16/07/2024 13:52

(Article begins on next page)

Tracking Social Groups Within and Across Cameras

Francesco Solera, Simone Calderara, Ergys Ristani, Carlo Tomasi, Rita Cucchiara

Abstract—We propose a method for tracking groups from single and multiple cameras with disjoint fields of view. Our formulation follows the tracking-by-detection paradigm where groups are the atomic entities and are linked over time to form long and consistent trajectories. To this end, we formulate the problem as a supervised clustering problem where a Structural SVM classifier learns a similarity measure appropriate for group entities. Multi-camera group tracking is handled inside the framework by adopting an orthogonal feature encoding that allows the classifier to learn inter- and intra-camera feature weights differently. Experiments were carried out on a novel annotated group tracking data set, the `DukeMTMC-Groups` data set. Since this is the first data set on the problem it comes with the proposal of a suitable evaluation measure. Results of adopting learning for the task are encouraging, scoring a $+15\%$ improvement in F_1 measure over a non-learning based clustering baseline. To our knowledge this is the first proposal of this kind dealing with multi-camera group tracking.

Index Terms—groups, crowd, detection, tracking, learning

I. INTRODUCTION

THE fast-growing interest in automated analysis of crowds and social gatherings for surveillance and security applications opens new challenges for the computer vision community as well. Modern sociological and psychological theories converge towards the notion of the crowd *as a set of individuals sharing an emergent collective behavior* [1], [2]. Well documented examples of crowds include the Los Angeles 1992 riots and the stock market crash of 1929 while many others, occurring every day, sideshow the flow of history. Group dynamics drives collective behavior by encouraging people to engage in acts they might otherwise consider unthinkable under typical social circumstances [3]. In contrast, people assemble in so-called temporary gatherings [1] without expressing unusual social structures—but many usual ones [4], [5]. Temporary gatherings capture the more common scenario of people walking in shopping malls, city squares, stations, or airports. Crowds and gatherings are constant features of the social world, and groups have proven to be the constitutional and structural building blocks of events related to them. This observation has led to an increased emphasis on group detection and tracking in the automatic analysis of surveillance video.

Group detection is the task of inferring the social relationships underlying an observed crowd of individuals. Groups are often detected independently in each frame [6], [7], [8], [9], [10], and individuals and groups are then tracked

Fig. 1: An example of groups detected in the four different cameras of the proposed data set `DukeMTMC-Groups`.

jointly online, leveraging the idea that individual tracking can help group tracking and *vice versa* [9]. Alternatively, groups are detected globally in a tracking-by-detection optimization framework [10]. Even so, methods that detect groups at the level of one or a few frames may miss important sociological clues that need a few seconds to unfold. To overcome this shortcoming, and similarly to previous work [11], [12], we recently proposed a group detection algorithm that works over temporal windows [13] to account for sociological as well as physical evidence. Working with temporal windows, on the other hand, introduces a consistency problem from window to window that can be neatly formulated as a tracking problem once we agree on Turner's definition of group [14]: "A social group is defined as two or more people interacting to reach a common goal and perceiving a shared membership, based on both physical and social identity."

For the purpose of tracking, we need to define what it means for a group to have a unique identity and to what extent the group itself can change before its identity changes as well. According to Turner's definition, a group is defined by the interactions among its members. Thus, in the remainder of this paper, we assume that when the set of members changes a new group instance is created. However, if a group re-constitute with its original members, its identity must be kept consistent. In our proposal we started from the group definition of Turner and broadened its meaning considering singletons as groups of cardinality equal to 1.

Starting from groups detected over non-overlapping sliding windows, we refer to the group tracking problem as the problem of recovering extended and consistent trajectories of groups within the same camera or across different cameras with disjoint fields of view. Observations, *i.e.*, groups detected in any time window and in any camera, are associated to identities through *Correlation Clustering* [15], [16] so that all observations regarding the same individuals share the

same group identity linked to the same set of individuals. The clustering procedure is cast within a structured learning framework (*Structural SVM* [17]), with the advantage that the intra- and inter-camera affinity measures employed in clustering are jointly and seamlessly defined by training examples and do not require manual tuning. We validated our solution on both tracking singletons and groups on two popular public single camera sequences `BIWI-eth` and `CBE-student`. Additionally, to assess our algorithm multi-camera capabilities, we manually annotated and make public group and singleton information on a subset of the `DukeMTMC` tracking data set[1]. In this subset, more than 120 groups appear in 4 different cameras across 20 minutes of 1080p video recorded at 30 frames per second. To our knowledge, this is the first multi-camera data set ever with group detection and tracking annotations. A new way to measure tracking performance is also introduced to consistently count association errors within and across cameras.

Experiments proved that performing tracking at group level (where singletons are groups with one member) can boost the tracking performance when dealing with moderately crowded scenarios.

The paper is organized as follow: Sec. II reviews literature on group detection and tracking. In Sec. III the group tracking problem is cast as a supervised Correlation Clustering problem while a feature encoding dealing with multiple camera assignments is presented in Sec. IV. The learning algorithm is described in Sec. V and Sec. VI provides details on the datasets, the adopted features, and quantitative results.

II. RELATED WORK

Group detection and tracking has become a feasible task in computer vision only nowadays, and it presents several open challenges from people detection [18] and people tracking in crowds [19] to trajectory analysis [20].

Sociological concepts such as *F-formations* by Kendon [21] have been exploited as a foundation for a few group detection methods [22]. F-formations can be interpreted as specific positional and orientational patterns that people assume when engaged in a social relationship. However, the theory holds only for stationary groups and is not defined for moving groups, a case which cannot be ignored in real world crowds.

In contrast, motion paths are considered by most current approaches. These can be broadly partitioned into three categories according to the type of available tracklets: group-based, joint individual-group, and individual-based. In *group-based* approaches, groups are considered as atomic entities in the scene and no higher level information can be extracted neatly, typically due to high noise or the high complexity of crowded scenes [23], [24], [25]. *Joint individual-group* approaches combine individuals tracking while tracking groups at a coarser level [8], [26]. Still, in this latter category, groups are identified through the identities of their individually

tracked members, with the notion of a group serving only as a prior on tracking. Finally, *individual-based* tracking algorithms build on individual pedestrian trajectories and no information about groups is used until the whole trajectory is recovered.

Some notable group detection and tracking works follow. Pellegrini *et al.* [27] employ a Conditional Random Field to jointly predict trajectories and estimate group memberships, modeled as latent variables, over a short time window. Yamaguchi *et al.* [28] predict groups by minimizing an energy function that encodes physical condition, personal motivation, and social interactions features. In these formulations, group identities are a covering rather than a partition of the set of pedestrians. Recently, Chang *et al.* [12] proposed a soft segmentation process to partition the crowd by using a weighted graphical model where the pairwise potentials on the edges encode the probability of two instances being in the same group. This model applies only to detection and cannot be extended to tracking.

Surprisingly, most of the individual-based solutions solve the multiple target/people tracking (MTT) problem and *then* build groups on top of the solution. The main drawback of this approach is that current MTT methods degrade in performance when the tracking switches from short term to long term [29]. Consequently, group tracking through individual tracking becomes harder in most cases relative to group tracking *per se*. Zanotto *et al.* [26] proposed one of the few approaches that consider the group tracking problem directly. They exploited a set of infinite-mixture distributions to model proxemic-inspired features in a particle filtering framework. The approach is capable of jointly tracking pedestrians and grouping them, exploiting only frame-wise information. A second work along this line is by Qin and Shelton [30], where tracklets are joint into trajectories only when visually coherent in predicted groups. Mazzon *et al.* [10] extended the delayed Social Force Model for group detection by defining plausible human behaviors for the localization of group formations. Results are measured as the improvement over group detection when temporal consistency is enforced trough tracking. In this vast landscape, the specific problem of tracking groups across multiple cameras is mostly neglected. All the aforementioned methods exploit peculiarities of the grouping phenomena to improve tracking of individuals over a short temporal window. Perhaps, the work most similar to ours is by Zheng *et al.* [31], where groups association across different cameras was used as contextual information to improve re-identification. The focus of the paper is re-identification and not tracking. The method works with query/gallery images and not video sequences.

In our proposal, as in group based approaches, groups are the focus of the analysis and we are committed to consistently track groups across cameras and possibly for a long amount of time. Thus, we propose to trust individual tracking information, *i.e.* people trajectories, only for the time needed to detect groups, and thereby assume that tracker reliability is high enough for short temporal windows [29]. Eventually, once groups are provided, we propose to track them neatly on data from single and multiple cameras by exploiting a clustering approach over long temporal windows.

---

[1]The `DukeMTMC`, comprising 90 minutes of footage from 8 cameras, is a fully annotated people tracking data set that will be released separately.

In the remainder of this section we review previous group detection/tracking datasets and highlight how our dataset could be of benefit to the community. `Friends Meet` [9] is a set of 20 sequences lasting 20 seconds on average, with never more than 2/3 groups in the same sequence. Moreover, in each video the motion of all groups is stable and limited. `CAVIAR`[2] is extremely short with at most 2 groups appearing at the same time. Both previous datasets focus on merging/splitting interactions but not at a crowd level. Besides showcasing very simple scenarios, the `ViCoMo` [32] dataset has also very simple blob like annotations, meaning the information is neither about individuals nor their bounding boxes. All of the aforementioned datasets are single camera. The `VANAHEIM`[3] and `i-Lids`[4] (multi-camera) datasets are either not publicly released or dismissed. Eventually, `stu003` [33] and `eth` [34] sequences are good examples of single camera, mildly crowded sequences with both individual tracking and groups annotation – but are of no help in the multi-camera setting. To this end we create, annotate and release the first multi-camera group dataset, which is bigger than others for number of frames, number of scenarios, number of individuals, number of groups and quality of annotation.

III. METHOD OVERVIEW

Our proposal is grounded in the idea that group tracking has peculiar elements that are distinct from individual tracking. This intuition, combined with the fact that studies on people attending events have underlined that most of the people tend to move in groups [4], testifies that handling groups is a mandatory task when dealing with mildly crowded scenarios.

The looseness of Turner's definition of the concept of "group," which comes from social psychology, extends to the visual context, in which distinct groups and members must be identified from camera-observed features. On one hand, people-tracking methods rely on a two-sided constancy assumption about targets [35]:

- The target appearance representation implies a constancy of some property to transfer from one frame to the next. Without any such constancy assumption tracking cannot work.
- The target motion implies the existence of a constant motion model that can adapt to the target by adjusting a set of parameters.

More precisely, if appearance is constant then an appearance model can be built to associate targets [36], while when motion model is constant then motion prediction and trajectory smoothness gain importance in the target association process [37], [38]. On the other hand, these assumptions may be violated when individual tracking is used to associate group members at individual level. In group tracking, appearance constancy cannot be based on a clear and unique group membership model, as members of the group can change their spatial location inside the group, occlude each other, or split apart (see Fig. 3). Similar considerations hold for motion

Fig. 3: One member is taking a picture of the rest of the group. Here, neither the appearance model of individual members nor their motion holds constant. The first fails because of mutual occlusion of members, the second because of a member's independent motion.

prediction, which is affected by the social dynamics inside the group, as members may wait for each other or adjust their paths depending on the behavior of other members and on group purpose (see Fig. 3). These considerations influence the tracking process leading to a high degree of uncertainty in establishing identities of individual group members. Therefore our approach forces consistency in individual identity only when singletons are present, assigning instead a joint ID to group members.

To address these difficulties, we propose a model and features that are appropriate for group tracking while still considering singletons as a special group case where these discontinuities become less evident. Specifically, we develop a structured learning framework based on a simple set of spatial and visual features. Our framework can learn the reliability of features and handle uncertainties in group association even as we expand the scenario from single camera to multiple cameras with disjoint fields of view. Fig. 2a summarizes our proposed tracking approach, which starts from a set of group observations $D_1, D_2, \ldots$ detected over short temporal windows and independently from each camera, as we proposed in [13]. A set of pairwise features, based on both appearance and scene location and detailed in Sec. VI-A, is computed for every pair of group detections D_i and D_j to form the correlation scores $W(i, j)$ (dashed lines in Fig. 2a). We then obtain the tracking solution through Correlation Clustering (solid lines) based on these scores. To reduce the complexity of the problem, we use a sliding temporal window and solve for associations only for groups D_i from all cameras over a single time window T_k span:

$$X_k = \{D_i | \gamma(D_i) \cap \gamma(T_k) \neq \emptyset\} \quad (1)$$

where $\gamma(D_i)$ and $\gamma(T_k)$ are the sets of frames for the i-th observed group and the k-th window respectively.

The core of our proposal is the definition of the correlation scores $W(i, j)$, which is learnt through a Structural SVM (SSVM) classifier, to optimally solve the problem of group association over a set of overlapping temporal windows $T_1, T_2, \ldots$. The classifier solves the problem globally for all the cameras by treating inter- and intra-camera group associations differently. Specifically, depending on the source and destination cameras for each group observation, the SSVM:

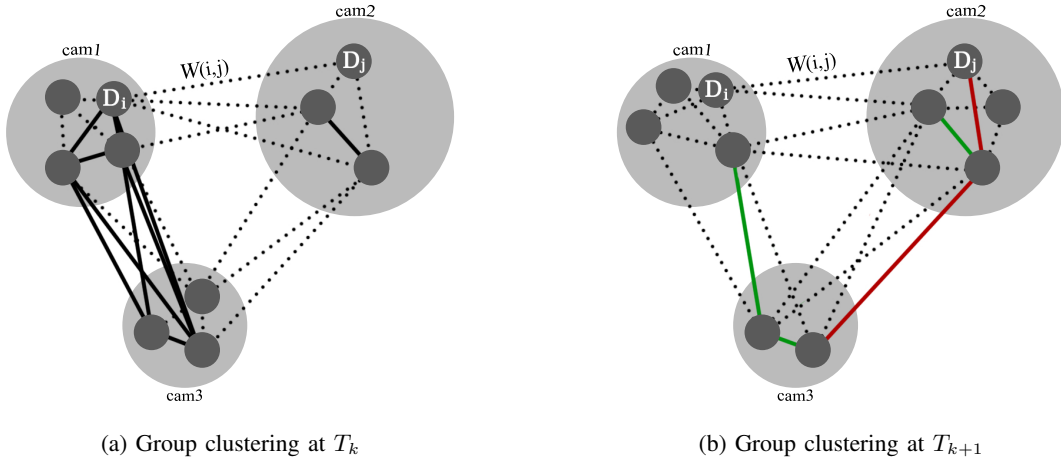(a) Group clustering at T_k (b) Group clustering at T_{k+1}

Fig. 2: The problem of tracking groups is cast as correlation clustering. In (a) all the detected groups D_i observed in the time window T_k are taken into account, all the pairwise correlation $W(i,j)$ are computed (dashed lines) and a solution to tracking is found (solid lines). In (b), since time windows overlap in time, T_{k+1} will include group associations that were already solved in T_k. The new clustering is thus constrained by the previous solution forcing some observations to join (green lines) and others to remain separated (red lines), inducing consistent results across different time windows.

- uses an orthogonal feature encoding that allows learning camera-dependent feature weights (Sec. IV-B);
- learns scene-dependent weights and biases to avoid arbitrary thresholds on features similarities (Sec. IV-A); and
- casts the problem in terms of correlation clustering [16], thereby providing an optimal solution for the given group similarity measures (Sec. IV).

IV. FEATURE ENCODING FOR CORRELATION CLUSTERING

We propose to formulate group tracking as a clustering problem and solve it through *Correlation Clustering* (CC) [15], similarly to what was done for individual tracking in single camera views [16], [39]. With this formulation, all observations that refer to the same group—whether from the same or from different cameras—are meant to fall inside the same cluster. The CC algorithm takes as input a correlation matrix W where, if $W(i,j) > 0$ ($W(i,j) < 0$), observations i and j refer to the same (different) group with certainty $|W(i,j)|$. The algorithm returns a partition Y of a set $X = \{D_1, D_2, \dots\}$ of group observations that maximizes the sum of the affinities between item pairs that are in the same cluster:

$$\max_{Y \in \mathcal{Y}(X)} \sum_{y \in Y} \sum_{\substack{i,j \in y \\ i \neq j}} W(i,j), \qquad (2)$$

where $\mathcal{Y}(X)$ is the set of all possible partitions of the set X of group observations. Remarkably, correlation clustering doesn't need to know the number of groups in advance. Moreover, by solving single- and multi-camera tracking jointly, a group could be positively tracked even if two consecutive detections from the same camera would not qualify to be clustered together, as it could happen due to visual clutter or occlusion.

Nevertheless, differences in camera position, people density, colors and light across views make it hard to define a unique best correlation score between groups. Withal, hand-crafting

correlations from multiple features, balancing the contribution of different features, and jointly setting distance-to-correlation thresholds is no easy task, as many parameters need to be tuned. In this section we present a linear parametrization of the correlation matrix W that can be easily learnt from examples.

A. From Distances to Correlations

To this end, let $\mathbf{d}(i,j) = [d_1(i,j), d_2(i,j), \dots, d_m(i,j)]^T$ be a feature vector containing a set of distances computed on different features extracted from group observation i and j, *e.g.* HSV histograms, SIFT matching, or motion prediction errors (detailed in Sec. VI-A). While distances are non-negative and increase as groups i and j are more distinct, correlations in W need to be more negative (positive) when the two groups are considered more distinct (similar). As we previously proposed [13], we can rewrite the correlation matrix W of Eq. (2) in terms of distances $\mathbf{d} \in [0,1]$ with the following $[\boldsymbol{\alpha}, \boldsymbol{\beta}]-$parametrization:

$$
\begin{aligned}
W(i,j) &= \mathbf{w}^T \mathbf{f}(i,j) \\
&= \boldsymbol{\alpha}^T [\mathbf{1} - \mathbf{d}(i,j)] \quad + \quad \boldsymbol{\beta}^T \mathbf{d}(i,j) \\
&= \underbrace{\boldsymbol{\alpha}^T \mathbf{1}}_{\text{distance-to-correlation}} \quad + \quad \underbrace{(\boldsymbol{\beta} - \boldsymbol{\alpha})^T \mathbf{d}(i,j)}_{\text{feature combination}}.
\end{aligned}
\qquad (3)
$$

The term $(\boldsymbol{\beta} - \boldsymbol{\alpha})^T \mathbf{d}(i,j)$ modifies distances into positive or negative correlation by modulating individual features weights. Concurrently, the bias $\boldsymbol{\alpha}^T \mathbf{1}$ adjusts the threshold on distance that is needed to define whether observations i and j are to be considered similar or dissimilar. Both parameters are learned from training examples with no need of manual distance thresholds or other tuning.

B. Feature Encoding for Multiple Cameras

By changing the set of parameters $\mathbf{w} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$ we can explore different correlation functions and clustering solutions,

both in terms of which features contribute most and of where the similar/dissimilar thresholds are set. In particular, we want to tailor parameters to each camera pair separately. To learn all these correlations simultaneously we need to replicate the weight vector $\mathbf{w}$ for $\binom{c+1}{2}$ times for c cameras. Accordingly, we apply an orthogonal encoding of the feature vector $\mathbf{f}$ where the set of features $\mathbf{f}(i,j)$ is shifted to a specific position depending on the cameras where groups i and j were observed, and leaving all other feature values to zero. Formally, if $\mathbb{1}[s]$ denotes the Iverson bracket function being either 1 or 0 according to the truth of statement s, and c_i and c_j the camera where D_i and D_j where observed, then the orthogonal encoding can be written as:

$$\mathbf{f}(i,j) = \begin{bmatrix} \mathbf{f}(i,j)\mathbb{1}[\{c_i,c_j\} = \{1,1\}] \\ \mathbf{f}(i,j)\mathbb{1}[\{c_i,c_j\} = \{1,2\}] \\ \dots \\ \mathbf{f}(i,j)\mathbb{1}[\{c_i,c_j\} = \{a,b\}] \\ \dots \end{bmatrix}, \quad (4)$$

for all $a,b = 1,2,\dots,c$. With only slight abuse of notation, we still refer to the extended weight vector and the encoded feature vector as $\mathbf{w}$ and $\mathbf{f}$ respectively. Every element in the extended vector $\mathbf{f}(i,j)$ is now a $\mathbf{0}$ vector, except for a vector of features correctly positioned according to the cameras c_i and c_j involved. In section V we describe the learning algorithm employed to automatically set $\mathbf{w}$ from examples.

C. Consistent Solutions from Overlapping Windows

Once weights are learnt, the inference problem can be solved for each time window T_k. Since these windows overlap, it is important to guarantee consistency from one window to the next one. To this end, we force (or deny) associations for groups that where already observed - and thus tracked - in the previous time window by inserting highly positive (or negative) values in the correlation matrix W. Fig. 2 depicts and summarizes the inference step and the procedure followed to ensure a consistent clustering.

V. STRUCTURAL SVM FOR GROUP TRACKING

The input of the algorithm that learns the weights $\mathbf{w}$ described above includes a tracking time window T_k, the set $X_k = \{D_1, D_2, \dots\}$ of groups detected in it, the pairwise features $\mathbf{f}_k$ orthogonally encoded as in Sec. IV-B and computed on all possible pairs of group observations i and j, and the respective tracking solution Y_k, i.e., a clustering of those observations into unique group identities. Since a partition Y_k of X_k is a structured output, we adopt the *Structural SVM* (SSVM) [17] framework to model and learn the solution. SSVM has been previously employed in the single camera multi-target tracking field [40]; however groups were not considered leading to a different objective function to optimize.

The goal of SSVM is to learn a classification mapping $h : \mathcal{X} \to \mathcal{Y}$ between input space $\mathcal{X}$ and structured output space $\mathcal{Y}$ given a training set of input-output pairs $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$. A discriminant score function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is defined over the joint input-output space and

$F(X,Y)$ can be interpreted as measuring the compatibility of X and Y. The prediction function h is then

$$h(X; \mathbf{w}) = \arg\max_{Y \in \mathcal{Y}(X)} F(X,Y; \mathbf{w}) \quad (5)$$

where the maximizer over the label space $\mathcal{Y}(X)$ is the predicted label, i.e., the solution of the inference problem of Eq. (2). Following the definition of CC in Eq. (2) and its parametrization introduced in Sec. IV, the compatibility of an input-output pair is described as

$$F(X,Y; \mathbf{w}) = \mathbf{w}^T \sum_{y \in Y} \sum_{\substack{i,j \in y \\ i \neq j}} \mathbf{f}(i,j) = \mathbf{w}^T \Psi(X,Y), \quad (6)$$

where $\Psi(X,Y)$ is a combined feature representation, a *feature map*. The problem of learning in structured output spaces is formulated as the SSVM n-slack, margin-rescaling, maximum-margin problem:

$$\min_{\mathbf{w}, \boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n} \sum_{k=1}^{n} \xi_k$$

$$\text{s.t.} \quad \forall k : \xi_k \geq 0,$$
$$\forall k, \forall Y \in \mathcal{Y}(X_k) \backslash Y_k : \mathbf{w}^T \delta\Psi_k(Y) \geq \Delta(Y,Y_k) - \xi_k, \quad (7)$$

where $\delta\Psi_k(Y) \stackrel{\text{def}}{=} \Psi(X_k, Y_k) - \Psi(X_k, Y)$, ξ_k are the slack variables that allow possible margin violation, $\Delta(Y_k, Y)$ is the loss function further defined in Sec. V-A, and C is the regularization trade-off parameter. At a glance, the objective is to maximize the margin and jointly guarantee that, for a given input, every possible output differs from the correct one by a margin at least $\Delta(Y_k, Y)$, a quantity that increase according to the difference of its arguments.

A. Loss Function and Maximization Oracle

The quadratic program in Eq. (7) introduces a constraint for every possible wrong clustering of the n examples, for a total of $\sum_{k=1}^{n}(|\mathcal{Y}(X_k)| - 1)$ constraints. Unfortunately, the number of ways to partition a set X scales more than exponentially with the number of items according to the Bell sequence [41], making the optimization intractable. Subgradient methods are an efficient way to approach the training of SSVMs [42]. In particular, all the constraints in Eq. (7) can be replaced by n piecewise-linear ones by defining the structured hinge-loss:

$$\widetilde{H}(X_k) \stackrel{\text{def}}{=} \max_{Y \in \mathcal{Y}} \Delta(Y_k, Y) - \mathbf{w}^T \delta\Psi_k(Y). \quad (8)$$

The computation of the structured hinge-loss for each element k of the training set amounts to finding the most "violating" output Y_k^* for a given input X_k and its correct associated output Y_k. The solution Y_k^* has simultaneously a high score $\mathbf{w}^T \Psi(X_k, Y_k^*)$ and a high loss $\Delta(Y_k, Y_k^*)$, underlining a contradiction in the description ability of the current $\mathbf{w}$. Having defined $\widetilde{H}(X_k)$, the SSVM problem can be written as:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \max\{0, \widetilde{H}(X_i)\}, \quad (9)$$

and solved with subgradients, where $\partial_{\mathbf{w}}\widetilde{H}(X_k) = -\delta\Psi_k(Y_k^*)$.

From Eq. (8), we see that the objective is to reduce the gap between the loss-function term and the weighted feature map difference. Ideally, the optimum is found when the inner product between the weight vector $\mathbf{w}$ and the feature map behaves exactly as the loss function. Thus, the choice of the loss function in SSVM is a crucial step for an efficient and effective learning algorithm.

Since the inference task of Eq. 2 is already NP-hard, the loss function should be carefully selected in order to be effective and eventually linear with respect to the solution. Then, we can solve the max oracle of Eq. 8 through the same algorithm employed by the inference step and, consequently, with the same complexity. If an efficient maximization oracle, *i.e.* a solver for Eq. (8), is available, the overall training becomes efficient as well.

In detail, in our scenario, the solution Y is a partition represented by a binary matrix of connected cluster components. The *Hamming loss* is then a natural choice for evaluating solutions discrepancies. Specifically we adopt a matrix representation for the output Y such that $Y(i,j) = 1$ if observations i and j refer to the same cluster and $Y(i,j) = 0$ otherwise. Through the following theorem, the max oracle becomes a CC problem:

Theorem V.1. The maximization oracle of Eq. (8) can be solved through CC if $\Delta(Y_k, Y) = \sum_{i,j} \mathbb{1}[Y_k(i,j) \neq Y(i,j)]$ and

$$Y_k^* = \arg\max_{Y \in \mathcal{Y}(X_k)} \sum_{y \in Y} \sum_{\substack{i,j \in y \\ i \neq j}} \left(\mathbf{w}^T \mathbf{f}_k(i,j) + 1 - 2Y_k(i,j)\right)$$

$$(10)$$

where $\mathbb{1}$ is the Iverson bracket function.

Proof. Let us add to Eq. (8) the $\arg$ to get the solution Y,

$$\arg\max_{Y \in \mathcal{Y}} \Delta(Y_k, Y) - \mathbf{w}^T \Psi(X_k, Y_k) + \mathbf{w}^T \Psi(X_k, Y). \quad (11)$$

Note that the second term does not depend on the specific choice of Y and the last term can already be cast as a CC problem by the definition of $\Psi(X, Y)$. By considering the matrix form of Y, the Hamming loss can be written as:

$$\Delta(Y_k, Y) = \underbrace{\sum_{Y_k(i,j)=1} (1 - Y(i,j))}_{\text{false negatives}} + \underbrace{\sum_{Y(i,j)=1} (1 - Y_k(i,j))}_{\text{false positives}}$$

$$= \sum_{Y_k(i,j)=1} 1 + \sum_{Y(i,j)=1} (1 - 2Y_k(i,j)),$$

$$(12)$$

since $\sum_{Y(i,j)=1} Y_k(i,j) = \sum_{Y_k(i,j)=1} Y(i,j)$ always counts true positives. Now, if we plug the loss decomposition of Eq. (12) into Eq. (11), since $\sum_{Y_k(i,j)=1} 1$ does not depend on Y, we obtain the max oracle formulation of Eq. (10), which is a CC with correlations defined by $\mathbf{w}^T \mathbf{f}_k(i,j) + 1 - 2Y_k(i,j)$. $\square$

As a consequence, in the loss-augmented problem of Eq. (12) the term $1 - 2Y_k(i,j)$ acts as a discount factor in terms of correlation for all the correct elements in Y_k. The affinity matrix used for solving the max-oracle through CC penalizes all the correct solutions while encouraging the wrong ones. At convergence, every most violated constraint contributed to change the weight vector to counterbalance this effect.

B. Subgradient Optimization

With the feature map, the loss, and the max oracle in place, we now describe the optimization procedure through the Block-Coordinate Frank-Wolfe algorithm [43], delineated in Alg. 1, which exploits the domain separability of the constraints and limits the number of oracle calls needed to converge to the optimal solution. The algorithm works by minimizing the objective function of Eq. (9) but restricted to a single random example at each iteration. By calling the max oracle upon the selected training sample (line 4) we obtain a new sub-optimal parameter set $\mathbf{w}_s$ by simple derivation (line 5). The best update is then found through a closed-form line search (line 6), greatly reducing convergence time compared to other subgradient or cutting plane methods.

Algorithm 1 Block-Coordinate Frank-Wolfe Algorithm

1: Let $\mathbf{w}^{(0)}, \mathbf{w}_i^{(0)} := \mathbf{0}$ and $l^{(0)}, l_i^{(0)} := 0$
2: **for** it $:= 0$ **to** maxIterations **do**
3:     Pick k at random in $\{1, \ldots, n\}$
4:     Solve $Y_k^* := \arg\max_{Y \in \mathcal{Y}} \Delta(Y_k, Y) - \mathbf{w}^T \delta\Psi_k(Y)$
5:     Let $\mathbf{w}_s := \frac{C}{n}\delta\Psi_k(Y_k^*)$ and $l_s := \frac{C}{n}\Delta(Y_k, Y_k^*)$
6:     Let $\gamma := \frac{(\mathbf{w}_k^{(it)} - \mathbf{w}_s)^T \mathbf{w}^{(it)} + \frac{C}{n}(l_s - l_k^{(it)})}{\|\mathbf{w}_k^{(it)} - \mathbf{w}_s\|^2}$ and clip to $[0,1]$
7:     Update $\mathbf{w}_k^{(it+1)} := (1 - \gamma)\mathbf{w}_k^{(it)} + \gamma\mathbf{w}_s$
       and $l_k^{(it+1)} := (1 - \gamma)l_k^{(it)} + \gamma l_s$
8:     Update $\mathbf{w}^{(it+1)} := \mathbf{w}^{(it)} + \mathbf{w}_k^{(it+1)} - \mathbf{w}_k^{(it)}$
       and $l^{(it+1)} := l^{(it)} + l_k^{(it+1)} - l_k^{(it)}$
9: **end for**

C. Notes on Complexity

Each training iteration and each prediction step requires to solve an instance of Correlation Clustering. As a result, an exact solution to the Binary Integer Problem would not let the training scale smoothly with the number of observations. Nevertheless, despite being also hard to approximate, the research community has put a lot of efforts on this problem and very good algorithms exist to deal with the complexity of CC. In particular, we adopt the *Adaptive-label ICM* [44] implementation[5], that reduced the number of variables required to solve the CC problem from approximately n^2 to n allowing to apply the algorithm even when the affinity matrix dimension reaches the order of hundreds of variables. When using an approximate oracle and inference solution, in exchange for a few more (but quicker) iterations, the learning algorithm is still guaranteed to converge, without any loss of accuracy with respect to optimal solutions [43].

---

[5]http://www.wisdom.weizmann.ac.il/~bagon/matlab.html

VI. EXPERIMENTS

The purpose of the presented experiments is twofold: First, we study how tracking performance on standard single-camera sequences benefits from group information; second, we assess the ability of our method to extend identities across multiple cameras. To evaluate group tracking in multi-camera scenes we provide the new `DukeMTMC-Groups` dataset and a suitable performance measure. Eventually, we discuss the importance of different features according to their contribution in the learnt distance metric and with respect to different camera views.

A. Features

We employ four different features to describe groups. While HSV histograms and SIFT account for visual similarity, we also use clues based loosely on location and time. These are particularly useful when appearance is poorly conserved across different camera views. However, the proposed framework can easily incorporate different or additional features, as explained in Sec. IV-A. For all the experiments presented in the remainder of this section, the distance vector $\mathbf{d}(i,j)$ of Sec. IV-A is composed as follows:

$$\mathbf{d}(i,j) = [d_{\text{HSV}}(i,j), d_{\text{SIFT}}(i,j), d_{\text{POS}}(i,j), d_{\text{TIME}}(i,j)]^T, \tag{13}$$

for any two group observation D_i and D_j.

HSV Histograms are computed on a subset of 5 evenly spaced frames from the group observations D_i and D_j. For each frame, a bounding box around the group is intersected with the foreground-mask (computed as in [45]) and quantized with 16, 4 and 4 bins for hue, saturation, and value respectively. The histograms extracted from the same group observation D_i are then averaged and normalized to obtain HSV_i and HSV_j. A simple histogram intersection measure is employed to define a distance between any pair of group detections D_i and D_j:

$$d_{\text{HSV}}(i,j) = 1 - \sum_{b=1}^{\#bins} \min\{\text{HSV}_i(b), \text{HSV}_j(b)\}. \tag{14}$$

SIFT Matching computes the Euclidean distance between any two SIFT descriptors from group observation D_i and D_j. A ratio test is employed between the two best matches to detect ambiguity[6]. SIFT features are extracted from a subset of 5 evenly spaced frames from the observation period and filtered through the foreground mask by checking whether the descriptor center is part of the foreground. The distance feature is then computed by taking the average of all the distances of the matched SIFT features and normalized to $[0,1]$:

$$d_{\text{SIFT}}(i,j) = \frac{1}{M Z_{\text{SIFT}}} \sum_{m=1}^{M} \sqrt{\sum_{b=1}^{128} \left(\text{SIFT}_i^m(b) - \text{SIFT}_j^m(b)\right)^2}, \tag{15}$$

where SIFT_i^m and SIFT_j^m are the histogram descriptors extracted from the m-th matched SIFT pair and $Z_{\text{SIFT}} = 255\sqrt{128}$ is a normalization coefficient that accounts for the maximum possible distance between descriptors.

[6]VLFeat lib [46] was used to extract SIFT and a standard threshold of 0.7 was used for the ratio test.

Position/Velocity Coherence is obtained by considering the minimum discrepancy value between a) the entering position and the forward predicted position and b) the exiting position and the backward predicted position, defined as follows. Suppose, without loss of generality, that D_i is observed before D_j. Then the forward predicted position is computed by adding to the exiting position p_{ex}^i of D_i its average exiting velocity $\bar{v}_{\text{ex}}^i$ (computed on the last 10 frames) multiplied by the number of frames n_f between the exiting position of D_i and the entering position of D_j. Similarly, the backward predicted position is computed by subtracting from the entering position p_{en}^j of D_j its average entering velocity $\bar{v}_{\text{en}}^j$, again multiplied by n_f. So if the forward discrepancy is $d_{\text{POS}}^{\text{for}} = p_{\text{en}}^j - p_{\text{ex}}^i - \bar{v}_{\text{ex}}^i n_f$ and the backward discrepancy $d_{\text{POS}}^{\text{bac}} = p_{\text{ex}}^i - p_{\text{en}}^j + \bar{v}_{\text{en}}^j n_f$, the position/velocity coherence is

$$d_{\text{POS}}(i,j) = \frac{\min\{|d_{\text{POS}}^{\text{for}}|; |d_{\text{POS}}^{\text{bac}}|\}}{Z_{\text{POS}}}, \tag{16}$$

where Z_{POS} is a normalization coefficient set to 200m, equal to the maximal error we want to measure. Finally, if $d_{\text{POS}}(i,j)$ is still greater than 1 we clip it to 1.

Time/Speed Coherence compensates for complex changes in direction occurring between camera views, for which position/velocity coherence does not capture errors appropriately. In particular, (if D_i is observed before D_j) the average entering and exiting speeds $\bar{s}_{\text{en}}^j$ and $\bar{s}_{\text{ex}}^i$ are averaged and used to predict the time needed to cover the distance between the entry and exit points p_{en}^j and p_{ex}^i. The time/speed coherence is measured by the discrepancy between this prediction and the actual time elapsed:

$$d_{\text{TIME}}(i,j) = \frac{1}{n_f} \left| n_f - \frac{2|p_{\text{en}}^j - p_{\text{ex}}^i|}{\bar{s}_{\text{en}}^j + \bar{s}_{\text{ex}}^i} \right|. \tag{17}$$

Since the normalization is done on the number of true frames and not the predicted ones (which could be more), we clip $d_{\text{TIME}}(i,j)$ to 1 if needed.

In addition, both d_{POS} and d_{TIME} (as well as their similarity counterparts in the feature vector $\mathbf{f}$) are discounted with an exponential function e^{-n_f} when evaluating group observations D_i and D_j that are increasingly farther in time. This allows the method to consider coherence for group detections which are close in time, while giving more importance to visual features when spatial and temporal predictions become too unreliable.

B. Data Sets

We selected three different datasets to conduct experiments with the proposed solution. The first two datasets are the public sequences `CBE-stu003` and `BIWI-eth`, both widely employed to test group detection algorithms. The sequences are recorded from a single camera with wide view and both groups and tracking annotations are publicly available on the datasets websites. These sequences contain both groups and singletons. The people density is lower in the `eth` sequence and higher in `stu003`. Details about the number of pedestrians, groups and groups composition are provided in Tab. I. We select the

first 1500 consecutive frames for parameter learning and the remaining ones for testing the algorithm.

For multi-camera evaluation we employ a new dataset, `DukeMTMC-Groups`, introduced here for the first time. The dataset is composed by 4×20 minutes of 1080p video recorded at 30 frames per second from 4 static cameras, deployed on the campus of Duke University during periods between lectures, when pedestrian traffic is heavy. Tab. II summarizes the peculiarities of this data set in terms of number of (unique) pedestrians and (unique) groups. Full annotations are provided in the form of trajectories for the feet of each person on the ground, and calibration data relates each image plane and the ground plane through a homography. Bounding boxes are also available and have been semi-automatically generated. This data set is part of a larger data set, `DukeMTMC`, comprising 8 cameras and 85 minutes of video, that will be soon released separately. We manually annotated `DukeMTMC-Groups` with group annotations maintaining group identities across cameras. A unique identity is given to each group as long as its member set is the same. Whenever a group splits, a new identity is created for each subgroup. The old identity is still maintained if the subgroups merge again. The first 5 minutes of video from each of the cameras are separated out to form a training set that can be used to set or learn parameters, and the remaining 15 minutes from each camera constitute the test set.

TABLE I: Public sequences data set statistics.

	Sequence	
	stu003	eth
# of unique individuals	434	362
# of groups	115	57
# of singletons	168	214
# of pairs	87	37
# of groups > 2	28	20

TABLE II: `DukeMTMC-Groups` data set statistics.

	CAMERAS			
	1	2	3	4
# of individual trajectories	80	88	35	49
# of unique individuals	128			
# of groups	40	45	16	24
# of frames for group trajectories (mean)	375	400	600	600
# of unique groups across cameras	64			

C. Evaluation Measure

There is no consensus in the literature on which measure to use to evaluate group detection and tracking performance. MOTA from CLEAR MOT [47] and GDSR [26] are the most used ones but both have failing aspects. In particular, GDSR (Group Detection Success Rate) precision (recall) counts how often a predicted (ground truth) group is found in the ground truth (prediction) by having at least 2/3 of its members in place. The choice of having a threshold, besides being arbitrary, doesn't let the measure distinguish between correct and loosely wrong solutions (in the extreme case, all predicted groups could be 30% wrong and still score a GDSR of 1). Moreover, the score is computed each frame separately, and then averaged over time. By not sharing information across

frames, GDSR is by all means a measure of group detection and not group tracking, despite it has been previously used to evaluate both tasks. Conversely, CLEAR MOT measures can be applied to the case of group tracking by considering average trajectories every time a group is composed by more than one member. Nevertheless, MOTA and companion scores are known to fail to successfully evaluate errors in the case of multi camera settings and have weaknesses in the case of single camera as well. Specifically, these shortcomings follow from CLEAR MOT inability to evaluate for how long –and not how often– an identity was tracked.

In the following experiments, we report CLEAR MOT measures whenever needed to compare to previous literature. Jointly, we introduce the MITRE measure [48] for group tracking and invite the community to consider it for future research. MITRE is founded on the idea that tracking is a clustering task, where each identity (of a group or of a singleton) corresponds to one cluster and all the observations in one cluster should belong from the same identity, both when they are observed from the same camera or across different ones. These observations can be frame-wise if the initial detector performs group detection frame-by-frame, or extended to be window-wise if the detector operates on a time window. This suggest that MITRE is capable of evaluating both the group detection and identity association (*i.e.* tracking) performances with one fell swoop. The only debatable point is that miss-matches are equally penalized independently of where they occur (*i.e.* at the beginning, in the middle or at the end of a trajectory). Yet, the significance of such considerations is still a controversial point in evaluating data association methods and no definitive answer exists.

The MITRE score is computed by transforming a tracking solution into a clustering solution by defining a track as the set/cluster of all the detected instances (*i.e.* group detections from all cameras) that have the same identity. The intuition behind the MITRE score is that a spanning forest is sufficient to represent the clustering solution, where each tree represents a group identity. Note that for any clustering solution, a spanning forest is an equivalence class, as multiple trees that describe the same identity configuration may exist. The final score is obtained by accounting for the number of links that need to be removed or added to recover a spanning forest equivalent to the correct solution. To keep the paper self contained we report in the following paragraph the MITRE computation algorithm. Consider two clusters Y_k and Y, and instances Q and R of their respective spanning forests. The connected components of Q and R are identified respectively by the set of trees $Q_1, Q_2, \ldots$ and $R_1, R_2, \ldots$ Note that with $|Q_j|$ elements in Q_j, only $l(Q_j) \stackrel{\text{def}}{=} |Q_j| - 1$ links are needed in order to create a spanning tree. We define $\pi_R(Q_j)$ as the partition of a tree Q_j w.r.t. forest R, as the set of subtrees obtained when considering only the membership relations in Q_j that are also in R. If R partitions Q_j into $|\pi_R(Q_j)|$ subtrees, then $v(Q_j) \stackrel{\text{def}}{=} |\pi_R(Q_j)| - 1$ links are sufficient to restore the original tree. Consequently, the recall error for Q_j is the number of missing links divided by the minimum number of links needed to create that spanning tree. The global

recall Q accounts for all trees Q_j and is computed as:

$$\mathcal{R}_Q = 1 - \frac{\sum_j v(Q_j)}{\sum_j l(Q_j)} = \frac{\sum_j |Q_j| - |\pi_R(Q_j)|}{\sum_j |Q_j| - 1} \quad (18)$$

The precision of Q (recall of R) can be computed by exchanging Q and R. Given the definition of precision, recall F-score F_1 is then computed as the standard harmonic mean.

As an implementation detail, since Y_k and Y must contain the same set of elements, the ground truth group detection data must be split in short temporal windows as well.

D. Do Groups Help in Tracking?

In these experiments we evaluate the performance improvement when tracking groups rather than every individual separately. To this aim, we employ the single-camera `stu003` and `eth` public sequences. Groups were extracted using the group detector of Solera *et al.* [13] on ground truth tracklets (approx. 6 seconds long). The group detector returns both groups and singletons (as 1-cardinality groups), and we track both of them. The detection window was empirically set to 250 frames while the tracking window was set to 750 frames with a stride of 250 frames. As a consequence, the input to our tracking algorithm is composed of approximately four detection instances in every tracking window. We evaluate the results using our proposed performance measure that accounts for both group detection and group tracking errors. Moreover, we provide quantitative results using the CLEAR MOT measures, a well established performance measure for single-camera tracking. In the CLEAR MOT evaluation, group trajectories have been obtained by averaging the trajectories of their members.

By looking at results in Tab. III, we observe that our solution (white rows) performs well in terms of both MITRE score and CLEAR MOT. The number of IDS (identity switch) in both sequences is low, while the high number of FRG (trajectory fragmentation) in `stu003` can be explained through the higher pedestrian density. On both `stu003` and `eth`, the MITRE score resembles the MOTA score in ranking methods, providing empirical evidence that MITRE is a competitive measure for the task. In order to highlight the importance of considering groups in crowded scenarios, in Tab. III (shaded lines) we report the results achieved by our algorithm when all the people in the scene are considered as singletons (*i.e.* no groups are present). By looking at these results the importance of considering groups in the tracking process is testified by the performance improvement over the case when all elements are considered as singletons and tracked individually. This evidence –in particular for high density scenarios as for `stu003`– suggests that groups have different visual and motion dynamics w.r.t. singletons and these peculiarities can improve the tracking quality when taken into account. Qualitative results on the sequences are shown in Fig. 4.

E. Results From an Automated Pipeline and Comparison

In the previous section, we have investigated group tracking by asserting single camera tracking could be solved, at least for a short time span. The community knows that is a strong
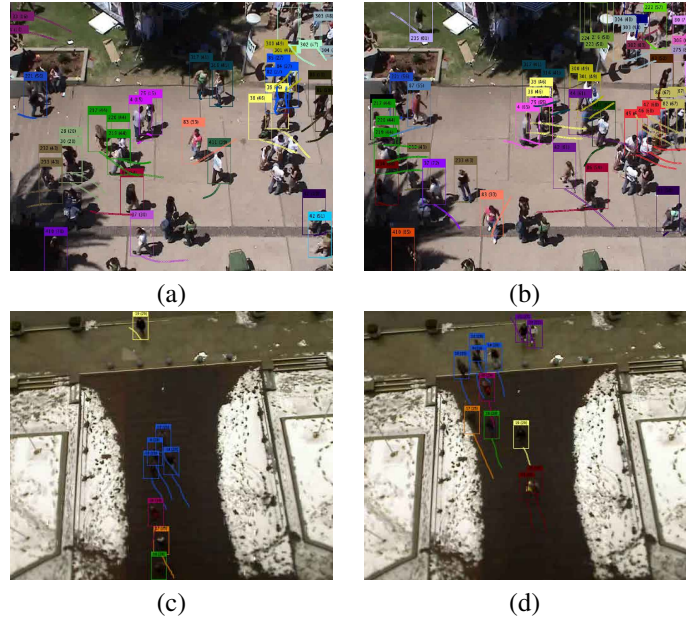

Fig. 4: Visual results of groups and singletons tracking on `stu003` (a,b) and on `eth` (c,d). Same color corresponds to same identities. Group members share the same color and group ID.

TABLE III: Tracking results on single camera public datasets. Standard tracking measures (CLEAR MOT) are computed by obtaining, for each group, an average trajectory, both in ground truth and predicted results. Shaded lines represent our method when groups are not considered (*i.e.* every one is a singleton). The proposed MITRE score emphasizes the set of identities found in each group more than their trajectories.

	CLEAR MOT				MITRE		
	MT	IDS	FRG	MOTA	P	R	F_1
`eth`	0.93	9	3	88.7	0.96	0.97	0.96
	0.86	12	8	81.9	0.89	0.93	0.91
`stu003`	0.77	10	98	79.2	0.74	0.79	0.76
	0.61	31	57	59.4	0.47	0.90	0.61

hypothesis and, while this choice is mandatory in order to assess the capability of our data association proposal, it leaves open questions on the practical applicability of the system as a whole. In this experiment, we propose to employ automated methods at all levels of the pipeline – from people detection to group tracking. The experiments are carried out on `eth` and `stu003` sequences, where trajectories were extracted with the CEM tracker [37] using the ACF detector [49]. The extracted trajectories were input to the group detector already employed in previous experiments but with a severely reduced temporal window, namely 1s for `stu003` and 2s for `eth`. The time window reduction is required to adapt the group detection to the length of stable tracklets, at the expense of loosing some group involved in complex motion and distance patterns. Observing Tab. IV, we can assess that impact of groups is beneficial for tracking only when both the density and the number of groups are consistently high. This is evident by comparing the MOTA improvement on `stu003` w.r.t. `eth`. The `eth`, as previously stated, contains few groups

TABLE IV: From left to right: people detection, people tracking, group detection and tracking of singletons and groups. On `eth` we also report results from [26], where the input to group tracking were detections from degraded ground truth trajectories (by 20%).

| | Individuals | | | | | | | Groups | | | | | | |
| | Detector | | Tracker | | | | | Detector | | Tracker | | | | |
	$\mathcal{P}$	$\mathcal{R}$	MOTA	MOTP	MT (%)	IDS	FRG	$\mathcal{P}$	$\mathcal{R}$	MOTA	MOTP	MT (%)	IDS	FRG
student	56.7	36.8	43.3	1.22	06.8	342	876	75.0	71.3	71.20	0.84	51.0	157	193
eth	68.2	53.7	92.3	0.80	75.0	21	68	67.3	64.3	47.38	0.85	61.9	48	97
eth [26]	80.0	80.0	-	-	-	-	-	-	-	29.43	0.44	-	-	-

TABLE V: Results on the proposed `DukeMTMC-Groups` data set. Shaded rows report results obtained with the baseline model (details in the text). Results refer to two different settings: (i) only groups are tracked and (ii) we track both groups and singletons in the *standard* last row.

| | Within-Camera Tracking | | | Across-Camera Tracking | | | Overall Tracking | | |
	P	R	F_1	P	R	F_1	P	R	F_1
Only groups	58.3	91.7	71.3	26.6	42.3	32.6	49.5	91.9	64.3
	87.1 ±2.6	93.9 ±1.8	90.4 ±2.0	76.67 ±2.8	75.00 ±2.0	75.82 ±2.1	82.2 ±1.3	83.7 ±0.9	83.0 ±0.5
Standard	69.58	83.36	75.85	46.34	49.30	47.77	69.55	71.85	69.11
	88.29 ±0.8	91.41 ±1.0	89.82 ±0.2	66.14 ±0.8	58.54 ±0.7	62.10 ±0.2	86.55 ±0.8	81.83 ±0.3	84.12 ±0.2

that rarely appear simultaneously. Moreover the most frequent path motion is straight with absence of occlusions, resulting in a problem easily solvable by any multi-target tracking method. Fairly speaking, the adoption of groups dynamics in tracking appears to be beneficial when dealing with medium to high density scenario and frequent occlusions, where standard tracking typically fails. Still, we report a comparison with another state of the art method [26] on `eth`. We show our method is able to obtain better tracking results even if a less accurate people detector was used. Unluckily, no other MOTA score is reported in literature on public dataset and further comparisons are not possible.

F. Results on DukeMTMC-Groups Multi-Camera Dataset

We evaluate the capability of our solution to deal with multi-camera scenarios on our `DukeMTMC-Groups` data set under different conditions (see Tab. V). Some illustrative examples are depicted in Fig. 5 and Fig. 1. The method was input with data from our group detector [13] and tested on both the tasks of tracking groups only and groups and singletons simultaneously. For both tests a baseline with no learning (*i.e.* by setting empirically the weights, prioritizing visual features) is provided (shaded rows in Tab. V). For evaluation, we used the MITRE score. To account for randomness in the training algorithm, tests were performed over 5 runs and mean and standard deviation values are reported. The detection window D has been empirically set to 5 seconds and the tracking window T to 2.5 minutes with an overlap between tracking windows of 100 seconds. The SSVM parameter C has been cross-validated to 10^4.

The quantitative results show that learning brings a significant improvement over the baseline score of $+15\%$ in term of F_1 score. In particular this difference is more evident when identities are matched across cameras, where features weights should be learned according to the source and destination cameras. In most of the multi-camera cases the baseline fails dramatically.

The performance improvement w.r.t. the baseline is mainly due to the success of the feature selection and weighting pro-

cess in our SSVM framework, whereby features are weighted according to the camera and the specific challenges of the data set. Moreover, the automatic thresholding and biasing scheme of Sec. IV-A finds the proper mapping from distances to correlations. In the considered case, when groups are automatically detected, performance are negatively affected by miss-detected groups as well as false positives. Although the method performs excellently in tracking groups only, its performance are almost the same even when both groups and singletons are considered (row *Standard* in Tab. V). This testifies that our method is able to adapt to the presence of singletons by finding a proper set of weights that deals with groups despite of their size (from cardinality 1 to n).

Additionally, in Fig. 7 we show and analyze performance of singleton and group tracking in terms of camera-camera F_1 scores. A cell in this matrix accounts only for association between two specific cameras. Within camera associations are solved with very few errors by our method (see diagonal row of F_1 matrix in Fig. 7). In particular, Camera 3 is an easy scene as people move in a constrained corridor and both their motion model and appearance remain consistent (refer to Fig. 5 for examples of camera viewpoints). In contrast, associations between Camera 1 and Camera 2 are the most difficult, mainly because of the viewpoint change between the cameras and a large number of mutual occlusions between group members. Camera 1 is a street scene where people can enter from both ends and walk either on the sidewalk (far field) or on the grass (near field). The challenges here are due to the presence of scene occlusions (*e.g.* a parked taxi), and the constrained motion patterns on the sidewalk versus the unconstrained ones on the grass. Moreover, the closer the groups are to the camera the more their members tend to occlude each other. Camera 2, on the other hand, is a fairly open scene where people can enter typically from a side view and group members, when arranged in lines, are often occluded by the member closest to the camera. Similar considerations hold for Camera 3 and Camera 4. Despite the viewpoint change, the unconstrained entry and exit points, and the mutual occlusions among group members, our method
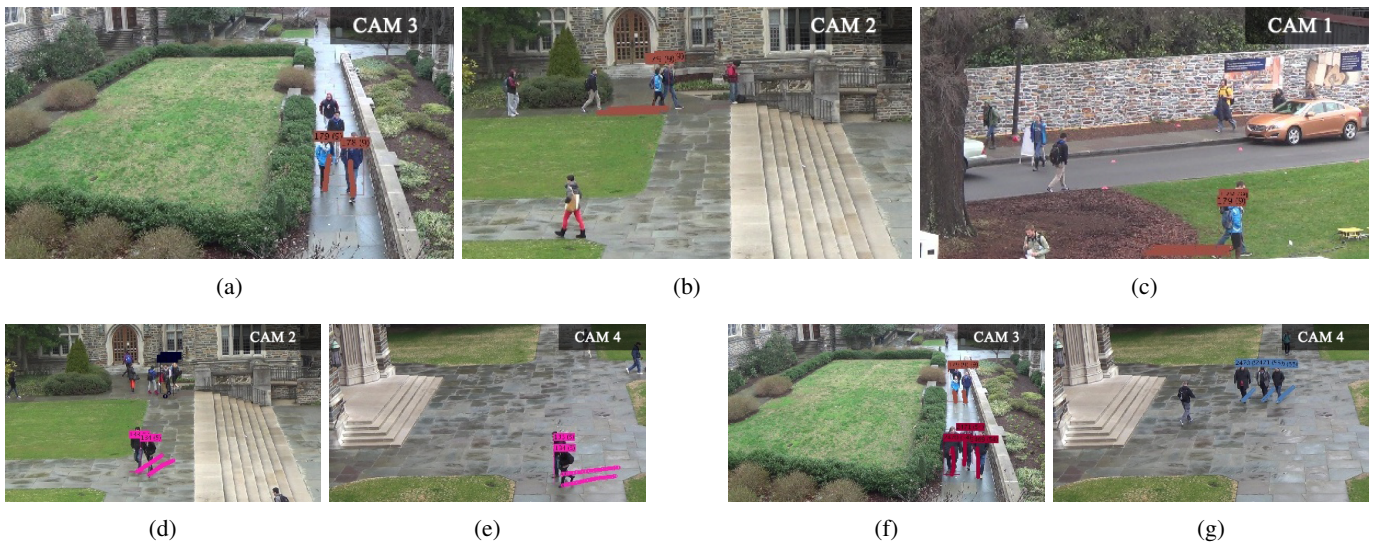
Fig. 5: Illustrative results obtained by the proposed method. In (a-c) and (d,e) two groups (brown and pink) were consistently tracked across 3 and 2 cameras, respectively. In contrast, (f,g) shows a failure case where the same group is identified differently in camera 3 and 4.
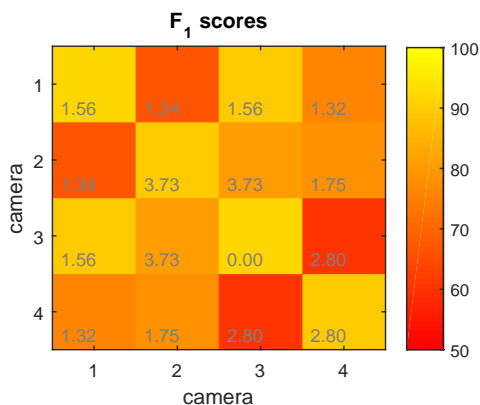

Fig. 7: This camera-camera matrix analyzes group and singleton tracking performance in terms of F_1. Averages of different runs are color encoded, standard deviation is text inside cells.

scores more than 60% in F_1 in both these challenging cases.

To our knowledge, this is the first work that addresses the problem of group tracking across multiple cameras, and we have no other methods to benchmark against. Our data set is the first of its kind with per-frame ground-truth annotation and consistency of group labels across cameras.

G. Feature Importance in Multi-Camera Group Tracking

To provide insight on the learning ability of our proposal, we also report the features weights learned during training in the case of group tracking alone (see Fig. 6a-d). In this experiment singletons have not been considered as the focus was to highlight which features are useful in handling groups in inter- and intra-camera associations.

The matrices shown in Fig. 6a-d report the learned feature importance in terms of difference between $w_s - w_d$, where $w_s \in \mathbf{w}$ is the parameter associated with the feature similarity

$(1-d)$ and $w_d \in \mathbf{w}$ is the parameter associated with the feature distance (d), for each pair of cameras and for each feature separately. It is interesting to observe how the system reacts to the different cameras. Specifically, by observing Fig. 6.a and 6.d, Camera 4 exhibits a significant variation in the colors w.r.t. Camera 2 due to illumination variation. This results in less importance assigned to the HSV features by the weights while the SIFT features is promoted in this case. The opposite occurs for example between Camera 2 and Camera 3 where the severe viewpoint change decreases the reliability of SIFT features. Accordingly, the system penalizes SIFT features in favor of HSV histograms. Fig. 6.d summarizes the relation among cameras in terms of their distance and placement on the map. Cameras with a high Time/Speed Coherence score are more likely to be close to each other and placed in a continuous and plausible path on the map. The Time/Speed Coherence scores for Camera 2 by itself show that most of the groups here break their motion continuity at some point. A straightforward interpretation is that groups tend to enter the scene and stop for a while in this zone. These considerations aside, a poor score in the feature importance matrices does not necessarily imply that the feature is neglected altogether, because in the clustering framework all features with non-zero weights contribute to the final tracking solution.

More generally, feature weights provide interesting information on the data set challenges and richness of information, while also capturing some aspects of the mutual relations among cameras. Besides appearance features, the weights of positional features can measure the usefulness of considering position in the multi-camera group association process: Low positional feature weights reflect a high distance between cameras, thus implying that speed and motion prediction cannot be unquestionably trusted in these cases while still being important in the single camera case (diagonal cells of Fig. 6.c). Our joint global formulation pursues the best
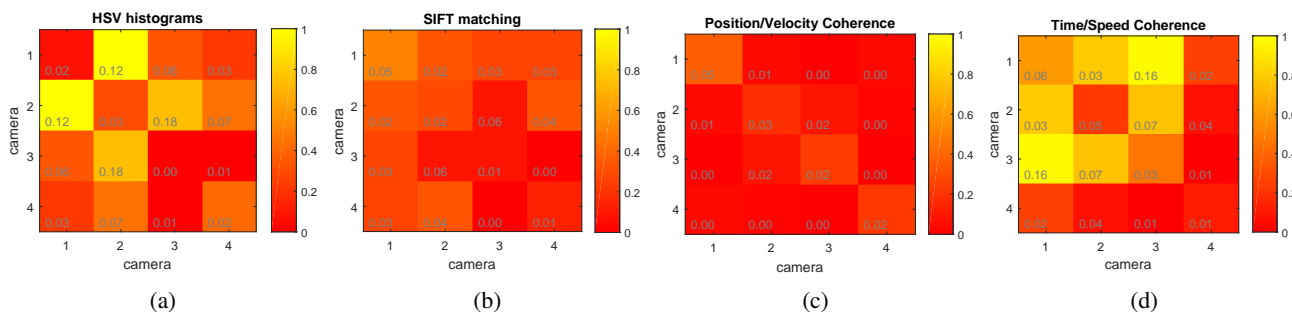
Fig. 6: Relative feature importance measured as $w_s - w_d$, where $w_s \in \mathbf{w}$ is the parameter associated with the feature similarity $(1 - d)$ and $w_d \in \mathbf{w}$ is the parameter associated with the feature distance (d), for each pair of cameras.

weight combination for the specific scenario, but still aims at exploiting as many features as possible.

VII. Discussion

In this work we propose a method for tracking groups in a multi-camera scenario. The tracking problem is cast as a correlation clustering problem coupled with a learning framework for feature weighting, selection, and global optimization. To our knowledge, this is the first proposal of this kind, handling single- and multi-camera groups and singletons association seamlessly within the same framework.

The lack of a data set on the topic, already observed also by other researchers in the field [26], reflects the novelty of the problem. Our `DukeMTMC-Groups` data set is the first multi-camera tracking data set with ground-truth annotations for group identity. This data set, in conjunction with the MITRE performance measure that evaluates single- and multi-camera group-tracking performance appropriately, are in our opinion important tools for future development in this research field.

Our method also implements a learning-to-cluster strategy as a starting point for addressing the uncertainties that affect visual group tracking. Both the proposed orthogonal encoding of features and the feature-to-correlation scheme perform well in a real scenario such as the `DukeMTMC-Groups` data set, which provides a new baseline for future experiments. Of course, there are still many open questions, including the handling of group splits and merges, which we leave for future research. The code and the data set can be downloaded[7] for evaluation and further improvement by the community.

References

[1] C. McPhail, *The Myth of the Madding Crowd*, ser. Social Institutions and Social Change. Aldine De Gruyter, 1991.

[2] D. Miller, *Introduction to Collective Behavior and Collective Action*. Waveland Press, 2013.

[3] D. Locher, *Collective Behavior*. Prentice Hall, 2001.

[4] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLoS ONE*, vol. 5, no. 4, 04 2010.

[5] S. Bandini, A. Gorrini, L. Manenti, and G. Vizzari, "Crowd and pedestrian dynamics: Empirical investigation and simulation," in *International Conference on Methods and Techniques in Behavioral Research*, 2012.

[6] M. Zanotto, L. Bazzani, M. Cristani, and V. Murino, "Online bayesian non-parametrics for social group detection," in *British Machine Vision Conference (BMVC)*, 2012.

[7] M. Feldmann, D. Franken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, April 2011.

[8] S. K. Pang, J. Li, and S. Godsill, "Detection and tracking of coordinated groups," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, no. 1, January 2011.

[9] L. Bazzani, M. Cristani, and V. Murino, "Decentralized particle filter for joint individual-group tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[10] R. Mazzon, F. Poiesi, and A. Cavallaro, "Detection and tracking of groups in crowd," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2013.

[11] W. Ge, R. Collins, and R. Ruback, "Vision-based analysis of small groups in pedestrian crowds," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 5, May 2012.

[12] M.-C. Chang, N. Krahnstoever, and W. Ge, "Probabilistic group-level motion analysis and scenario recognition," in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011.

[13] F. Solera, S. Calderara, and R. Cucchiara, "Socially constrained structural learning for groups detection in crowd," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. PP, no. 99, 2015.

[14] J. C. Turner, "Towards a cognitive redefinition of the social group," *Social identity and intergroup relations*, 1982.

[15] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, 2004.

[16] E. Ristani and C. Tomasi, "Tracking multiple people online and in real time," in *Asian Conference on Computer Vision*, November 2014.

[17] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *International Conference on Machine Learning (ICML)*, 2004.

[18] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014.

[19] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert, "Density-aware person detection and tracking in crowds," in *Proc. Int'l Conf. Computer Vision (ICCV)*, 2011.

[20] F. Solera, S. Calderara, and R. Cucchiara, "Structured learning for detection of social groups in crowd," in *Proc. IEEE Int'l Conf. Advanced Video and Signal Based Surveillance (AVSS)*, 2013.

[21] A. Kendon, *Conducting Interaction: patterns of Behavior in Focused Encounters*. Cambridge University Press, 1990.

[22] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegaz, and V. Murino, "Social interaction discovery by statistical analysis of f-formations," in *Proc. British Machine Vision Conference (BMVC)*, 2011.

[23] Y. D. Wang, J. K. Wuand, A. A. Kassim, and W. M. Huang, "Tracking a variable number of human groups in video using probability hypothesis density," in *Proc. Int'l Conf. Pattern Recognition (ICPR)*, 2006.

[24] M. Feldmann, D. Fränken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Trans. Signal Processing*, vol. 59, Apr. 2011.

[25] W. C. Lin and Y. Liu, "A lattice-based mrf model for dynamic near-regular texture tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, May 2007.

[26] L. Bazzani, M. Zanotto, M. Cristani, and V. Murino, "Joint individual-group modeling for tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 4, April 2015.

[7] https://github.com/francescosolera/MC-groups

[27] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Proc. Eur. Conf. Computer Vision (ECCV)*, 2010.

[28] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg, "Who are you with and where are you going?" in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[29] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, Apr. 2015, arXiv: 1504.01942. [Online]. Available: http://arxiv.org/abs/1504.01942

[30] Z. Qin and C. R. Shelton, "Improving multi-target tracking via social grouping," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

[31] W.-S. Zheng, S. Gong, and T. Xiang, "Associating groups of people," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2009.

[32] I. M. Creusen, S. Javanbakhti, M. J. Loomans, L. B. Hazelhoff, N. Roubtsova, S. Zinger *et al.*, "Vicomo: visual context modeling for scene understanding in video surveillance," *Journal of Electronic Imaging*, vol. 22, no. 4, 2013.

[33] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Computer Graphics Forum*, vol. 26, Sep. 2007.

[34] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[35] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 36, Jul. 2014.

[36] Y. Li, J. Zhu, and S. C. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[37] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 36, Jan. 2014.

[38] F. Solera, S. Calderara, and R. Cucchiara, "Learning to divide and conquer for online multi-target tracking," in *Proceedings of International Converence on Computer Vision (ICCV)*, December 2015.

[39] W. Luo, B. Stenger, X. Zhao, and T.-K. Kim, "Automatic topic discovery for multi-object tracking." in *AAAI*, 2015, pp. 3820–3826.

[40] A. Dehghan, Y. Tian, P. H. Torr, and M. Shah, "Target identity-aware network flow for online multiple target tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[41] G.-C. Rota, "The number of partitions of a set," *The American Mathematical Monthly*, vol. 71, May 1964.

[42] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," *JMLRW*, vol. 32, Jun. 2014.

[43] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, "Block-coordinate frank-wolfe optimization for structural SVMs," in *Proc. Int'l Conf. Machine Learning (ICML)*, 2013.

[44] S. Bagon and M. Galun, "Large scale correlation clustering optimization," *arXiv preprint arXiv:1112.2903*, 2011.

[45] J. Yao and J. Odobez, "Multi-layer background subtraction based on color and texture," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[46] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[47] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, 2008.

[48] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, "A model-theoretic coreference scoring scheme," in *Proceedings of the 6th Conference on Message Understanding*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995.

[49] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, 2014.

Francesco Solera obtained a master degree in computer engineering from the University of Modena and Reggio Emilia in 2013. He is now a PhD candidate within the ImageLab group in Modena, researching on applied machine learning and social computer vision.

Simone Calderara received a computer engineering master degree in 2005 and a PhD degree in 2009 from the University of Modena and Reggio Emilia, where he is now an assistant professor within the Imagelab group. His current research interests include computer vision and machine learning applied to human behavior analysis, visual tracking in crowded scenarios and time series analysis for forensic applications.

Ergys Ristani is a PhD candidate in the computer science department at Duke University. His research interests include tracking multiple people in multiple cameras and detecting occlusion boundaries in video.

Carlo Tomasi received a degree in Computer Science from Carnegie Mellon University in 1991. He was assistant professor at Cornell and Stanford, and is currently full professor of computer science at Duke University. He teaches undergraduate and graduate courses in computer vision and mathematics, and supervises students in computer vision research. His work emphasizes video analysis, image retrieval, and medical imaging.

Rita Cucchiara received her master degree in electronic engineering and the PhD degree in computer engineering from the University of Bologna, Italy, in 1989 and 1992 respectively. Since 2005, she is a full professor at University of Modena and Reggio Emilia, Italy, where she heads the ImageLab group and the SOFTECH-ICT research center. Her research focuses on pattern recognition, computer vision and multimedia.