

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

Yonatan Tariku Tesfaye\*, *Student Member, IEEE*, Eyasu Zemene\*, *Student Member, IEEE*, Andrea Prati, *Senior member, IEEE*, Marcello Pelillo, *Fellow, IEEE*, and Mubarak Shah, *Fellow, IEEE*

**Abstract**—In this paper, a unified three-layer hierarchical approach for solving tracking problems in multiple non-overlapping cameras 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 merging tracks of the same person in all cameras in a simultaneous fashion. To best serve our purpose, a constrained dominant sets clustering (CDSC) technique, a parametrized version of standard quadratic optimization, is employed to solve both tracking tasks. The tracking problem is casted as finding constrained dominant sets from a graph. That is, given a constraint set and a graph, CDSC generates cluster (or clique), which forms a compact and coherent set that contains a subset of the constraint set. The approach is based on a parametrized family of quadratic programs that generalizes the standard quadratic optimization problem. In addition to having a unified framework that simultaneously solves within- and across-camera tracking, the third layer helps link broken tracks of the same person occurring during within-camera tracking. A standard algorithm to extract constrained dominant set from a graph is given by the so-called replicator dynamics whose computational complexity is quadratic per step which makes it handicapped for large-scale applications. In this work, we propose a fast algorithm, based on dynamics from evolutionary game theory, which is efficient and salable to large-scale real-world applications. We have tested this approach on a very large and challenging dataset (namely, MOTchallenge DukeMTMC) and show that the proposed framework outperforms the current state of the art. Even though the main focus of this paper is on multi-target tracking in non-overlapping cameras, proposed approach can also be applied to solve *re-identification* problem. Towards that end, we also have performed experiments on MARS, one of the largest and challenging video-based person re-identification dataset, and have obtained excellent results. These experiments demonstrate the general applicability of the proposed framework for non-overlapping across-camera tracking and person re-identification tasks.

**Index Terms**—Quadratic optimization, Multi-target multi-camera tracking, Dominant Sets, Constrained Dominant Sets

## 1 INTRODUCTION

As the need for visual surveillance grow, a large number of cameras have been deployed to cover large and wide areas like airports, shopping malls, city blocks etc.. Since the fields of view 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 the work presented in this paper, the goal is to track multiple targets and maintain their identities as they move from one camera to the 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 have been significant progresses in both problems separately, tracking multiple target jointly in both within and across non-overlapping cameras remains a less explored topic. Most approaches, which solve multi-target tracking in multiple non-overlapping cameras [1], [2], [3], [4], [5], assume tracking within each camera has already been performed and try to solve tracking problem only in non-overlapping cameras; the results obtained from such approaches are far from been optimal [4].

In this paper, we propose a hierarchical approach in which we first determine tracks within each camera, (Figure 1(a)) by solving data association, and later we associate tracks of the same person 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 problem in a hierarchical manner is common: tracklets are generated within short temporal window first and later they are merged to form full tracks (or trajectories) [6], [7], [8]. Often, across-camera tracking is more challenging than solving within-camera tracking due to the fact that appearance of people may exhibit significant differences due to illumination variations

- \* The first and second authors have equal contribution.
- Y. T. Tesfaye is with the department of Design and Planning in Complex Environments of the University IUAV of Venice, Italy. E-mail: y.tesfaye@stud.iuav.it
- E. Zemene and M. Pelillo are with the department of Computer Science, Ca' Foscari University of Venice, Italy. E-mail: {eyasu.zemene, pelillo}@unive.it
- A. Prati is with the department of Department of Engineering and Architecture of the University of Parma, Italy. E-mail: andrea.prati@unipr.it
- M. Shah is with the Center for Research in Computer Vision (CRCV), University of Central Florida, USA. E-mail: {haroon,shah}@eecs.ucf.edu

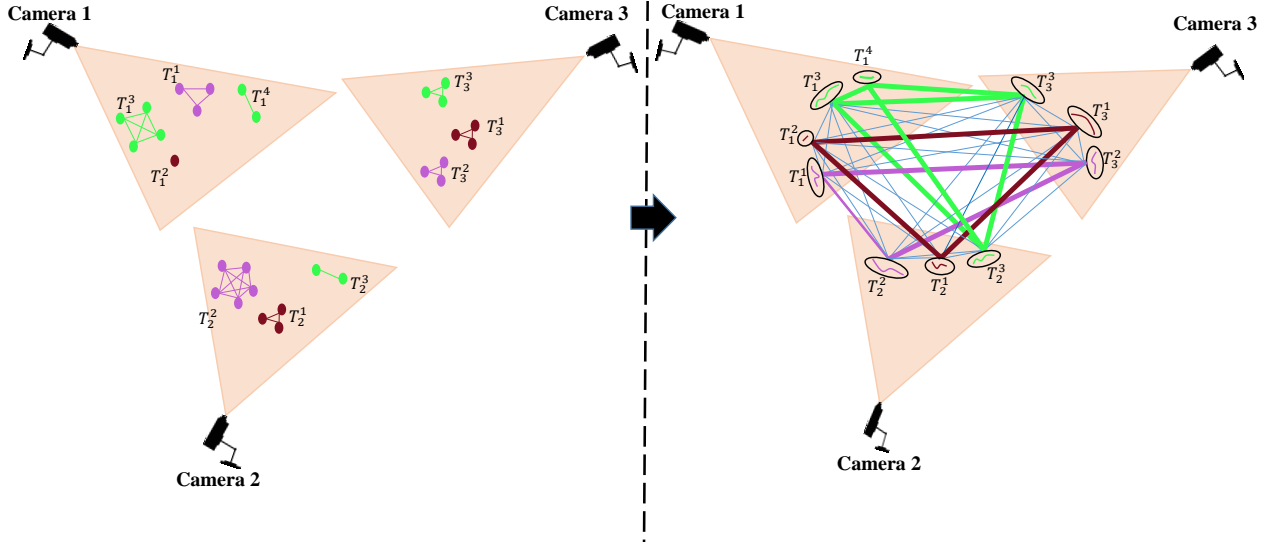


Fig. 1: A general idea of the proposed framework. (a) First, tracks are determined within each camera, then (b) tracks of the same person from different non-overlapping cameras are associated, solving the across-camera tracking. Nodes in (a) represent tracklets and nodes in (b) represent tracks. The  $i^{th}$  track of camera  $j$ ,  $T_j^i$ , is a set of tracklets that form a clique. In (b) each clique in different colors represent tracks of the same person in non-overlapping cameras. Similar color represents the same person. (Best viewed in color)

and pose changes between cameras.

Therefore, this paper proposes a unified three-layer framework to solve both within- and across-camera tracking. In the first two layers, we generate tracks within each camera and in the third layer we associate all tracks of the same person across all cameras in a simultaneous fashion.

To best serve our purpose, a constrained dominant sets clustering (CDSC) technique, a parametrized version of standard quadratic optimization, is employed to solve both tracking tasks. The tracking problem is casted as finding constrained dominant sets from a graph. That is, given a constraint set and a graph, CDSC generates cluster (or clique), which forms a compact and coherent set that contains all or part of the constraint set. *Clusters* represent tracklets and tracks in the first and second layers, respectively. 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. Also, the proposed across-camera tracking method using CDSC and later followed by refinement step offers several advantages. More specifically, CDSC 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 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 clique, 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 formulate multi-target tracking in multiple non-overlapping cameras as finding constrained dominant sets from a graph. We propose a three-layer hierarchical approach, in which we first solve within-camera tracking using the first two layers, and using the third layer we solve the across-camera tracking problem.
- We propose a technique to further speed up our optimization by reducing the search space, that is, instead of running the dynamics over the whole graph, we localize it on the sub graph selected using the dominant distribution, which is much smaller than the original graph.
- Experiments are performed on MOTchallenge DukeMTMCT dataset and MARS dataset, and 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. Overall proposed approach for within- and across-cameras tracking modules is summarized in section 3, while sections 3.2 and 3.3 provide more in details of the two modules. In section 4, we present the proposed approach to further speed up our method. Experimental results are presented in Section 5. Finally, section 6 concludes the paper.

## 2 RELATED WORK

Object tracking is a challenging computer vision problem and has been one of the most active research areas for many

years. In general, it can be divided in two broad categories: tracking in single and multiple cameras. Single camera object tracking associates object detections across frames in a video sequence, so as to generate the object motion trajectory over time. Multi-camera tracking aims to solve handover problem from one camera view to another and hence establishes target correspondences among different cameras, so as to achieve consistent object labelling across all the camera views. Early multi-camera target tracking research works fall in different categories as follows. Target tracking with partially overlapping camera views has been researched extensively during the last decade [9], [10], [11], [12], [13], [14]. Multi target tracking across multiple cameras with disjoint views has also been researched in [1], [2], [3], [4], [5]. Approaches for overlapping field of views compute spatial proximity of tracks in the overlapping area, while approaches for tracking targets across cameras with disjoint fields of view, leverage appearance cues together with spatio-temporal information.

Almost all early multi-camera research works try to address only across-camera tracking problems, assuming that within-camera tracking results for all cameras are given. Given tracks from each camera, similarity among tracks is computed and target correspondence across cameras is solved, using the assumption that a track of a target in one camera view can match with at most one target track in another camera view. Hungarian algorithm [15] and bipartite graph matching [3] formulations are usually used to solve this problem. Very recently, however, researchers have argued that assumptions of cameras having overlapping fields of view and the availability of intra-camera tracks are unrealistic [4]. Therefore, the work proposed in this paper addresses the more realistic problem by solving both within- and across-camera tracking in one joint framework.

In the rest of this section, we first review the most recent works for single camera tracking, and then describe the previous related works on multi-camera multi-view tracking.

Single camera target tracking associates target detections across frames in a video sequence in order to generate the target motion trajectory over time. Zamir *et al.* [6] formulate tracking problem as generalized maximum clique problem (GMCP), where the relationships between all detections in a temporal window are considered. In [6], a cost to each clique is assigned and the selected clique maximizes a score function. Nonetheless, the approach is prone to local optima as it uses greedy local neighbourhood search. Deghan *et al.* [7] cast tracking as a generalized maximum multi clique problem (GMMCP) and follow a joint optimization for all the tracks simultaneously. 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. Tesfaye *et al.* [8] consider all the pairwise relationships between detection responses in a temporal sliding window, which is used as an input to their optimization based on fully-connected edge-weighted graph. They formulate tracking as finding dominant set clusters. Though the dominant set framework is effective in extracting compact sets from a graph [16][17][18] [19] [20], it follows a pill-off strategy to enumerate all possible clusters, that is, at each iteration it

removes the found cluster from the graph which results in a change in scale (number of nodes in a graph) of the original problem. In this paper, we propose a multiple target tracking approach, which in contrast to previous works, does not need additional nodes to handle occlusion nor encounters change in the scale of the problem.

Across-camera tracking aims to establish target correspondences among trajectories from different cameras so as to achieve consistent target labelling across all camera views. It 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 illumination of the appearance cues has been leveraged using different techniques such as Brightness Transfer Functions (BTFs). To handle the appearance change of a target as it moves from one camera to another, the authors in [21] show that all brightness transfer functions from a given camera to another camera lie in a low dimensional subspace, which is learned by employing probabilistic principal component analysis and used for appearance matching. Authors of [22] used an incremental learning method to model the colour variations and [23] proposed a Cumulative Brightness Transfer Function, which is a better use of the available colour information from a very sparse training set. Performance comparison of different variations of Brightness Transfer Functions can be found in [24]. Authors in [25] 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 Brightness Transfer Functions or color correction models requires 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 cue which improve multi-target tracking performance [26], [27], [28], [29], [30], [31]. Chen *et al.* [26] 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 (MRF). Kuo *et al.* [27] used Multiple Instance Learning (MIL) to learn an appearance model, which effectively combines multiple image descriptors and their corresponding similarity measurements. The proposed appearance model combined with spatio-temporal information improved across-camera track association solving the target handover problem across cameras. Gao *et al.* [28] employ tracking results of different trackers and use their spatio-temporal correlation, which help them enforce tracking consistency and establish pairwise correlation among multiple tracking results. Zha *et al.* [29] formulated tracking of multiple interacting targets as a network flow problem, for which the solution can be obtained by the K-shortest paths algorithm. Spatio-temporal relationships among targets are utilized to identify group merge and split events. In [30] spatio-temporal context is used for collecting samples for discriminative appearance learning, where target-specific appearance models are learned to distinguish different people from each other.

And the relative appearance context models inter-object appearance similarities for people walking in proximity and helps disambiguate individual appearance matching across cameras.

The problem of target tracking across multiple non-overlapping cameras is also tackled in [32] by extending their previous single camera tracking method [33], where they formulate the tracking task as a graph partitioning problem. Authors in [31], learn across-camera transfer models including both spatio-temporal and appearance cues. While a color transfer method is used to model the changes of color across cameras for learning across-camera appearance transfer models, the spatio-temporal model is learned using an unsupervised topology recovering approach. Recently Chen *et al.* [5] argued that low-level information (appearance model and spatio-temporal information) is unreliable for tracking across non-overlapping cameras, and integrated contextual information such as social grouping behaviour. They formulate tracking using an online-learned Conditional Random Field (CRF), which favours track associations that maintain group consistency. In this paper, for tracks to be associated, besides their high pairwise similarity (computed using appearance and spatio-temporal cues), their corresponding constrained dominant sets should also be similar.

Another recent popular research topic, video-based person re-identification (ReID) [34], [35], [36], [37], [38], [39], [40], [41], [42], 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. Compared to most video-based ReID approaches, in which only pairwise similarity between the probes and gallery is exploited, our across-camera tracking framework not only considers the relationship between probes and gallery but it also takes in to account the relationship among tracks in the gallery.

### 3 OVERALL APPROACH

In this section, first we briefly introduce the basic definitions and properties of constrained dominant set clustering. This is followed by formulation of within- and across-camera tracking.

#### 3.1 Constrained Dominant Set clustering.

As introduced in [43], constrained dominant set clustering, a constrained quadratic optimization program, is an efficient and accurate approach, which has been applied for interactive image segmentation. The approach generalizes dominant set framework [17], 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  $Q \subseteq V$ , where  $V, E$  and  $w$ , respectively, denote the set of nodes (of cardinality  $n$ ), edges and edge weights. The objective is to find the sub-graph that contains all or some of elements of the constraint set, which forms a coherent and compact set.

In our formulation, in the first layer, each node in our graph represents a short-tracklet along a temporal window

(typically 15 frames). Applying constrained dominant set clustering here aim at determining cliques in this graph, which correspond to tracklets. Likewise, each node in a graph in the second layer represents a tracklet, obtained from the first layer, and CDSC is applied here to determine cliques, which correspond to tracks. Finally, in the third layer, nodes in a graph correspond to tracks from different non-overlapping cameras, obtained from the second layer, and CDSC is applied to determine cliques, which relate tracks of the same person across non-overlapping cameras. Consider a graph,  $G$ , with  $n$  vertices (set  $V$ ), and its weighted adjacency matrix  $A$ . Given a parameter  $\alpha > 0$ , let us define the following parametrized quadratic program:

$$\begin{aligned} & \text{maximize} && f_Q^\alpha(\mathbf{x}) = \mathbf{x}^\top (A - \alpha I_Q) \mathbf{x}, \\ & \text{subject to} && \mathbf{x} \in \Delta, \end{aligned} \quad (1)$$

where  $\Delta = \{\mathbf{x} \in \mathbb{R}^n : \sum_i x_i = 1, \text{ and } x_i \geq 0 \text{ for all } i = 1 \dots n\}$ ,  $\mathbf{x}$  contains a membership score for each node and  $I_Q$  is the  $n \times n$  diagonal matrix whose diagonal elements are set to 1 in correspondence to the vertices contained in  $V \setminus Q$  (a set  $V$  without the element  $Q$ ) and to zero otherwise.

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

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

$$\alpha > \lambda_{\max}(A_{V \setminus Q}), \quad (2)$$

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

#### 3.2 Within-Camera Tracking

Figure 2 shows proposed within-camera tracking framework. First, we divide a video into multiple short segments, each segment contains 15 frames, and generate short-tracklets, where human detection bounding boxes in two consecutive frames with 70% overlap, are connected [7]. Then, short-tracklets from 10 different non-overlapping segments are used as input to our first layer of tracking. Here the nodes are short-tracklets (Figure 2, bottom left). 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, bottom right). In the second layer, tracklets of the same person from different segment are associated forming tracks of a person within a camera.

##### 3.2.1 Formulation Using 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, bottom left) and tracklet ( $t_k^l$ , that is,  $l^{th}$  tracklet of camera  $k$ ), in the second layer (Figure 2, bottom right). The corresponding affinity matrix  $A = \{a_{i,j}\}$ , 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

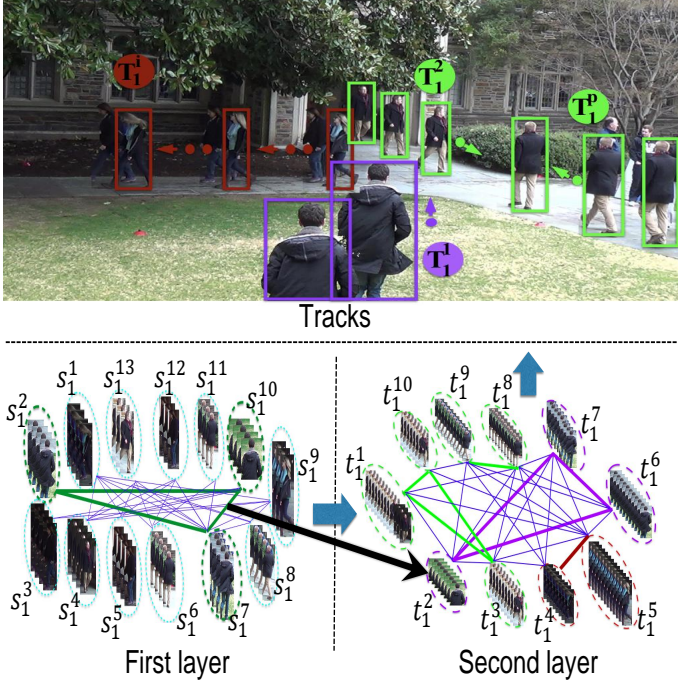


Fig. 2: The figure shows within-camera tracking where short-tracklets from different segments are used as input to our first layer of tracking. The resulting tracklets from the first layer are inputs to the second layer, which determine a tracks for each person. The three dark green short-tracklets ( $s_1^2, s_1^{10}, s_1^7$ ), shown by dotted ellipse in the first layer, form a cluster resulting in 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 cliques (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)

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) [44] fine-tuned using the trainval sequence of DukeMTMC dataset. Similar to [6], we employ a global constant velocity model to compute motion similarity between two nodes.

**Determining cliques:** In our formulation, a clique of 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, using game dynamics, by utilizing intrinsic properties of constrained dominant sets. Note that we do not use peel-off strategy to remove the nodes of found cliques from the graph, this keeps the scale of our problem (number of nodes in a graph) which guarantees that all the found local solutions are the local solutions of the (original) graph. After the extraction of each cluster, the constraint set is changed in such a way to make the extracted cluster unstable under the dynamics. The within-camera tracking starts with all nodes as constraint set. Let us say  $\Gamma^i$  is the  $i^{th}$  extracted cluster,  $\Gamma^1$  is then the first extracted cluster which contains a subset of elements from the whole

set. After our first extraction, we change the constraint set to a set  $V \setminus \Gamma^1$ , hence rendering its associated nodes unstable (making the dynamics not able to select sets of nodes in the interior of associated nodes). The procedure iterates, updating the constraint set at the  $i^{th}$  extraction as  $V \setminus \bigcup_{l=1}^i \Gamma^l$ , until the constraint set becomes empty. Since we are not removing the nodes of the graph (after each extraction of a compact set), we may end up with a solution that assigns a node to more than one cluster.

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

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

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

Algorithm (1), putting the number of cameras under consideration ( $\mathcal{I}$ ) to 1 and  $\mathcal{Q}$  as short-tracklets(tracklets) in the first(second) layer, is used to determine constrained dominant sets which correspond to tracklet(track) in the first (second) layer.

### 3.3 Across-Camera Tracking

#### 3.3.1 Graph Representation of Tracks and the Payoff Function

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

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

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

Figure 3 shows exemplar graph for across-camera tracking among three cameras.  $T_j^i$  represents the  $i^{th}$  track of camera  $j$ . Black and orange edges, respectively, represent within- and across-camera relations of the tracks. From the affinity  $A$ ,  $A^{i \times j}$  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.

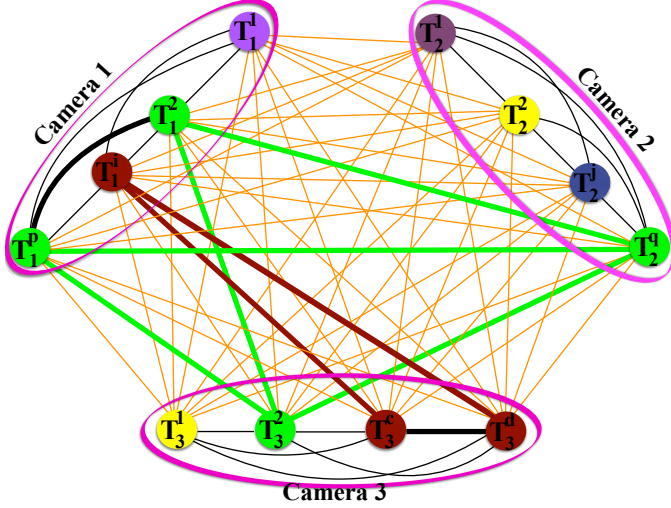


Fig. 3: Exemplar graph of tracks from three cameras.  $T_j^i$  represents the  $i^{\text{th}}$  track of camera  $j$ . Black and colored edges, respectively, represent within- and across-camera relations of tracks. Colours of the nodes depict track IDs, nodes with similar colour represent tracks of the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

### 3.3.2 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. A person, represented by the green node from our exemplar graph (Figure 3), has two tracks which are difficult to merge during within-camera tracking; however, they belong to clique (or cluster) with tracks in camera 2 and camera 3, 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), our constraint set  $\mathcal{Q}$  contains nodes of tracks of camera 1,  $\mathcal{Q} = \{T_1^1, T_1^2, T_1^q, T_1^p\}$ .  $I_{\mathcal{Q}}$  is then  $n \times n$  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 sub-matrix  $I_{\mathcal{Q}}$ , 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 guarantee that the maximizer of program in eq. (1) contains some elements from set  $\mathcal{Q}$ : i.e.,  $C_1^1 = \{T_1^2, T_1^p, T_1^q, T_1^2\}$  forms a clique 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,  $C_1^2$ , contains tracks shown with the dark red color, which illustrates the case where within- and across-camera tracks are in one clique. Lastly, the  $C_1^3 = T_1^1$  represents a track of a person that appears only in camera 1. As a

general case,  $C_j^i$ , represents the  $i^{\text{th}}$  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.

Though Algorithm (1) 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.  $T_p^{\theta}$  typifies the  $\theta^{\text{th}}$  track from camera  $p$  and  $T_p$  contains all the tracks in camera  $p$ . The function  $\mathcal{F}(\mathcal{Q}, A)$  takes as an input a constraint set  $\mathcal{Q}$  and the affinity  $A$ , 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  $\Delta$ , e.g. using game dynamics, and then changing the constraint set  $\mathcal{Q}$ , until all members of the constraint set have been clustered.

---

#### Algorithm 1: Track Association

---

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

---

### 3.4 Track Refinement

The proposed framework, together with the notion of centrality of constrained dominant sets and the notion of reciprocal neighbours, helps us in refining tracking results using tracks from different cameras as different constraint sets. 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  $p^{\text{th}}$  camera.  $M_p^{i^k}$  is the membership score assigned to the  $l^{\text{th}}$  track in the set  $C_p^i$ .

We use two constraints during track refinement stage, which helps us refining false positive association.

**Constraint-1:** A track can not be found in two different sets generated using 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  $l^{\text{th}}$  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:

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

where  $i = 1, \dots, |\mathcal{K}_p^l|$ . 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 the number of cameras under consideration.* If we consider  $\mathcal{I}$  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, i.e. by enforcing the reciprocal properties of the sets.

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 a track  $\vartheta$  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  $\vartheta$  considering tracks from camera  $q$  as constraint set is chosen as:

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

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

#### 4 FAST APPROACH FOR SOLVING CONSTRAINED DOMINANT SET CLUSTERING

Our constrained quadratic optimization program can be solved using dynamics from evolutionary game theory. The well-known standard game dynamics to equilibrium selection, replicator dynamics, though efficient, poses serious efficiency problems, since the time complexity for each iteration of the replicator dynamics is  $\mathcal{O}(n^2)$ , which makes it not efficient for large scale data sets [43]. Rota Bulò *et al.* [19] proposed a new class of evolutionary game dynamics, called Infection and Immunization Dynamics (InflmDyn). InflmDyn solves the problem in linear time. However, it needs the whole affinity matrix to extract a dominant set which, more often than not, exists in local range of the whole graph. Dominant Set Clustering (DSC) [17] is an iterative method which, at each iteration, peels off a cluster by performing a replicator dynamics until its convergence. Efficient out-of-sample [45], extension of dominant sets, is the other approach which is used to reduce the computational cost by sampling the nodes of the graph using some given sampling rate that affects the framework efficacy. Liu *et al.* [46] proposed an iterative clustering algorithm, which operates in two steps: Shrink and Expansion. These steps help reduce the runtime of replicator dynamics on the whole data, which might be slow. The approach has many limitations such as its preference of sparse graph with many small clusters and the results are sensitive to

some additional parameters. Another approach which tries to reduce the computational complexity of the standard quadratic program (StQP [47]) is proposed by [48].

All the above formulations, with their limitations, try to minimize the computational complexity of StQP using the standard game dynamics, whose complexity is  $\mathcal{O}(n^2)$  for each iteration.

In this work we propose a fast approach (listed in Algorithm 2), based on InflmDyn approach which solves StQP in  $\mathcal{O}(n)$ , for the recently proposed formulation,  $\mathbf{x}^\top (\mathbf{A} - \alpha \mathbf{I}_Q) \mathbf{x}$ , which of-course generalizes the StQP.

InflmDyn is a game dynamics inspired by Evolutionary game theory. The dynamics extracts a dominant set using a two-steps approach (infection and immunization), that iteratively increases the compactness measure of the objective function by driving the (probability) distribution with lower payoff to extinction, by determining an ineffective distribution  $\mathbf{y} \in \Delta$ , that satisfies the inequality  $(\mathbf{y} - \mathbf{x})^\top \mathbf{A} \mathbf{x} > 0$ , the dynamics combines linearly the two distributions ( $\mathbf{x}$  and  $\mathbf{y}$ ), thereby engendering a new population  $\mathbf{z}$  which is immune to  $\mathbf{y}$  and guarantees a maximum increase in the expected payoff. In our setting, given a set of instances (tracks, tracklets) and their affinity, we first assign all of them an equal probability (a distribution at the centre of the simplex, a.k.a. barycenter). The dynamics then drives the initial distribution with lower affinity to extinction; those which have higher affinity start getting higher, while the other get lower values. A selective function,  $\mathcal{S}(\mathbf{x})$ , is then run to check if there is any infective distribution; a distribution which contains instances with a better association score. By iterating this process of infection and immunization the dynamics is said to reach the equilibrium, when the population is driven to a state that cannot be infected by any other distribution, that is there is no distribution, whose support contains a set of instances with a better association score. The selective function, however, needs whole affinity matrix, which makes the InflmDyn inefficient for large graphs. 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 [43], the algorithm computes the eigenvalue of the submatrix for every extraction of the compact sets, 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 running the dynamics over the whole graph, we localize it on the sub-matrix, 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  $\alpha$  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  $r$ , which is much smaller than the original affinity, is the size of the sub-matrix where the dynamics is run.

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

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

For a distribution  $x \in \Delta$  to be a KKT-point, in order to satisfy the first-order necessary conditions for local optimality [49], it should satisfy the following two conditions:

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

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

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

Since both the  $x_i$  and the  $\mu_i$  values 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:

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

We then 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

$$\left\{ \sum_{i,j=1}^n x_i y_j a_{ij} - \alpha x_i y_j \right\} > \left\{ \sum_{i,j=1}^n x_i x_j a_{ij} - \alpha x_i x_j \right\} \quad (4)$$

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

**proposition 1.** Given an affinity  $A$  and a distribution  $\mathbf{x} \in \Delta$ , if  $(A\mathbf{x})_i > \mathbf{x}' A \mathbf{x} - \alpha \mathbf{x}'_{\mathcal{Q}} \mathbf{x}_{\mathcal{Q}}$ , 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  $\mathbf{x}$

(We refer the reader to appendix for the proof)

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

Let a function,  $\mathcal{S}(A, x)$ , returns a dominant distribution for distribution,  $x$ ,  $\emptyset$  otherwise and  $\mathcal{G}(A, \mathcal{Q}, x)$  returns the local maximizer of program (1). We summarize the details of our proposed algorithm in Algorithm (2)

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

## 5 EXPERIMENTAL RESULTS

The proposed framework has been evaluated on recently-released large dataset, MOTchallenge DukeMTMC [32], [50], [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 challenging video-based person re-identification dataset, to show that the proposed cross-camera tracking approach can efficiently solve this task also.

**DukeMTMC** is recently-released dataset to evaluate the performance of multi-target multi-camera tracking systems.

---

### Algorithm 2: Fast CDSC

---

**INPUT:** Affinity  $B$ , 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**  
     $\mathbf{x}_d \leftarrow \mathcal{S}(B, \mathbf{x})$ , Find dominant distribution for  $x$  ;  
    if  $\mathbf{x}_d = \emptyset$  break ;  
     $\mathcal{H} \leftarrow \sigma(\mathbf{x}_d) \cup \mathcal{Q}$ , subgraph nodes;  
     $A \leftarrow B_{\mathcal{H}}$ ;  
     $\mathbf{x}_l \leftarrow \mathcal{G}(A, \mathcal{Q}, x)$ ;  
     $\mathbf{x} \leftarrow \mathbf{x} * 0$ ;  
     $\mathbf{x}(\mathcal{H}) \leftarrow \mathbf{x}_l$ ;  
**end**  
**OUTPUT:**  $\{\mathbf{x}\}$

---

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 0 to 54 people. The video is 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 more 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), 891 people walking in front of only one camera.

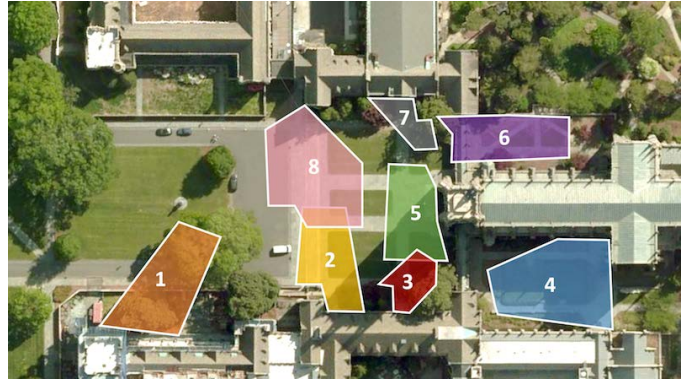


Fig. 4: Camera topology for DukeMTMC dataset. Detections from the overlapping fields of view are not considered. More specifically, intersection occurred between camera (8 & 2) and camera (5 & 3).

**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) [52] and GMMCP tracker [7] were used to automatically generate the



tracklets (mostly 25-50 frames long). Since the video and the detections are not available we use 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 [32]: Identification F-measure (IDF1), Identification Precision (IDP) and Identification Recall (IDR) [32]. The standard performance measures such as CLEAR MOT report the amount of incorrect decisions made by a tracker. Ristani *et al.* [32] 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-Score 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 [32], we use them for the evaluation of our single camera tracking results.

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  $A$  adapting kernel trick distance function from [53], is constructed as follows:

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

where  $K(x_i, x_j)$  is chosen as the Laplacian kernel

$$\exp(-\gamma \|x_i - x_j\|_1).$$

The kernel parameter  $\gamma$  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 things 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 [54], i.e if a track in camera  $i$  and a track in camera  $j$  have a probability to form a trajectory, and track  $j$  in turn have

linkage possibility 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 patch, of the given track, extracted from the last fully-connected layer of Imagenet pre-trained 50-layers Residual Network (ResNet\_50) [44], 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) [38] is then used for pairwise distance computation between instances. For the experiments on MARS, patch representation is obtained using CNN features used in [51]. The pairwise distances between instances are then computed in XQDA, KISSME [55] and euclidean spaces.

## 5.1 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. The symbol  $\uparrow$  means higher scores indicate better performance, while  $\downarrow$  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 7% in IDF1 metrics, while in that of the Test-Hard sequence, our method produces 5% larger average MOTA score than [32], and 1% improvement is achieved in IDF1. Table 3 and Table 4 respectively present Multi-Camera performance of our and state-of-the-art approach [32] on the Test-Easy and Test-Hard sequence (respectively) of DukeMTMC dataset. We have improved IDF1 for both Test-Easy and Test-Hard sequences by 4% and 3%, respectively.

Figure 8 depicts 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 the first row, all the four targets, even under significant illumination and pose changes, are successfully tracked in four cameras, where they appear. In the second row, target 714 is successfully tracked through three cameras. Observe its significant illumination and pose changes from camera 5 to camera 7. In the third row, targets that move through camera 1, target six, seven and eight are tracked. The last row shows tracks of targets that appear in cameras 1 to 4.

	Methods	MOTA $\uparrow$	MOTP $\uparrow$	FAF $\downarrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDS $\downarrow$	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$
Camera1	[32]	43.0	79.0	0.03	24	46	2,713	107,178	39	57.3	91.2	41.8
	Ours	69.9	76.3	0.06	137	22	5,809	52,152	156	76.9	89.1	67.7
Camera2	[32]	44.8	78.2	0.51	133	8	47,919	53,74	60	68.2	69.3	67.1
	Ours	71.5	74.6	0.09	134	21	8,487	43,912	75	81.2	90.9	73.4
Camera3	[32]	57.8	77.5	0.02	52	22	1,438	28,692	16	60.3	78.9	48.8
	Ours	67.4	75.6	0.02	44	9	2,148	21,125	38	64.6	76.3	56.0
Camera4	[32]	63.2	80.2	0.02	36	18	2,209	19,323	7	73.5	88.7	62.8
	Ours	76.8	76.6	0.03	45	4	2,860	10,689	18	84.7	91.2	79.0
Camera5	[32]	72.8	80.4	0.05	107	17	4,464	35,861	54	73.2	83.0	65.4
	Ours	68.9	77.4	0.10	88	11	9,117	36,933	139	68.3	76.1	61.9
Camera6	[32]	73.4	80.2	0.06	142	27	5,279	45,170	55	77.2	87.5	69.1
	Ours	77.0	77.2	0.05	136	11	4,868	38,611	142	82.7	91.6	75.3
Camera7	[32]	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	[32]	60.7	76.7	0.03	102	53	2,730	52,806	46	72.4	92.2	59.6
	Ours	63.4	73.6	0.04	92	28	4,184	47,565	91	73.0	89.1	61.0
Average	[32]	59.4	78.7	0.09	665	234	68,147	361,672	300	70.1	83.6	60.4
	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 approach [32] on the Test-Easy sequence of DukeMTMC dataset.

	Methods	MOTA $\uparrow$	MOTP $\uparrow$	FAF $\downarrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDS $\downarrow$	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$
Camera1	[32]	37.8	78.1	0.03	6	34	1,257	78,977	55	52.7	92.5	36.8
	Ours	63.2	75.7	0.08	65	17	2,886	44,253	408	67.1	83.0	56.4
Camera2	[32]	47.3	76.5	0.74	68	12	26526	46898	194	60.6	65.7	56.1
	Ours	54.8	73.9	0.24	62	16	8,653	54,252	323	63.4	78.8	53.1
Camera3	[32]	46.7	77.9	0.01	24	4	288	18182	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	[32]	85.3	81.5	0.04	21	0	1,215	2,073	1	84.3	86.0	82.7
	Ours	75.6	77.7	0.05	17	0	1,571	3,888	61	82.3	87.1	78.1
Camera5	[32]	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	[32]	59.4	76.7	0.14	85	23	5,156	77,031	225	64.1	81.7	52.7
	Ours	53.3	76.5	0.17	68	36	5,989	88,164	547	53.1	71.2	42.3
Camera7	[32]	50.8	73.3	0.08	43	23	2,971	38,912	148	59.6	81.2	47.1
	Ours	50.8	74.0	0.05	34	20	1,935	39,865	266	60.6	84.7	47.1
Camera8	[32]	73.0	75.9	0.02	34	5	706	9735	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	[32]	54.6	77.1	0.14	338	103	39,599	283,376	652	64.5	81.2	53.5
	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 approach [32] on the Test-Hard sequence of DukeMTMC dataset.

	Methods	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$
Multi-Camera	[32]	56.2	67.0	48.4
	Ours	60.0	68.3	53.5

TABLE 3: Multi-camera performance of our and state-of-the-art approach [32] on the Test-Easy sequence of DukeMTMC dataset.

	Methods	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$
Multi-Camera	[32]	47.3	59.6	39.2
	Ours	50.9	63.2	42.6

TABLE 4: Multi-Camera performance of our and state-of-the-art approach [32] on the Test-Hard sequence of DukeMTMC dataset.

## 5.2 Evaluation on MARS dataset:

In Table 5 we compare our results (using the same settings as in [51]) on MARS dataset with the state-of-the-art methods. The proposed approach achieves 3% 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

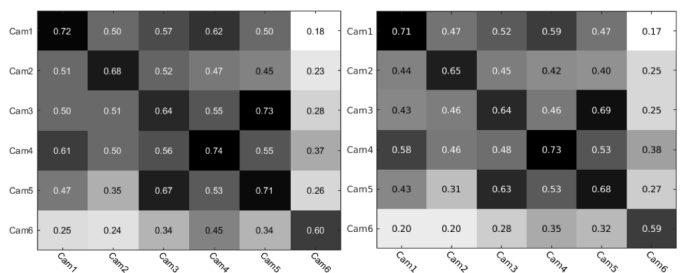


Fig. 5: The results show the performance of our algorithm on MARS (both using CNN + XQDA) when the final ranking is done using membership score (left) and using pairwise euclidean distance (right).

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 constrained dominant set is, we conduct an experiment on the MARS dataset computing the final ranking using the membership score and pairwise distances. The confusion matrix in Fig. 5 shows the detail result of both the within

Methods	rank 1
HLBP + XQDA	18.60
BCov + XQDA	9.20
LOMO + XQDA	30.70
BoW + KISSME	30.60
SDALF + DVR	4.10
HOG3D + KISSME	2.60
CNN + XQDA [51]	65.30
CNN + KISSME [51]	65.00
Ours	<b>68.22</b>

TABLE 5: The table shows the comparison (based on rank-1 accuracy) of our approach with the state-of-the-art approaches: SDALF [39], HLBP [40], BoW [41], BCov [42], LOMO [38], HOG3D [56] on MARS dataset.

Feature+Distance	Methods	Within	Across
CNN + Eucl	[51]	0.59	0.28
	Ours (PairwiseDist)	0.59	0.29
	Ours (MembershipS)	<b>0.60</b>	<b>0.29</b>
CNN + KISSME	[51]	0.61	0.34
	Ours (PairwiseDist)	0.64	0.41
	Ours (MembershipS)	<b>0.67</b>	<b>0.44</b>
CNN + XQDA	[51]	0.62	0.35
	Ours (PairwiseDist)	0.65	0.42
	Ours (MembershipS)	<b>0.68</b>	<b>0.45</b>

TABLE 6: The results show performance of our(using pairwise distance and membership score) 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.

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 constrained dominant set framework. Note that 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.

### 5.3 Computational Time.

Figure 6 shows the time taken for each track - from 100 randomly selected (query) tracks - to be associated, with the rest of the (gallery) tracks, running CDSC over the whole graph (CDSC without speedup) and running it on a small portion of the graph using the proposed approach (called FCDSC, CDSC with speedup). 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.

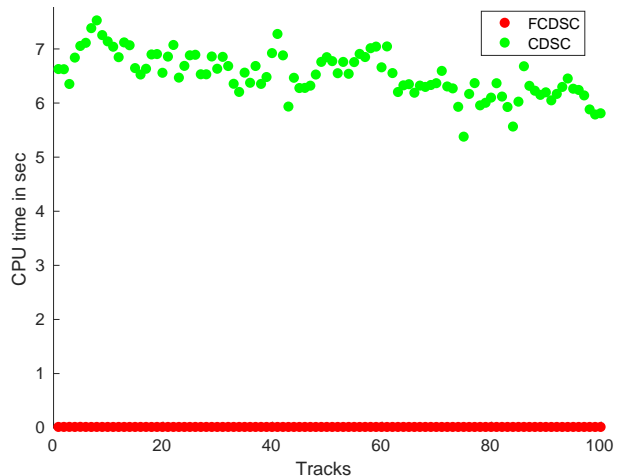


Fig. 6: CPU time taken for each track association using our proposed fast approach (FCDSC - fast CDSC) and CDSC.

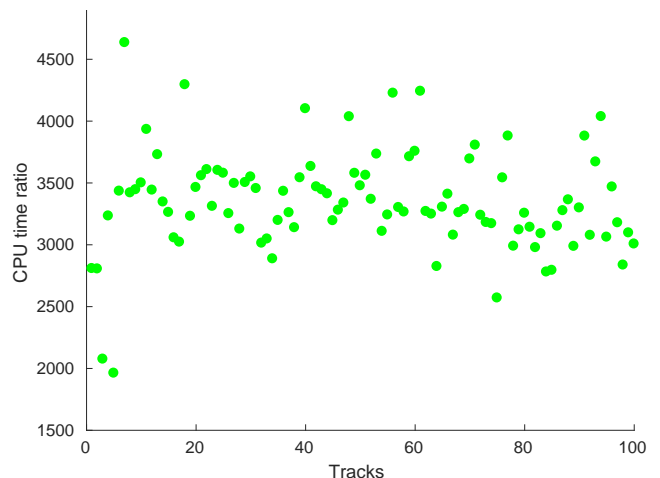


Fig. 7: The ratio of CPU time taken between CDSC and proposed fast approach (FCDSC), computed as CPU time for CDSC/CPU time for FCDSC.

## 6 CONCLUSIONS

In this paper we presented a constrained dominant set clustering (CDSC) based framework for solving multi-target tracking problem in multiple non-overlapping cameras. 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 effectiveness of our model.

We further perform additional experiments to show effectiveness of the proposed across-camera tracking on one of the largest video-based people re-identification datasets (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

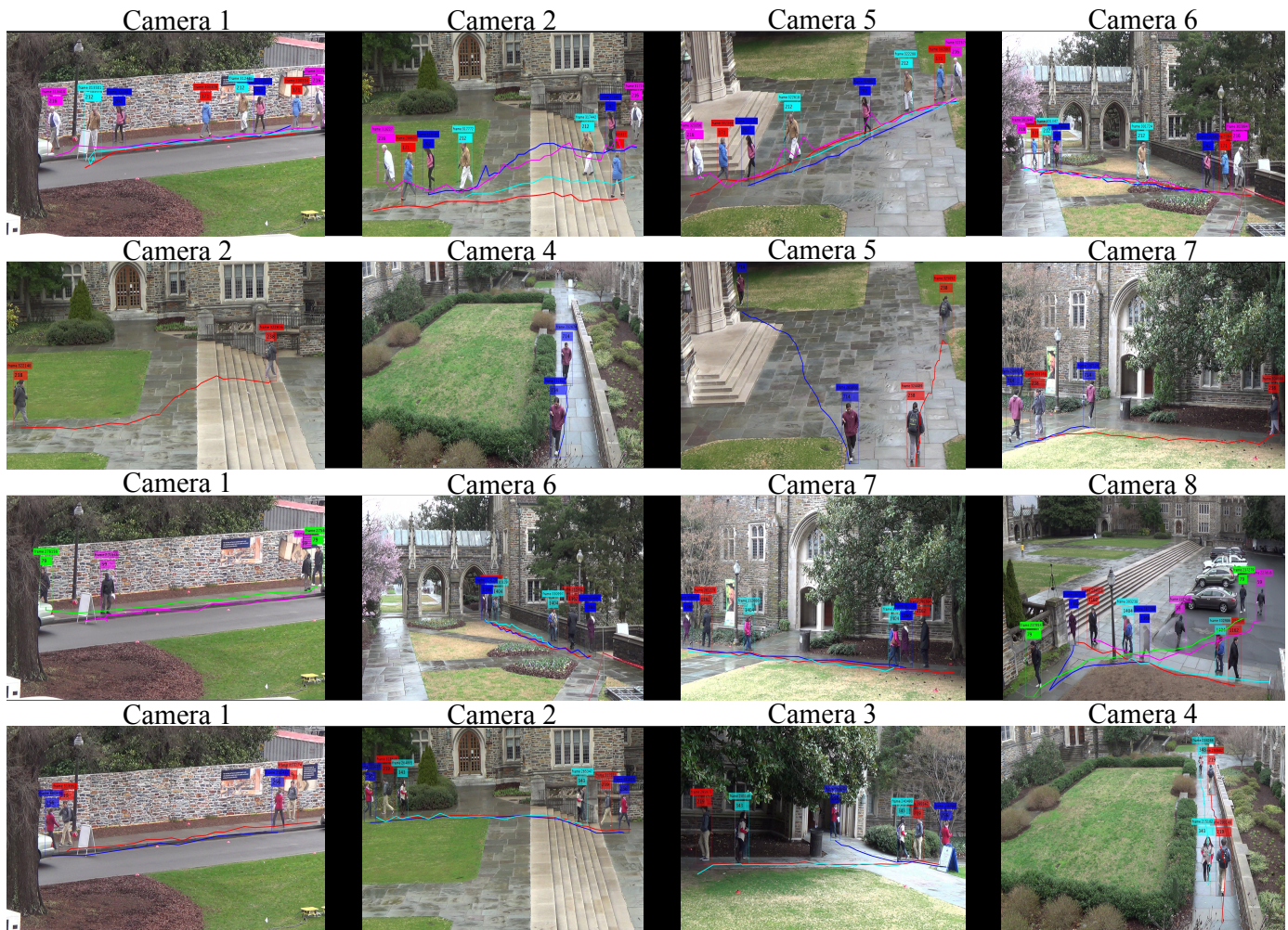


Fig. 8: Sample qualitative results of the proposed approach on DukeMTMC dataset. Bounding boxes and lines with the same color indicate the same target (Best viewed in color).

matches to their corresponding query.

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

## REFERENCES

- [1] Y. Cai, K. Huang, and T. Tan, "Human appearance matching across multiple non-overlapping cameras," in *International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.
- [2] A. Chilgunde, P. Kumar, S. Ranganath, and W. Huang, "Multi-camera target tracking in blind regions of cameras with non-overlapping fields of view," in *British Machine Vision Conference (BMVC)*, 2004, pp. 1–10.
- [3] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *International Conference on Computer Vision (ICCV)*, 2003, pp. 952–957.
- [4] Y. Wang, S. Velipasalar, and M. C. Gursoy, "Distributed wide-area multi-object tracking with non-overlapping camera views," *Multimedia Tools and Applications*, vol. 73, no. 1, pp. 7–39, 2014.
- [5] X. Chen and B. Bhanu, "Integrating social grouping for multi-target tracking across cameras in a crf model," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, in press.
- [6] A. R. Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 343–356.
- [7] A. Dehghan, S. M. Assari, and M. 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.
- [8] Y. T. Tesfaye, E. Zemene, M. Pelillo, and A. Prati, "Multi-object tracking using dominant sets," *IET computer vision*, vol. 10, pp. 289–298, 2016.
- [9] N. Anjum and A. Cavallaro, "Trajectory association and fusion across partially overlapping cameras," in *IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, 2009, pp. 201–206.
- [10] S. Calderara, R. Cucchiara, and A. Prati, "Bayesian-competitive consistent labeling for people surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 2, 2008.
- [11] C. R. del-Blanco, R. Moledano, N. N. García, L. Salgado, and F. Jaureguizar, "Color-based 3d particle filtering for robust tracking in heterogeneous environments," in *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2008, pp. 1–10.
- [12] S. Khan and M. Shah, "Consistent labeling of tracked objects in

- multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, no. 10, pp. 1355–1360, 2003.
- [13] B. Möller, T. Plötz, and G. A. Fink, "Calibration-free camera hand-over for fast and reliable person tracking in multi-camera setups," in *International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.
- [14] S. Velipasalar, J. Schlessman, C. Chen, W. H. Wolf, and J. Singh, "A scalable clustered camera system for multiple object tracking," *EURASIP J. Image and Video Processing*, vol. 2008, 2008.
- [15] H. W. Kuhn, "Variants of the hungarian method for assignment problems," *Naval Research Logistics Quarterly*, vol. 3, no. 4, pp. 253–258, 1956.
- [16] E. Zemene, Y. Tariku, H. Idrees, A. Prati, M. Pelillo, and M. Shah, "Large-scale image geo-localization using dominant sets," *CoRR*, vol. abs/1702.01238, 2017.
- [17] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 1, pp. 167–172, 2007.
- [18] E. Zemene, Y. Tariku, A. Prati, and M. Pelillo, "Simultaneous clustering and outlier detection using dominant sets," in *ICPR*, 2016.
- [19] S. Rota Bulò, M. Pelillo, and I. M. Bomze, "Graph-based quadratic optimization: A fast evolutionary approach," *Computer Vision and Image Understanding*, vol. 115, no. 7, pp. 984–995, 2011.
- [20] E. Z. Mequanint, S. Rota Bulò, and M. Pelillo, "Dominant-set clustering using multiple affinity matrices," in *SIMBAD*, 2015, pp. 186–198.
- [21] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, "Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views," *Computer Vision and Image Understanding*, vol. 109, no. 2, pp. 146–162, 2008.
- [22] A. Gilbert and R. Bowden, "Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 125–136.
- [23] B. J. Prosser, S. Gong, and T. Xiang, "Multi-camera matching using bi-directional cumulative brightness transfer functions," in *British Machine Vision Conference (BMVC)*, 2008, pp. 1–10.
- [24] T. D'Orazio, P. L. Mazzeo, and P. 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.
- [25] S. Srivastava, K. K. Ng, and E. 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.
- [26] 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, 2017.
- [27] C. Kuo, C. Huang, and R. 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.
- [28] Y. Gao, R. Ji, L. Zhang, and A. G. Hauptmann, "Symbiotic tracker ensemble toward a unified tracking framework," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 24, no. 7, pp. 1122–1131, 2014.
- [29] S. Zhang, Y. Zhu, and A. K. Roy-Chowdhury, "Tracking multiple interacting targets in a camera network," *Computer Vision and Image Understanding*, vol. 134, pp. 64–73, 2015.
- [30] Y. Cai and G. G. Medioni, "Exploring context information for inter-camera multiple target tracking," in *IEEE Workshop on Applications of Computer Vision (WACV)*, 2014, pp. 761–768.
- [31] X. Chen, K. Huang, and T. Tan, "Object tracking across non-overlapping views by learning inter-camera transfer models," *Pattern Recognition*, vol. 47, no. 3, pp. 1126–1137, 2014.
- [32] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 17–35.
- [33] E. Ristani and C. Tomasi, "Tracking multiple people online and in real time," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 444–459.
- [34] J. You, A. Wu, X. Li, and W. Zheng, "Top-push video-based person re-identification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1345–1353.
- [35] N. McLaughlin, J. M. del Rincón, and P. 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.
- [36] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 688–703.
- [37] D. N. T. Cong, C. Achard, L. Khoudour, and L. Douadi, "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.
- [38] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "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.
- [39] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2360–2367.
- [40] F. Xiong, M. Gou, O. I. Camps, and M. Szaier, "Person re-identification using kernel-based metric learning methods," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 1–16.
- [41] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124.
- [42] B. Ma, Y. Su, and F. Jurie, "Covariance descriptor based on bio-inspired features for person re-identification and face verification," *Image Vision Computing*, vol. 32, no. 6-7, pp. 379–390, 2014.
- [43] E. Zemene and M. Pelillo, "Interactive image segmentation using constrained dominant sets," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 278–294.
- [44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [45] M. Pavan and M. Pelillo, "Efficient out-of-sample extension of dominant-set clusters," in *Annual Conference on Neural Information Processing Systems (NIPS)*, 2004, pp. 1057–1064.
- [46] H. Liu, L. J. Latecki, and S. Yan, "Fast detection of dense subgraphs with iterative shrinking and expansion," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 9, pp. 2131–2142, 2013.
- [47] I. M. Bomze, "Branch-and-bound approaches to standard quadratic optimization problems," *J. Global Optimization*, vol. 22, no. 1-4, pp. 17–37, 2002.
- [48] L. Chu, S. Wang, S. Liu, Q. Huang, and J. Pei, "ALID: scalable dominant cluster detection," *Conference on Very Large DataBases (VLDB)*, vol. 8, no. 8, pp. 826–837, 2015.
- [49] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 2008, vol. 3.
- [50] F. Solera, S. Calderara, E. Ristani, C. Tomasi, and R. Cucchiara, "Tracking social groups within and across cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.
- [51] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "MARS: A video benchmark for large-scale person re-identification," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 868–884.
- [52] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [53] R. Grossman, R. J. Bayardo, and K. P. Bennett, Eds., *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2005.
- [54] E. Zemene and M. Pelillo, "Path-based dominant-set clustering," in *ICIAI*, 2015, pp. 150–160.
- [55] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2288–2295.
- [56] A. Kläser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference (BMVC)*, 2008, pp. 1–10.



**Yonatan Tariku Tesfaye** received his BSc degree in computer science from Arba Minch University in 2007. He has worked 5 years at Ethio-telecom as senior programmer and later joined CaFoscari University of Venice and received his MSc degree (with Honor) in computer science in 2014. He is currently a PhD student at IUAV university of Venice starting from 2014. He is now a research assistance, towards his PhD, at Center for Research in Computer Vision at University of Central Florida. His research interests include

multi-target tracking, people re-identification, segmentation, image geolocalization, game theoretic model and graph theory.



**Eyasu Zemene** received the BSc degree in Electrical Engineering from Jimma University in 2007, he then worked at Ethio Telecom for 4 years till he joined CaFoscari University (October 2011) where he got his MSc in Computer Science in June 2013. September 2013, he won a 1 year research fellow to work on Adversarial Learning at Pattern Recognition and Application lab of University of Cagliari. Since September 2014 he is a PhD student of CaFoscari University under the supervision of prof. Pelillo. Work-

ing towards his Ph.D. he is trying to solve different computer vision and pattern recognition problems using theories and mathematical tools inherited from graph theory, optimization theory and game theory. Currently, Eyasu is working as a research assistant at Center for Research in Computer Vision at University of Central Florida under the supervision of Dr. Mubarak Shah. His research interests are in the areas of Computer Vision, Pattern Recognition, Machine Learning, Graph theory and Game theory.



**Andrea Prati** Andrea Prati graduated in Computer Engineering at the University of Modena and Reggio Emilia in 1998. He got his PhD in Information Engineering in 2002 from the same University. After some post-doc position at University of Modena and Reggio Emilia, he was appointed as Assistant Professor at the Faculty of Engineering of Reggio Emilia (University of Modena and Reggio Emilia) from 2005 to 2011, and then as Associate Professor at the Department of Design and Planning in Complex

Environments of the University IUAV of Venice, Italy. In 2013 he has been promoted to full professorship, waiting for official hiring in the new position. In December 2015 he moved to the Department of Engineering and Architecture of the University of Parma. Author of 7 book chapters, 31 papers in international referred journals (including 9 papers published in IEEE Transactions) and more than 100 papers in proceedings of international conferences and workshops. Andrea Prati is Senior Member of IEEE, Fellow of IAPR ("For contributions to low- and high-level algorithms for video surveillance"), and member of GIRPR.



**Marcello Pelillo** is Full Professor of Computer Science at CaFoscari University in Venice, Italy, where he directs the European Centre for Living Technology (ECLT) and leads the Computer Vision and Pattern Recognition group. He held visiting research positions at Yale University, McGill University, the University of Vienna, York University (UK), the University College London, and the National ICT Australia (NICTA). He has published more than 200 technical papers in refereed journals, handbooks, and conference

proceedings in the areas of pattern recognition, computer vision and machine learning. He is General Chair for ICCV 2017 and has served as Program Chair for several conferences and workshops (EMMCVPR, SIMBAD, S+SSPR, etc.). He serves (has served) on the Editorial Boards of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Pattern Recognition, IET Computer Vision, Frontiers in Computer Image Analysis, Brain Informatics, and serves on the Advisory Board of the International Journal of Machine Learning and Cybernetics. Prof. Pelillo is a Fellow of the IEEE and a Fellow of the IAPR.



**Mubarak Shah**, the Trustee chair professor of computer science, is the founding director of the Center for Research in Computer Vision at the University of Central Florida (UCF). He is an editor of an international book series on video computing, editor-in-chief of Machine Vision and Applications journal, and an associate editor of ACM Computing Surveys journal. He was the program cochair of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2008, an associate editor of the IEEE

Transactions on Pattern Analysis and Machine Intelligence, and a guest editor of the special issue of the International Journal of Computer Vision on Video Computing. His research interests include video surveillance, visual tracking, human activity recognition, visual analysis of crowded scenes, video registration, UAV video analysis, and so on. He is an ACM distinguished speaker. He was an IEEE distinguished visitor speaker for 1997-2000 and received the IEEE Outstanding Engineering Educator Award in 1997. In 2006, he was awarded a Pegasus Professor Award, the highest award at UCF. He received the Harris Corporations Engineering Achievement Award in 1999, TOKTEN awards from UNDP in 1995, 1997, and 2000, Teaching Incentive Program Award in 1995 and 2003, Research Incentive Award in 2003 and 2009, Millionaires Club Awards in 2005 and 2006, University Distinguished Researcher Award in 2007, Honorable mention for the ICCV 2005 Where Am I? Challenge Problem, and was nominated for the Best Paper Award at the ACM Multimedia Conference in 2005. He is a fellow of the IEEE, AAAS, IAPR, and SPIE.

## APPENDIX

**proposition 2.** Given an affinity  $A$  and a distribution  $\mathbf{x} \in \Delta$ , if  $(A\mathbf{x})_i > \mathbf{x}'A\mathbf{x} - \alpha\mathbf{x}'_Q\mathbf{x}_Q$ , 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 the first condition holds: Let's assume  $\mathbf{x}$  is a KKT point

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

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

$$\mathbf{x}^\top (A - \alpha I_Q)\mathbf{x} = \sum_{i=1}^n x_i * \lambda/2 = \lambda/2$$

From the second condition, we have:

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

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

$$(A\mathbf{x})_i \leq \mathbf{x}^\top (A - \alpha I_Q)\mathbf{x}$$

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

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

$$\{e_i^\top (A - \alpha I_Q)\mathbf{x}\} > \{\mathbf{x}^\top (A - \alpha I_Q)\mathbf{x}\}$$

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

$$(Ax)_i > \{\mathbf{x}^\top (A - \alpha I_Q)\mathbf{x}\}$$

Which concludes the proof

□