Figure 2(a)). Resulting tracklets from the first layer are used as an input to the second layer, that is, a tracklet from the first layer is now represented by a node in the second layer (Figure 2(b)). In the second layer, tracklets of the same person from different segments are associated forming tracks of a person within a camera.

5.1 Formulation Using Fast-Constrained Dominant Sets

We build an input graph, $G(V, E, w)$, where nodes represent short-tracklet (s_i^j, that is, j^{th} short-tracklet of camera i) in the case of first layer (Figure 2(a)) and tracklet (t_k^l, that is, l^{th} tracklet of camera k), in the second layer (Figure 2(b)). The corresponding affinity matrix $\mathtt{A} = \{a_{i,j}\}$, where $a_{i,j} = w(i, j)$ is built. The weight $w(i, j)$ is assigned to each edge, by considering both motion and appearance similarity between the two nodes. Fine-tuned CNN features are used to model the appearance of a node. These features are extracted from the last fully-connected layer of Imagenet pre-trained 50-layers Residual Network (ResNet 50) [14] fine-tuned using the *Trainval* sequence of DukeMTMC dataset. Similar to [47], we employ a global constant velocity model to compute motion similarity between two nodes.

Determining clusters: In our formulation, a cluster from graph G represents tracklet (track) in the first

(second) layer. Using short-tracklets/tracklets as a constraint set (in eq. 1), we enumerate all clusters, employing the proposed approach, by utilizing intrinsic properties of fast-constrained dominant sets. Note that we do not remove the nodes of found clusters from the graph, this keeps the scale of our problem the same (number of nodes in a graph), which guarantees that all the local solutions found are the local solutions of the (original) graph. After the extraction of each cluster, the constraint set is changed in such a way as to make the extracted cluster less favored by the dynamics. This is achieved by enforcing that our algorithm will not be able to select sets of nodes from the previous solutions. The within-camera tracking starts with all nodes as constraint set. Let us say Γ^i is the i^{th} extracted cluster, Γ^1 is then the first extracted cluster which contains a subset of elements from the whole set. After our first cluster extraction, we change the constraint set to a set $V \backslash \Gamma^1$, hence rendering its associated nodes unstable. The procedure iterates, updating the constraint set at the i^{th} extraction as $V \backslash \bigcup_{l=1}^{i} \Gamma^l$, until the constraint set becomes empty. Since we are not removing the nodes of the graph (after each extraction of a cluster), we may end up with the solution that assigns a node to more than one cluster, which results in overlapping clusters.

To find the final solution, we use the notion of centrality of fast-constrained dominant sets. The true class of a node j, which is assigned to $\mathtt{K} > 1$ cluster, $\psi = \{\Gamma^1 \dots \Gamma^{\mathtt{K}}\}$, is computed as:

$$\arg\max_{\Gamma^i \in \psi} \left(|\Gamma^i| * \delta_j^i\right),$$

where the cardinality $|\Gamma^i|$ is the number of nodes that forms the i^{th} cluster and δ_j^i is the membership score of node j obtained when assigned to cluster Γ^i. The normalization using the cardinality is important to avoid any unnatural bias to a smaller set.

Algorithm (2), setting the number of cameras under consideration ($\mathcal{I}$) equal to 1 and $\mathcal{Q}$ as short-tracklets (tracklets) set in the first(second) layer, is used to determine a cluster which corresponds to tracklet (track) in the first (second) layer.

6 Across-Camera Tracking

6.1 Graph Representation of Tracks and the Payoff Function

Given tracks T_i^j of different cameras from the previous step, we build the graph $G'(V', E', w')$, where nodes represent tracks and their corresponding affinity matrix $\mathtt{A}$ depicts the similarity between tracks.

Assuming we have $\mathcal{I}$ cameras and $\mathtt{A}^{i\times j}$ represents the similarity among tracks of camera i and j, the final track based affinity $\mathtt{A}$, is built as

$$\mathtt{A} = \begin{pmatrix} \mathtt{A}^{1\times 1} & .. & \mathtt{A}^{1\times j} & .. & \mathtt{A}^{1\times \mathcal{I}} \\ . & . & . & . & . \\ \mathtt{A}^{i\times 1} & .. & \mathtt{A}^{i\times j} & . & \mathtt{A}^{i\times \mathcal{I}} \\ . & & . & . & . \\ \mathtt{A}^{\mathcal{I}\times 1} & .. & \mathtt{A}^{\mathcal{I}\times j} & .. & \mathtt{A}^{\mathcal{I}\times \mathcal{I}} \end{pmatrix}.$$

Figure 3 shows an example of a graph for across-camera tracking among three cameras. Black and orange edges, respectively, represent within- and across-camera relations of the tracks. From the affinity $\mathtt{A}$, $\mathtt{A}^{i\times j}$ represents the black edges of camera i if $i = j$, which otherwise represents the across-camera relations using the orange edges. The colors of the nodes depict the track ID: nodes with similar color represent tracks of the same person. Due to several reasons such as long occlusions, severe pose change of a person, reappearance and others, a person may have more than one track (a *broken track*) within a camera. The green nodes of camera 1 (the second and the p^{th} tracks) typify two *broken tracks* of the same person, due to reappearance as shown in Figure 2. The proposed unified approach, as discussed in the next section, is able to deal with such cases.

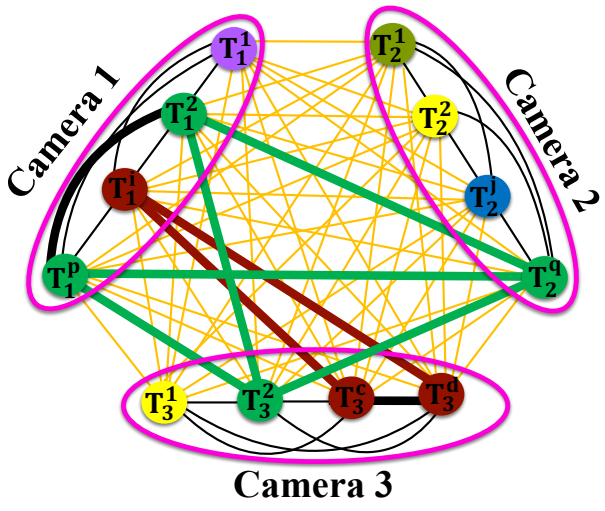

Fig. 3: Exemplar graph of tracks from three cameras. T_j^i represents the i^{th} track of camera j. Black and colored edges, respectively, represent within- and across-camera relations of tracks. Colors of the nodes depict track IDs, nodes with similar color representing tracks of the same person, and the thick lines show both within- and across-camera association.

6.2 Across-camera Track Association

In this section, we discuss how we simultaneously solve within- and across-camera tracking. Our framework is naturally able to deal with the errors listed above. In our exemplar graph (Figure 3), targets represented by a green and red nodes in camera 1 and 2, respectively, has two tracks which are difficult to merge during within-camera tracking; however, they belong to a cluster with track(s) in camera 2 and 3 in the case of the green target and camera 1 for the red target, since they are highly similar. The algorithm applied to a such across-camera graph is able to cluster all the correct tracks. This helps us linking broken tracks of the same person occurring during within-camera track generation stage.

Using the graph with nodes of tracks from a camera as a constraint set, data association for both within- and across-camera are performed simultaneously. Let us assume, in our exemplar graph (Figure 3), that our constraint set $\mathcal{Q}$ contains nodes corresponding to tracks of camera 1, $\mathcal{Q} = \{\ T_1^1, T_1^2, T_1^i, T_1^p\ \}$. $I_{\mathcal{Q}}$ is then a $n \times n$, where n is number of tracks in all cameras, diagonal matrix, whose diagonal elements are set to 1 in correspondence to the vertices contained in all cameras, except camera 1 which takes the value zero. That is, the submatrix $I_{\mathcal{Q}}$, that corresponds to $A^{1\times 1}$, will be a zero matrix of size equal to number of tracks of the corresponding camera. Setting $\mathcal{Q}$ as above, we have guaranteed that the maximizer of program in eq. (1) contains some elements from set $\mathcal{Q}$: i.e., $\mathcal{C}_1^1 = \{T_1^2, T_1^p, T_2^q, T_3^2\}$ forms a cluster which contains set $\{T_1^2, T_1^p\} \in \mathcal{Q}$. This is shown in Figure 3, using the thick green edges (which illustrate across-camera track association) and the thick black edge (which typifies the within-camera track association). The second set $\mathcal{C}_1^2$ contains tracks shown with the dark red color, which illustrates the case where within- and across-camera tracks are in one cluster. Lastly, the $\mathcal{C}_1^3 = T_1^1$ represents a track of a person that appears only in camera 1. As a general case, C_j^i represents the i^{th} track set using tracks in camera j as a constraint set and C_j is the set that contains track sets generated using camera j as a constraint set, e.g. $C_1 = \{C_1^1, C_1^2, C_1^3\}$. We iteratively process all the cameras and then apply track refinement step, described in Section 7.

Though Algorithm (2) is applicable to within-camera tracking also, here we show the specific case for across-camera track association. Let $\mathcal{T}$ represents the set of tracks from all the cameras we have and C is the set which contains sets of tracks, as C_p^i, generated using our algorithm. T_p^ϑ typifies the ϑ^{th} track from camera p and T_p contains all the tracks in camera p. The function $\mathcal{F}(\mathcal{Q}, \mathtt{A})$ takes as an input a constraint set $\mathcal{Q}$ and the affinity $\mathtt{A}$, and provides as output all the m local solutions $\mathcal{X}^{n\times m}$ of program (1) that contain element(s) from the constraint set. This can be accomplished by iteratively finding a local maximizer of equation (program) (1) in Δ, e.g. using game dynamics, and then changing the constraint set $\mathcal{Q}$, until all members of the constraint set have been clustered.

Algorithm 2: Track Association

INPUT: Affinity $\mathtt{A}$, Sets of tracks $\mathcal{T}$ from $\mathcal{I}$ cameras;
$C \leftarrow \emptyset$ Initialize the set with empty-set ;
Initialize $\mathbf{x}$ to the barycenter and i and p to 1;
while $p \leq \mathcal{I}$ **do**
  $\mathcal{Q} \leftarrow T_p$, define constraint set;
  $\mathcal{X} \leftarrow \mathcal{F}(\mathcal{Q}, \mathtt{A})$;
  $C_p^i = \leftarrow \sigma(\mathcal{X}^i)$, compute for all $i = 1 \dots m$;
  $p \leftarrow p + 1$;
end
$C = \bigcup_{p=1}^{\mathcal{I}} C_p$;
OUTPUT: $\{C\}$

7 Track Refinement

During cross camera tracking, tracking results using tracks from distinct cameras as different constraint sets, might result in overlapping clusters (clusters with some common nodes), since we are not removing nodes of extracted clusters from the graph at every iteration. Thus, we propose a refinement step to help assign those ambiguous nodes (which are assigned to multiple clusters) to their right cluster. To achieve this, we employ the notion of centrality of fast-constrained dominant sets and the notion of reciprocal neighbors, that is, if two nodes (from different camera) are similar to each other, then their corresponding neighbors (nodes which are in their respective cluster) are expected to be similar too.

Let us assume we have $\mathcal{I}$ cameras and $\mathcal{K}^i$ represents the set corresponding to track i, while $\mathcal{K}_p^i$ is the subset of $\mathcal{K}^i$ that corresponds to the p^{th} camera. $\mathcal{M}_p^{l^i}$ is the membership score assigned to the l^{th} track in the set $\mathcal{C}_p^i$.

We use two constraints during track refinement stage, which helps us refining false positive association.
Constraint-1: *A track can not belong to two different sets generated using the same constraint set*, i.e. it must hold that:

$$|\mathcal{K}_p^i| \leq 1$$

Sets that do not satisfy the above inequality should be refined, as there is one or more tracks that exist in different sets of tracks collected using the same constraint,

i.e. T_p. The corresponding track is removed from all the sets which contain it and is assigned to the right set based on its membership score in each of the sets. Let us say the l^{th} track exists in q different sets, when tracks from camera p are taken as a constraint set, $|\mathcal{K}_p^l| = q$. The right set which contains the track, C_p^r, is chosen as:

$$C_p^r = \underset{C_p^i \in \mathcal{K}_p^l}{\arg\max} \left(|C_p^i| * \mathcal{M}_p^{l^i} \right).$$

where $i = 1, \ldots, |\mathcal{K}_p^l|$. This must be normalized with the cardinality of the set to avoid a bias towards smaller sets.

Constraint-2: *The maximum number of sets that contain track i should be equal to the number of cameras under consideration.* For $\mathcal{I}$ cameras, the cardinality of the set which contains sets with track i is not larger than $\mathcal{I}$, i.e.:

$$|\mathcal{K}^i| \leq \mathcal{I}.$$

If there are sets that do not satisfy the above condition, the tracks are refined based on the cardinality of the intersection of sets that contain the track by enforcing the reciprocal properties of the sets which contain a track. Assume we collect sets of tracks considering tracks from camera q as constraint set, and assume track ϑ in the set C_p^j, $p \neq q$ exists in more than one sets of C_q. The right set, C_q^r, for ϑ considering tracks from camera q as constraint set is chosen as:

$$C_q^r = \underset{C_q^i \in \mathcal{K}_q^\vartheta}{\arg\max} \left(C_q^i \cap C_p^j \right).$$

where $i = 1, \ldots, |\mathcal{K}_q^\vartheta|$.

8 Experimental Results

The proposed framework has been evaluated on a recently-released large dataset, MOTchallenge DukeMTMC [34, 38, 33]. Even though the main focus of this paper is on multi-target tracking in multiple non-overlapping cameras, we also perform additional experiments on MARS [51], one of the largest and most challenging video-based person re-identification dataset, to show that the proposed across-camera tracking approach can efficiently solve this task too.

DukeMTMC has been recently released to evaluate the performance of multi-target multi-camera tracking systems. It is the largest (to date), fully-annotated and calibrated high resolution 1080p, 60fps dataset that covers a single outdoor scene from 8 fixed synchronized cameras. The topology of cameras is shown in Fig. 4. The dataset consists of 8 videos of 85 minutes each from the 8 cameras, with 2,700 unique identities (IDs) in more than 2 millions frames in each video, containing from 0 to 54 people. The videos are split in three parts: (1) Trainval (first 50 minutes of the video), which is for training and validation; (2) Test-Hard (next 10 minutes after Trainval sequence); and (3) Test-Easy, which covers the last 25 minutes of the video. Some of the properties which make the dataset challenging include: huge amount of data to process; it contains 4,159 hand-overs; there are more than 1,800 self-occlusions (with 50% or more overlap); there are 891 people walking in front of only one camera.

MARS (Motion Analysis and Re-identification Set) is an extension of the Market-1501 dataset [51]. It has been collected from six near-synchronized cameras. It consists of 1,261 different pedestrians, who are captured by at least 2 cameras. The variations in poses, colors and illuminations of pedestrians, as well as the poor image quality, make it very difficult to yield high matching accuracy. Moreover, the dataset contains 3,248 distractors in order to make it more realistic. Deformable Part Model (DPM) [9] and GMMCP tracker [6] were used to automatically generate the tracklets (mostly 25-50 frames long). Since the video and the detections are not available we used the generated tracklets as an input to our framework.

Performance Measures: In addition to the standard Multi-Target Multi-Camera tracking performance measures, we evaluate our framework using additional measures recently proposed in [34]: Identification F-measure (IDF1), Identification Precision (IDP) and Identification Recall (IDR). The standard performance measures, such as CLEAR MOT, report the amount of incorrect decisions made by a tracker. Ristani *et al.* [34] argue and demonstrate that some system users may instead be more interested in how well they can determine who is where at all times. After pointing out that different measures serve different purposes, they proposed the three measures (IDF1, IDP and IDR) which can be applied both within- and across-cameras. These measure tracker's performance not by how often ID switches occur, but by how long the tracker correctly tracks targets.

***Identification precision IDP (recall IDR)*:** is the fraction of computed (ground truth) detections that are correctly identified.

***Identification F-measure IDF1*:** is the ratio of correctly identified detections over the average number of ground-truth and computed detections.

Since MOTA and its related performance measures under-report across-camera errors [34], we use them for the evaluation of our single camera tracking results.

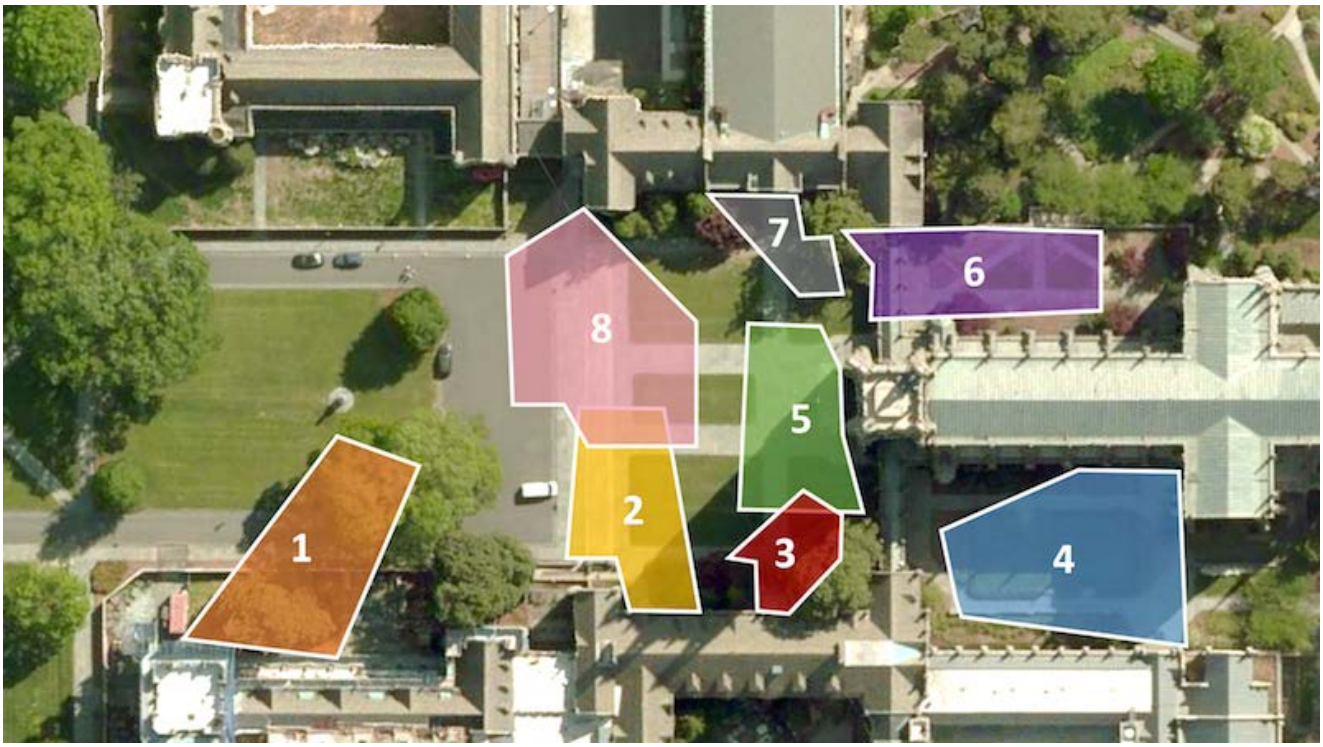

Fig. 4: Camera topology for DukeMTMC dataset. Detections from the overlapping fields of view of Cameras (#2 and #8) and (#3 and #5) are not considered.

The performance of the algorithm for re-identification is evaluated employing rank-1 based accuracy and confusion matrix using average precision (AP).

Implementation: In the implementation of our framework, we do not have parameters to tune. The affinity matrix $\mathtt{A}$ is constructed as follows (by adapting kernel trick distance function from [12]):

$$\mathtt{A}_{i,j} = 1 - \sqrt{\frac{\mathtt{K}(x_i, x_i) + \mathtt{K}(x_j, x_j) - 2 * \mathtt{K}(x_i, x_j)}{2}},$$

where $\mathtt{K}(x_i, x_j)$ is chosen as the Laplacian kernel

$$\mathtt{K}(x_i, x_j) = exp\left(-\gamma \parallel x_i - x_j \parallel_1\right).$$

The kernel parameter γ is set as the inverse of the median of pairwise distances.

In our similarity matrix for the final layer of the framework, which is sparse, we use spatio-temporal information based on the time duration and the zone of a person moving from one zone of a camera to other zone of another camera, which is learned from the Trainval sequence of DukeMTMC dataset. The affinity between track i and track j is different from zero if and only if they have a possibility, based on the direction a person is moving and the spatio-temporal information, to be linked and form a trajectory (across-camera tracks of a person). However, this may have a drawback due to *broken tracks* or track of a person who is standing and talking or doing other activities in one camera, which results in a track that does not meet the spatio-temporal constraints. To deal with this problem, we add, for the across-camera track's similarity, a path-based information as used in [49], i.e. if a track in camera i and a track in camera j have a probability to be linked, and similarly j in turn have a possibility to be linked with a track in camera z, the tracks in camera i and camera z are considered to have a possibility to be linked.

The similarity between two tracks is computed using the Euclidean distance of the max-pooled features. The max-pooled features are computed as the row maximum of the feature vector of individual patches of the given track, extracted from the last fully-connected layer of Imagenet pre-trained 50-layers Residual Network (ResNet_50) [14], fine-tuned using the Trainval sequence of DukeMTMC dataset. The network is fine-tuned with classification loss on the Trainval sequence, and activations of its last fully-connected layer are extracted, L2-normalized and taken as visual features. Cross-view Quadratic Discriminant Analysis (XQDA) [21] is then used for pairwise distance computation between instances. For the experiments on MARS, patch

	Methods	MOTA↑	MOTP↑	FAF↓	MT↑	ML↓	FP↓	FN↓	IDS↓	IDF1↑	IDP↑	IDR↑
Camera1	[34]	43.0	79.0	0.03	24	46	2,713	107,178	39	57.3	91.2	41.8
	[27]	42.9	79.0	0.03	24	46	2,911	107,156	41	57.8	91.9	42.2
	Ours	69.9	76.3	0.06	137	22	5,809	52,152	156	76.9	89.1	67.7
Camera2	[34]	44.8	78.2	0.51	133	8	47,919	53,74	60	68.2	69.3	67.1
	[27]	44.7	78.2	0.51	133	39	47,788	54,125	52	69.2	70.4	68.0
	Ours	71.5	74.6	0.09	134	21	8,487	43,912	75	81.2	90.9	73.4
Camera3	[34]	57.8	77.5	0.02	52	22	1,438	28,692	16	60.3	78.9	48.8
	[27]	57.8	77.5	0.02	52	22	1,438	28,692	19	59.8	78.2	48.4
	Ours	67.4	75.6	0.02	44	9	2,148	21,125	38	64.6	76.3	56.0
Camera4	[34]	63.2	80.2	0.02	36	18	2,209	19,323	7	73.5	88.7	62.8
	[27]	63.2	80.2	0.02	36	18	2,209	19,323	9	76.0	91.7	64.9
	Ours	76.8	76.6	0.03	45	4	2,860	10,689	18	84.7	91.2	79.0
Camera5	[34]	72.8	80.4	0.05	107	17	4,464	35,861	54	73.2	83.0	65.4
	[27]	72.6	80.4	0.05	107	7	4,713	35,849	46	73.3	83.0	65.6
	Ours	68.9	77.4	0.10	88	11	9,117	36,933	139	68.3	76.1	61.9
Camera6	[34]	73.4	80.2	0.06	142	27	5,279	45,170	55	77.2	87.5	69.1
	[27]	73.4	80.2	0.06	142	27	5,279	45,170	58	80.9	91.7	72.4
	Ours	77.0	77.2	0.05	136	11	4,868	38,611	142	82.7	91.6	75.3
Camera7	[34]	71.4	74.7	0.02	69	13	1,395	18,904	23	80.5	93.6	70.6
	[27]	71.4	74.7	0.02	69	13	1,395	18,904	23	80.5	93.6	70.6
	Ours	73.8	74.0	0.01	64	4	1,182	17,411	36	81.8	94.0	72.5
Camera8	[34]	60.7	76.7	0.03	102	53	2,730	52,806	46	72.4	92.2	59.6
	[27]	60.9	76.6	0.03	103	52	2,901	52,370	42	72.7	92.2	60.0
	Ours	63.4	73.6	0.04	92	28	4,184	47,565	91	73.0	89.1	61.0
Average	[34]	59.4	**78.7**	0.09	665	234	68,147	316,672	300	70.1	83.6	60.4
	[27]	59.3	**78.7**	0.09	666	234	68,634	361,589	**290**	71.2	84.8	61.4
	Ours	**70.9**	75.8	**0.05**	**740**	**110**	**38,655**	**268,398**	693	**77.0**	**87.6**	**68.6**

Table 1: The results show detailed (for each camera) and average performance of our and state-of-the-art approaches [34, 27] on the Test-Easy sequence of DukeMTMC dataset.

representation is obtained using CNN features used in [51]. The pairwise distances between instances are then computed in XQDA, KISSME [17] and Euclidean spaces.

8.1 Evaluation on DukeMTMC dataset:

In Table 1 and Table 2, we compare quantitative performance of our method with state-of-the-art multi-camera multi-target tracking method on the DukeMTMC dataset. As the ground truth for the test set is not publicly available, we compared with approaches whose results are present on the scoring board [1]. The symbol ↑ means higher scores indicate better performance, while ↓ means lower scores indicate better performance. The quantitative results of the trackers shown in Table 1 represent the performance on the Test-Easy sequence, while those in Table 2 show the performance on the Test-Hard sequence. For a fair comparison, we use the same detection responses obtained from MOTchallenge DukeMTMC as the input to our method. In both cases, the reported results of row 'Camera 1' to 'Camera 8' represent the within-camera tracking performances. The last row of the tables represent the average performance over 8 cameras. Both tabular results demonstrate that the proposed approach improves tracking performance for both sequences. In the Test-Easy sequence, the performance is improved by 11.5% in MOTA and 5.8% in IDF1 metrics, while in that of the Test-Hard sequence, our method produces 5% higher average MOTA score than [34], and 0.4% improvement is achieved in IDF1 w.r.t. [27]. Table 3 and Table 4, respectively, present Multi-Camera performance of our and state-of-the-art approaches [34, 27] on the Test-Easy and Test-Hard sequence of DukeMTMC dataset. We have improved IDF1 for both Test-Easy and Test-Hard sequences by about 4% and 3%, respectively.

Figures (8 - 11) depict sample qualitative results. Each person is represented by (similar color of) two bounding boxes, which represent the person's position at some specific time, and a track which shows the path s(he) follows. In Figure 8, all the four targets, even under significant illumination and pose changes, are successfully tracked in four cameras, where they appear. In Figure 9, target 714 is successfully tracked through three cameras. Observe its significant illumination and pose changes from camera 5 to camera 7. Figure 10 shows targets that move through camera 1, 6, 7 and 8. Finally, Figure 11 shows sample tracks of targets that appear in cameras 1 to 4.

[1] `https://motchallenge.net/results/DukeMTMCT/` (standing 01/13/2018)

	Methods	MOTA↑	MOTP↑	FAF↓	MT↑	ML↓	FP↓	FN↓	IDS↓	IDF1↑	IDP↑	IDR↑
Camera1	[34]	37.8	78.1	0.03	6	34	1,257	78,977	55	52.7	92.5	36.8
	[27]	37.4	78.1	0.04	6	35	1,575	79,189	61	52.5	91.9	36.7
	Ours	63.2	75.7	0.08	65	17	2,886	44,253	408	67.1	83.0	56.4
Camera2	[34]	47.3	76.5	0.74	68	12	26,526	46,898	194	60.6	65.7	56.1
	[27]	46.6	76.5	0.76	66	12	27,354	47,123	194	61.0	66.0	56.7
	Ours	54.8	73.9	0.24	62	16	8,653	54,252	323	63.4	78.8	53.1
Camera3	[34]	46.7	77.9	0.01	24	4	288	18,182	6	62.7	96.1	46.5
	[27]	46.7	77.9	0.01	24	4	288	18,182	6	62.7	96.1	46.5
	Ours	68.8	75.1	0.06	18	2	2,093	8,701	11	81.5	91.1	73.7
Camera4	[34]	85.3	81.5	0.04	21	0	1,215	2,073	1	84.3	86.0	82.7
	[27]	85.5	81.4	0.04	21	0	1,304	1,948	2	92.3	93.6	91.0
	Ours	75.6	77.7	0.05	17	0	1,571	3,888	61	82.3	87.1	78.1
Camera5	[34]	78.3	80.7	0.04	57	2	1,480	11,568	13	81.9	90.1	75.1
	[27]	78.3	80.7	0.04	57	2	1,480	11,568	13	81.9	90.1	75.1
	Ours	78.6	76.7	0.03	47	2	1,219	11,644	50	82.8	91.5	75.7
Camera6	[34]	59.4	76.7	0.14	85	23	5,156	77,031	225	64.1	81.7	52.7
	[27]	59.4	76.7	0.14	85	23	5,170	76,981	230	64.7	82.4	53.3
	Ours	53.3	76.5	0.17	68	36	5,989	88,164	547	53.1	71.2	42.3
Camera7	[34]	50.8	73.3	0.08	43	23	2,971	38,912	148	59.6	81.2	47.1
	[27]	50.6	73.3	0.09	42	23	3,090	38,995	145	59.8	81.4	47.2
	Ours	50.8	74.0	0.05	34	20	1,935	39,865	266	60.6	84.7	47.1
Camera8	[34]	73.0	75.9	0.02	34	5	706	9735	10	82.4	94.9	72.8
	[27]	73.0	75.9	0.02	34	5	717	9,718	10	82.4	94.9	72.8
	Ours	70.0	72.6	0.06	37	6	2,297	9,306	26	81.3	90.3	73.9
Average	[34]	54.6	**77.1**	0.14	338	103	39,599	283,376	652	64.5	81.2	53.5
	[27]	54.4	**77.1**	0.14	335	104	40,978	283,704	**661**	65.0	81.8	54.0
	Ours	**59.6**	75.4	**0.09**	**348**	**99**	**26,643**	**260,073**	1637	**65.4**	**81.4**	**54.7**

Table 2: The results show detailed (for each camera) and average performance of our and state-of-the-art approaches [34, 27] on the Test-Hard sequence of DukeMTMC dataset.

	Methods	IDF1↑	IDP↑	IDR↑
Multi-Camera	[34]	56.2	67.0	48.4
	[27]	34.9	41.6	30.1
	Ours	**60.0**	**68.3**	**53.5**

Table 3: Multi-camera performance of our and state-of-the-art approaches [34, 27] on the Test-Easy sequence of DukeMTMC dataset.

	Methods	IDF1↑	IDP↑	IDR↑
Multi-Camera	[34]	47.3	59.6	39.2
	[27]	32.9	41.3	27.3
	Ours	**50.9**	**63.2**	**42.6**

Table 4: Multi-Camera performance of our and state-of-the-art approaches [34, 27] on the Test-Hard sequence of DukeMTMC dataset.

Methods	rank 1
HLBP [42] + XQDA	18.60
BCov [25] + XQDA	9.20
LOMO [21] + XQDA	30.70
BoW [52] + KISSME	30.60
SDALF [8] + DVR	4.10
HOG3D [16] + KISSME	2.60
CNN + XQDA [51]	65.30
CNN + KISSME [51]	65.00
Ours	**68.22**

Table 5: The table shows the comparison (based on rank-1 accuracy) of our approach with the state-of-the-art approaches on MARS dataset.

8.2 Evaluation on MARS dataset:

In this experiment, given the query (constraint), we first extract a cluster which is guaranteed to contain the query and other members from the gallery which are highly similar with each other. We then use the notion of centrality of FCDSC, where a membership score (which depicts their proximity to the query) is assigned to each element in the cluster. These scores are then used to sort the extracted tracks. In Table 5 we compare our results (using the same settings as in [51]) on MARS dataset. The proposed approach achieves about 3% of improvement. In Table 6 the results show performance of our and state-of-the-art approach [51] in solving the within- (average of the diagonal of the confusion matrix, Fig. 5) and across-camera (off-diagonal average) ReID using average precision. Our approach shows up to 10% improvement in the across-camera ReID and up to 6% improvement in the within-camera ReID.

To show how much meaningful the notion of centrality of fast-constrained dominant set is, we conduct an experiment on the MARS dataset computing the fi-

Feature+Distance	Methods	Within	Across
CNN + Eucl	[51]	0.59	0.28
	Ours (PD)	0.59	0.29
	Ours (MS)	**0.60**	**0.29**
CNN + KISSME	[51]	0.61	0.34
	Ours (PD)	0.64	0.41
	Ours (MS)	**0.67**	**0.44**
CNN + XQDA	[51]	0.62	0.35
	Ours (PD)	0.65	0.42
	Ours (MS)	**0.68**	**0.45**

Table 6: The results show performance of our, using pairwise distance (PD) and membership score (MS), and state-of-the-art approach [51] in solving within- and across-camera ReID using average precision on MARS dataset using CNN feature and different distance metrics.

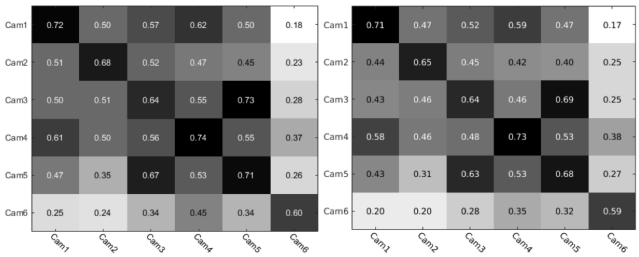

Fig. 5: The results show the performance of our algorithm on MARS (both using CNN + XQDA) when the final ranking is done using membership score (**left**) and using pairwise euclidean distance (**right**).

nal ranking using the membership score and pairwise distances. The confusion matrix in Fig. 5 shows the detail result of both the within-cameras (diagonals) and across-cameras (off-diagonals), as we consider tracks from each camera as query. Given a query, a set which contains the query is extracted using the fast-constrained dominant set framework. Note that fast-constraint dominant set comes with the membership scores for all members of the extracted set. We show in Figure 5 the results based on the final ranking obtained using membership scores (**left**) and using pairwise Euclidean distance between the query and the extracted nodes (**right**). As can be seen from the results in Table 6 (average performance) the use of membership score outperforms the pairwise distance approach, since it captures the interrelation among targets.

8.3 Computational Time

Figure 6 shows the time taken for each track - from 100 randomly selected (query) tracks from MARS dataset - to be associated with the rest of the (gallery) tracks, after running CDSC [48] and the newly proposed FCDSC

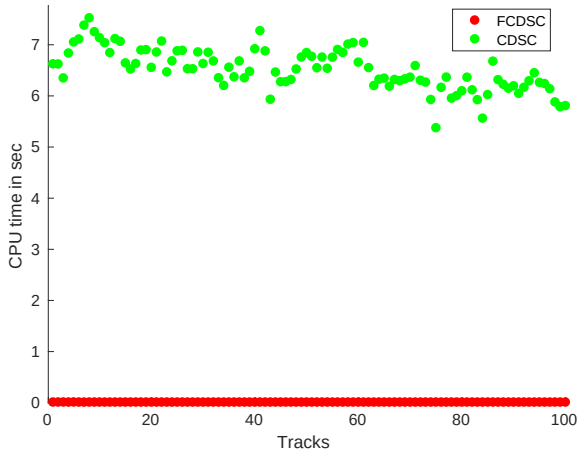

Fig. 6: CPU time taken for each track association using our proposed approach (FCDSC) and CDSC.

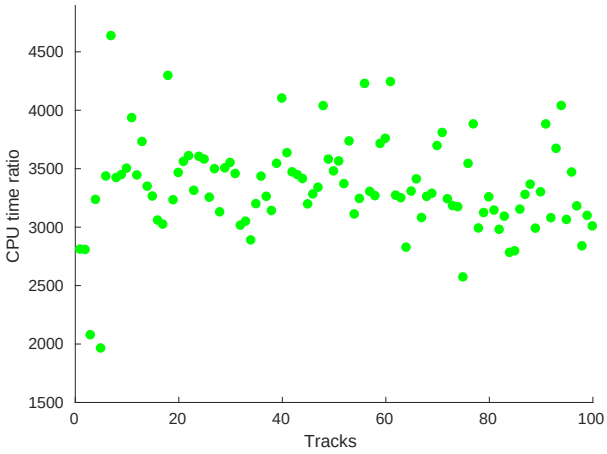

Fig. 7: The ratio of CPU time taken between CDSC and proposed approach (FCDSC), computed as CPU time for CDSC/CPU time for FCDSC.

over the constructed graph. The vertical axis is the CPU time in seconds and horizontal axis depicts the track IDs. As it is evident from the plot, our approach takes a fraction of second (red points in Fig. 6). Conversely, the CDSC takes up to 8 seconds for some cases (green points in Fig. 6). Fig. 7 further elaborates how fast our proposed approach is over CDSC, where the vertical axis represents the ratio between CDSC (numerator) and FCDSC (denominator) in terms of CPU time. This ratio ranges from 2000 (the proposed FCDSC 2000x faster than CDSC) to a maximum of above 4500.

In our non-optimized Matlab code, with 60GB RAM, i7, 3.1GHz windows machine, the whole tracking algorithms (excluding detection), runs at 18 fps.

9 Conclusions

In this paper we presented a novel fast-constrained dominant set clustering (FCDSC) framework for solving multi-target tracking problem in multiple non-overlapping camera settings. The proposed method utilizes a three-layers hierarchical approach, where within-camera tracking is solved using first two layers of our framework resulting in tracks for each person, and later in the third layer the proposed across-camera tracker merges tracks of the same person across different cameras. Experiments on a challenging real-world dataset (MOTchallenge DukeMTMCT) validate the effectivness of our model.

We performed additional experiments to show effectiveness of the proposed approach on one of the largest video-based people re-identification dataset (MARS). Here, each query is treated as a constraint set and its corresponding members in the resulting constrained dominant set cluster are considered as possible candidate matches to their corresponding query.

There are few directions we would like to pursue in our future research. In this work, we considered a static cameras with known topology (layout of the scene, including the positions of the cameras in the scene), but it is important for the approach to be able to handle more challenging scenarios, where some views are from cameras with ego motion (e.g., PTZ cameras or taken from mobile devices) with unknown camera topology. Moreover, here we considered features from static images, however, we believe video features which can be extracted using LSTM could boost the performance and help us extending the method to handle challenging scenarios.

Acknowledgements This research is based upon work supported in parts by the U. S. Army Research Laboratory and the U. S. Army Research Office (ARO) under contract/grant number W911NF-14-1-0294; and the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. D17PC00345. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] William Brendel, Mohamed Amer, and Sinisa Todorovic. "Multiobject tracking as maximum weight independent set". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 1273–1280.

[2] Yinghao Cai and Gérard G. Medioni. "Exploring context information for inter-camera multiple target tracking". In: *IEEE Workshop on Applications of Computer Vision (WACV)*. 2014, pp. 761–768.

[3] Xiaotang Chen, Kaiqi Huang, and Tieniu Tan. "Object tracking across non-overlapping views by learning inter-camera transfer models". In: *Pattern Recognition* 47.3 (2014), pp. 1126–1137.

[4] De Cheng et al. "Part-aware trajectories association across non-overlapping uncalibrated cameras". In: *Neurocomputing* 230 (2017), pp. 30–39.

[5] Dung Nghi Truong Cong et al. "Video Sequences Association for People Re-identification across Multiple Non-overlapping Cameras". In: *IAPR International Conference on Image Analysis and Processing (ICIAP)*. 2009, pp. 179–189.

[6] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. "GMMCP tracker: Globally optimal Generalized Maximum Multi Clique problem for multiple object tracking". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4091–4099.

[7] Tiziana D'Orazio, Pier Luigi Mazzeo, and Paolo Spagnolo. "Color Brightness Transfer Function evaluation for non overlapping multi camera tracking". In: *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*. 2009, pp. 1–6.

[8] Michela Farenzena et al. "Person re-identification by symmetry-driven accumulation of local features". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 2360–2367.

[9] Pedro F. Felzenszwalb et al. "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 32.9 (2010), pp. 1627–1645.

[10] Yue Gao et al. "Symbiotic Tracker Ensemble Toward A Unified Tracking Framework". In: *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* 24.7 (2014), pp. 1122–1131.

[11] Andrew Gilbert and Richard Bowden. "Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns

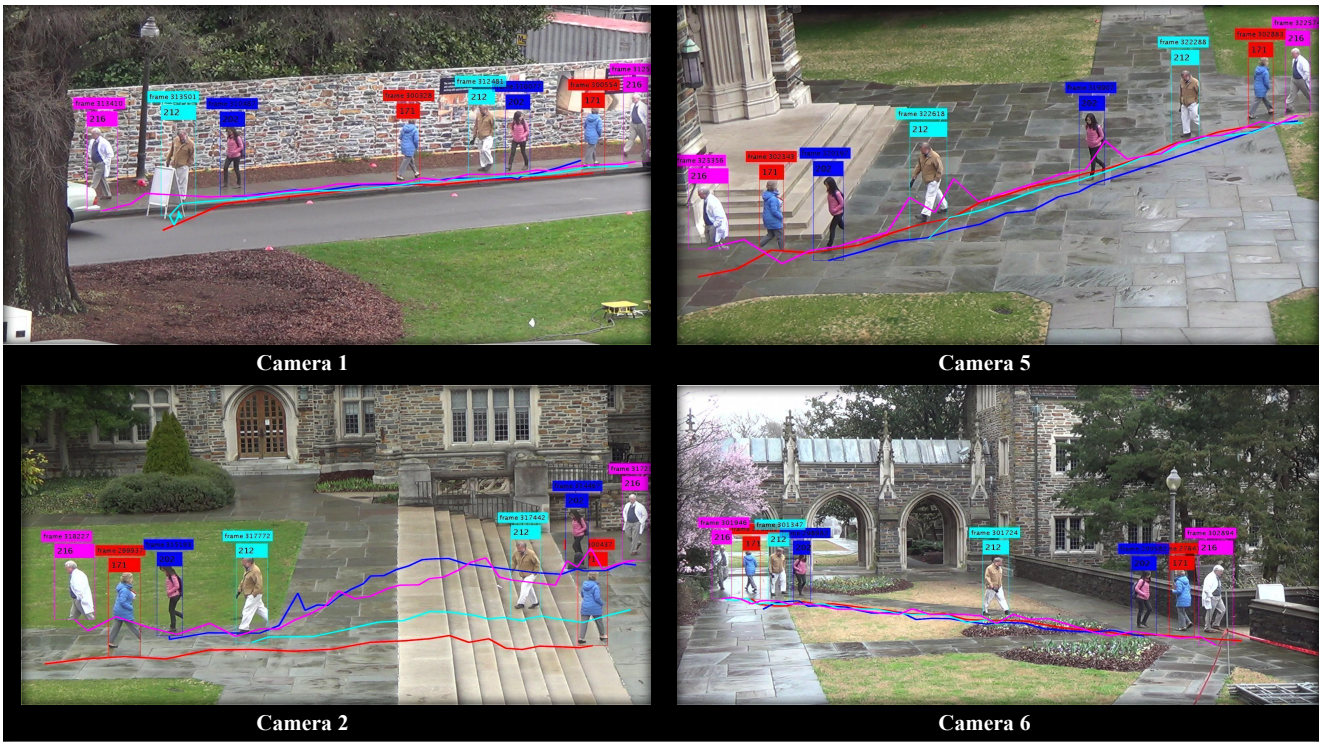

Fig. 8: Sample qualitative results of the proposed approach on DukeMTMC dataset. All the four targets (with id, 216 (pink), 212 (light blue),202 (blue) and 171 (red)), even under significant illumination and pose changes, are successfully tracked in four cameras (cam 1,5,2 and 6), where they appear. (Best viewed in color and zoomed)

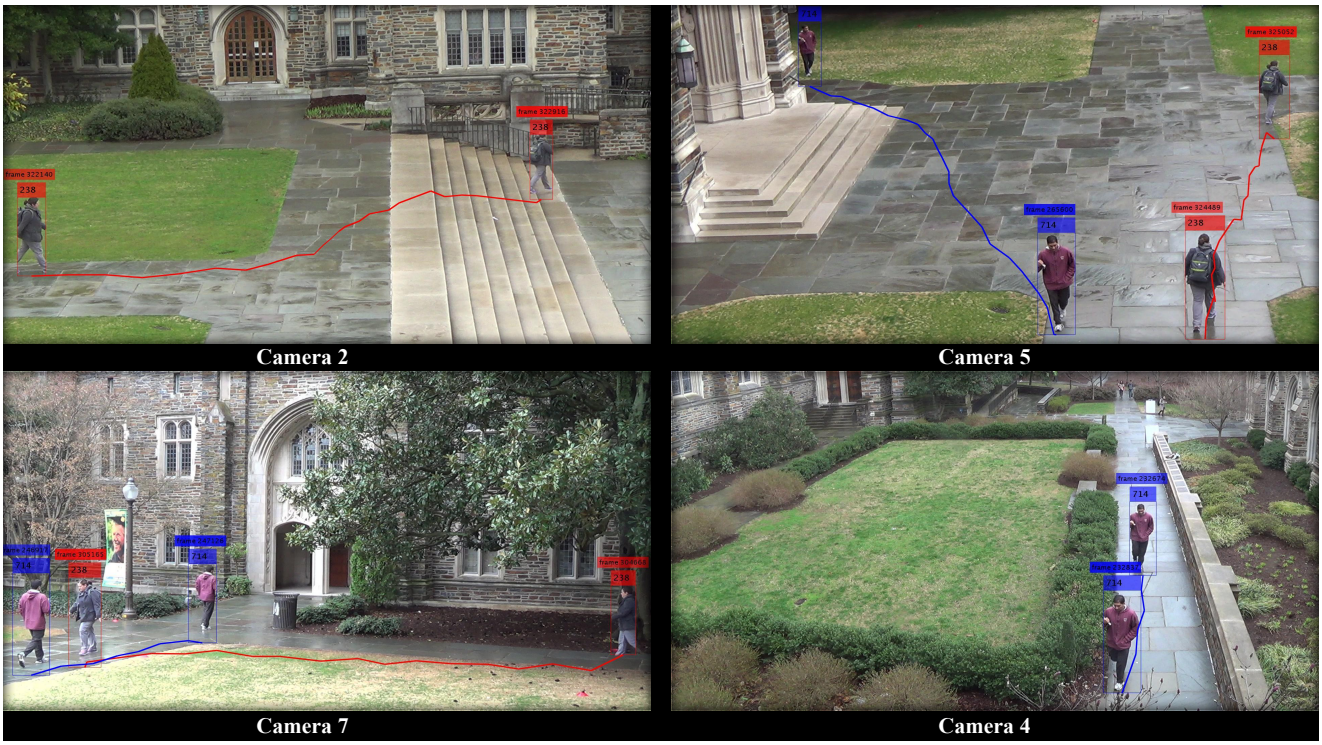

Fig. 9: Target 714 (blue) is successfully tracked through camera 5,7 and 4. Discern its significant illumination and pose changes from camera 5 to camera 7. Similarly, Target 238 (red) is correctly tracked in camera 2,5 and 7. (Best viewed in color and zoomed)

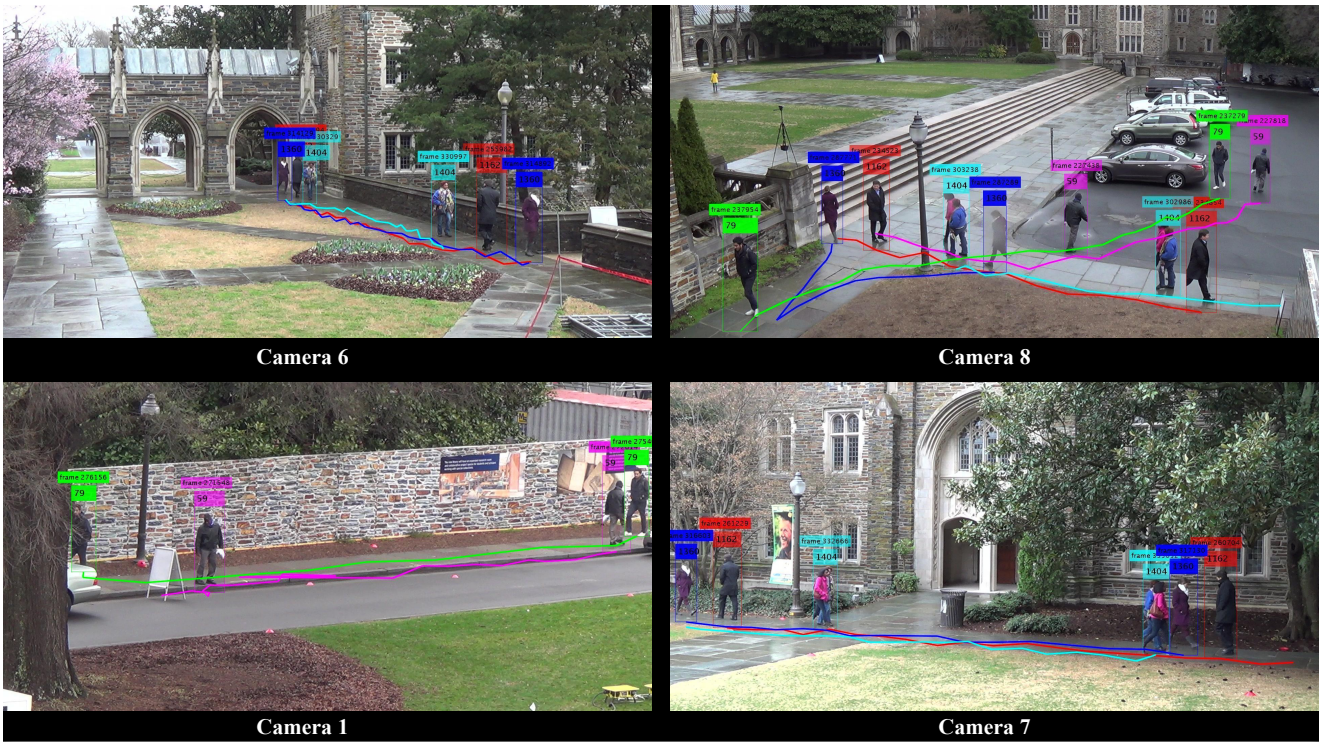

Fig. 10: Regardless of the change in pose and illumination, targets 1404 (light blue), 1162 (red) and 1360 (blue) are tracked correctly in camera 6, 8, and 7. Similarly, target 59 (pink) and 79 (green) are correctly tracked as they appear in camera 1 and 8. (Best viewed in color and zoomed)

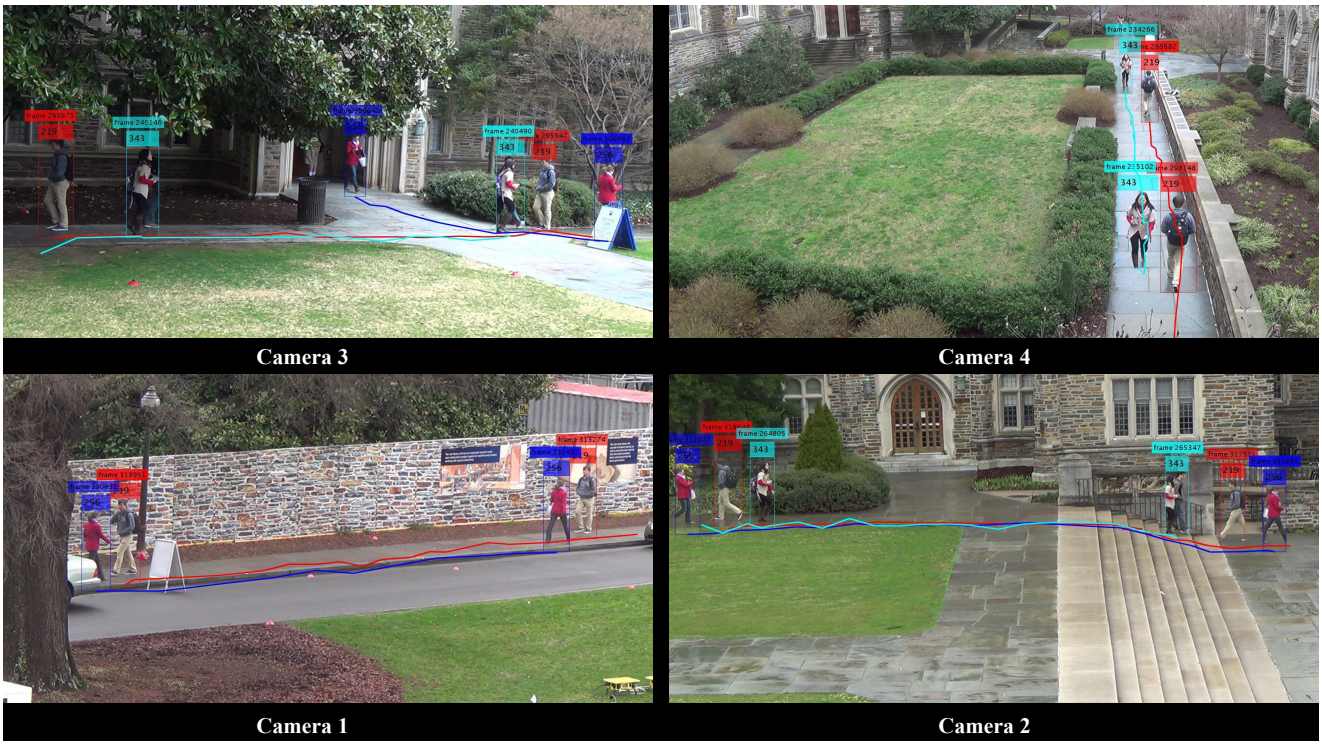

Fig. 11: The proposed approach is able to track target 219 (red), with heavy change in pose and illumination, correctly in all four cameras (3,4,1 and 2). Target 256 (blue) is correctly tracked in camera 3,1 and 2. Similarly, our method correctly identity target 343 (light blue) as she appears in camera 3,4 and 2. (Best viewed in color and zoomed)

of Activity". In: *European Conference on Computer Vision (ECCV)*. 2006, pp. 125–136.

[12] Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, eds. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2005.

[13] Seyed Hamid Rezatofighi et al. "Joint probabilistic data association revisited". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3047–3055.

[14] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[15] Omar Javed et al. "Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views". In: *Computer Vision and Image Understanding* 109.2 (2008), pp. 146–162.

[16] Alexander Kläser, Marcin Marszalek, and Cordelia Schmid. "A Spatio-Temporal Descriptor Based on 3D-Gradients". In: *British Machine Vision Conference (BMVC)*. 2008, pp. 1–10.

[17] Martin Köstinger et al. "Large scale metric learning from equivalence constraints". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 2288–2295.

[18] Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. "Inter-camera Association of Multi-target Tracks by On-Line Learned Appearance Affinity Models". In: *European Conference on Computer Vision (ECCV)*. 2010, pp. 383–396.

[19] Bastian Leibe, Konrad Schindler, and Luc Van Gool. "Coupled detection and trajectory estimation for multi-object tracking". In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 2007, pp. 1–8.

[20] Xi Li et al. "A survey of appearance models in visual object tracking". In: *ACM transactions on Intelligent Systems and Technology (TIST)* 4.4 (2013), p. 58.

[21] Shengcai Liao et al. "Person re-identification by Local Maximal Occurrence representation and metric learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2197–2206.

[22] Shengcai Liao et al. "Person re-identification by Local Maximal Occurrence representation and metric learning". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 2197–2206.

[23] Hairong Liu, Longin Jan Latecki, and Shuicheng Yan. "Fast Detection of Dense Subgraphs with Iterative Shrinking and Expansion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 35.9 (2013), pp. 2131–2142.

[24] David G Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Vol. 3. 2008.

[25] Bingpeng Ma, Yu Su, and Frédéric Jurie. "Covariance descriptor based on bio-inspired features for person re-identification and face verification". In: *Image Vision Computing* 32.6-7 (2014), pp. 379–390.

[26] Andrii Maksai, Xinchao Wang, and Pascal Fua. "What players do with the ball: a physically constrained interaction modeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 972–981.

[27] Andrii Maksai et al. "Non-Markovian Globally Consistent Multi-Object Tracking". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 2563–2573.

[28] Niall McLaughlin, Jesús Martínez del Rincón, and Paul C. Miller. "Recurrent Convolutional Network for Video-Based Person Re-identification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1325–1334.

[29] Massimiliano Pavan and Marcello Pelillo. "Dominant sets and pairwise clustering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 29.1 (2007), pp. 167–172.

[30] Massimiliano Pavan and Marcello Pelillo. "Efficient Out-of-Sample Extension of Dominant-Set Clusters". In: *Annual Conference on Neural Information Processing Systems (NIPS)*. 2004, pp. 1057–1064.

[31] Bryan James Prosser, Shaogang Gong, and Tao Xiang. "Multi-camera Matching using Bi-Directional Cumulative Brightness Transfer Functions". In: *British Machine Vision Conference (BMVC)*. 2008, pp. 1–10.

[32] Ergys Ristani and Carlo Tomasi. "Features for Multi-Target Multi-Camera Tracking and Re-Identification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6036–6046.

[33] Ergys Ristani and Carlo Tomasi. "Tracking Multiple People Online and in Real Time". In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 444–459.

[34] Ergys Ristani et al. "Performance Measures and a Data Set for Multi-target, Multi-camera Tracking". In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 17–35.

[35] Samuel Rota Bulò, Marcello Pelillo, and Immanuel M. Bomze. "Graph-based quadratic optimization:

A fast evolutionary approach". In: *Computer Vision and Image Understanding* 115.7 (2011), pp. 984–995.

[36] Arnold WM Smeulders et al. "Visual tracking: An experimental survey". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1 (2013), p. 1.

[37] John Maynard Smith. "Evolution and the Theory of Games". In: *Did Darwin Get It Right?* Springer, 1988, pp. 202–215.

[38] F. Solera et al. "Tracking Social Groups Within and Across Cameras". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2016).

[39] Satyam Srivastava, Ka Ki Ng, and Edward J. Delp. "Color correction for object tracking across multiple cameras". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 1821–1824.

[40] Bing Wang et al. "Tracklet association with online target-specific metric learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1234–1241.

[41] Taiqing Wang et al. "Person Re-identification by Video Ranking". In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 688–703.

[42] Fei Xiong et al. "Person Re-Identification Using Kernel-Based Metric Learning Methods". In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 1–16.

[43] Alper Yilmaz, Omar Javed, and Mubarak Shah. "Object tracking: A survey". In: *Acm computing surveys (CSUR)* 38.4 (2006), p. 13.

[44] K. Yoon, Y. Song, and M. Jeon. "Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views". In: *IET Image Processing* 12.7 (2018), pp. 1175–1184. ISSN: 1751-9659.

[45] Jinjie You et al. "Top-Push Video-Based Person Re-identification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1345–1353.

[46] Shoou-I Yu et al. "The solution path algorithm for identity-aware multi-object tracking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3871–3879.

[47] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. "GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs". In: *European Conference on Computer Vision (ECCV)*. 2012, pp. 343–356.

[48] Eyasu Zemene and Marcello Pelillo. "Interactive Image Segmentation Using Constrained Dominant Sets". In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 278–294.

[49] Eyasu Zemene and Marcello Pelillo. "Path-Based Dominant-Set Clustering". In: *ICIAP*. 2015, pp. 150–160.

[50] Shu Zhang, Yingying Zhu, and Amit K. Roy-Chowdhury. "Tracking multiple interacting targets in a camera network". In: *Computer Vision and Image Understanding* 134 (2015), pp. 64–73.

[51] Liang Zheng et al. "MARS: A Video Benchmark for Large-Scale Person Re-Identification". In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 868–884.

[52] Liang Zheng et al. "Scalable Person Re-identification: A Benchmark". In: *International Conference on Computer Vision (ICCV)*. 2015, pp. 1116–1124.