Social Grouping for Multi-target Tracking and Head Pose Estimation in Video

Zhen Qin and Christian R. Shelton

Abstract—Many computer vision tasks are more difficult when tackled without contextual information. For example, in multi-camera tracking, pedestrians may look very different in different cameras with varying pose and lighting conditions. Similarly, head direction estimation in high-angle surveillance video in which human head images are low resolution is challenging. Even humans can have trouble without contextual information. In this work, we couple novel contextual information, social grouping, with two important computer vision tasks: multi-target tracking and head pose/direction estimation in surveillance video. These three components are modeled in a probabilistic formulation and we provide effective solvers. We show that social grouping effectively helps to mitigate visual ambiguities in multi-camera tracking and head pose estimation. We further notice that in single-camera multi-target tracking, social grouping provides a natural high-order association cue that avoids existing complex algorithms for high-order track association. In experiments, we demonstrate improvements with our model over models without social grouping context and several state-of-art approaches on a number of publicly available datasets on tracking, head pose estimation, and group discovery.

Index Terms—Multi-target tracking, multi-camera tracking, head pose estimation, social grouping, video analysis, context.

1 Introduction

It is difficult to achieve satisfactory results purely by using visual information for many computer vision tasks due to the inherent visual ambiguities in real-world images and videos. Take multi-camera tracking as an example. Pedestrians may look quite different under cameras with varying conditions. Another example is head pose estimation in high-angle surveillance video. (We focus on yaw angle estimation in such scenarios.) Human head images are usually of low resolution, which makes visual evidence unreliable (see Fig. 1). Thus, contextual information is needed for these tasks.

We introduce social grouping as one such context. Sociology research [29] shows that in natural scenes up to 70% of people walk in groups, possessing similar trajectories, speed, and destinations. These factors should help to disambiguate confusing tracking decisions in both single-camera (similar trajectories and speed) and multi-camera tracking (similar destinations). For example, in multi-camera tracking, the tracker usually finds it difficult to decide linking or splitting two detections, since one person usually looks quite different in two cameras. However, if the two detections are accompanied by another person, linking is preferred. It is also intuitively clear that when people form groups, their head directions are correlated, as they tend to look at each other or the same area of interest.

In this work, we provide a probabilistic framework with effective solvers to utilize social grouping for visual tracking and head pose estimation. The joint optimization of tracking and social grouping is modeled as a constrained nonlinear optimization problem, which results in steps involving standard fast procedures. Head pose estimation in groups is modeled as a graph labeling problem using a conditional random field (CRF) that allows exact convex learning and inference, with tractability supported by sociology research. The generality of our social grouping model makes it applicable to most existing tracklet linking and head pose estimation frameworks.

Our experiments show that social context can help in multi-target tracking and head pose estimation on real-world datasets. Of particular interest, social grouping provides a natural high-order cue for the single-camera multi-target tracking problem, while existing approaches usually depend on complex solvers to go beyond single-order association. Furthermore, social grouping is also an output of the complete system. Our model produces results that are comparative to or better than state-of-art methods on benchmark datasets (see Tbl. 1) on all three tasks (tracking, head pose estimation, and group discovery), though our model employs only simple motion and visual features.

Preliminary pieces of this work described the coupling of social grouping with single-camera [36] and multi-camera tracking [37]. In this paper, we also include head pose estimation and provide a unified view. In addition, we provide more comprehensive experimental results, including group discovery performance.

2 Related Work

Head pose estimation, group discovery, and especially multi-target tracking, have been extensively researched in the computer vision community. We focus on the literature that is most related to our work.

• *Z. Qin and C. R. Shelton are with the Department of Computer Science and Engineering, University of California, Riverside, Riverside, CA, 92521.*
E-mail: {zqin001, cshelton}@cs.ucr.edu
This work was supported by DARPA (FA8750-12-2-0010).

Fig. 1: (Left) Social grouping behavior not only generally exists in one scene, but also usually persists (with the same group members) across wide areas. (Right) Given head images alone, it is sometimes difficult for human beings to correctly identify head pose directions in challenging scenarios. Social context provides strong evidence for this difficult problem.

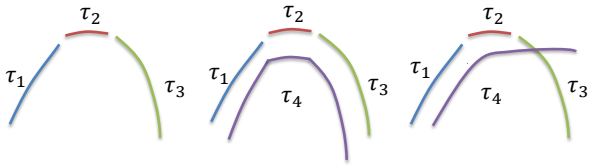


Fig. 2: (Left) Motion dependency problem for order-one association methods [48]: though $\tau_1 - \tau_2$ and $\tau_2 - \tau_3$ can be reasonably pairwise linked, the full trajectory is not probable. (Middle, Right) Social context from τ_4 gives strong evidence to disambiguate the dependency among tracks, indicating $\tau_1 - \tau_2 - \tau_3$ is probable (middle) or not (right).

Single-camera multi-target tracking. Multi-target tracking is a key step in many computer vision tasks, including visual surveillance, activity recognition, and scene understanding. Time-critical approaches usually use particle filtering algorithms for state estimation [51]. However, it is very difficult for such systems to handle long-term occlusions and detection failures. Thus recently, data association-based tracking (DAT, also known as the tracklet-linking problem) has dominated the research community. With the help of state-of-art tracklet extraction methods such as human detector approaches [25], researchers look at extended time periods and link conservatively extracted tracklets (short tracks) to recover full tracks. Many focus on how to obtain more reliable linking probabilities between tracklets [25][23][21]. To effectively infer the best matching given the affinity measurements among tracklets, different optimization methods such as the Hungarian algorithm [25][41], K-shortest path [4], MWIS [5], set-cover [46], min-cost flow [6], approximate dynamic programming [34], and continuous energy minimization [28] have been proposed. Some of them are shown to be equivalent to each other [19]. Importantly, these methods are mostly order-one methods, meaning that they optimize only pairwise similarities. This might lead to global inconsistencies. One typical problem is the motion dependency problem described in Fig. 2.

Yang et al. [48] employ a CRF model to mitigate the motion dependency problem for single tracks. Butt and Collins [6] use a relaxation to the min-cost network flow framework to explore higher-order smoothness constraints such as constant velocity. These models involve complex solvers and still possess limitations as they only address the motion dependency problem for single tracks: As shown in Fig. 2, the likelihood of one track with sudden motion change might depend on whether it is accompanied by a group member with a similar trajectory. Our model, on the other hand, models such scenarios by design, can be built upon simple solvers, and naturally helps higher-order tracking when coupling with social grouping information (modeled as a global spatial-temporal clustering procedure).

Multi-camera multi-target tracking. Multi-camera systems are ubiquitous, and a reliable multi-camera tracking system allows wide-area scene understanding. Researchers typically employ spatial-temporal and appearance cues to handover targets across cameras. For spatial-temporal information, Javed et al. [20] use a Parzen window density estimator to jointly model the inter-camera travel time intervals, locations of exit/entrances, and velocities of objects. Makris et al. [26] propose an unsupervised learning method to validate the camera network model. In terms of appearance similarity, Javed et al. [20] show that the Brightness Transfer Function (BTF) between cameras lies in a low dimensional subspace and proposes a method to learn them with labeled correspondences. A cumulative brightness transfer function (CBTF) is proposed by Prosser et al. [35] for mapping color between cameras using sparse training set. Kuo et al. [22] use Multiple Instance Learning (MIL) to learn a discriminative appearance affinity model online. The work by Orazio et al. [15] evaluates several BTFs and shows that they demonstrate similar behaviors and limitations. Our work, on the other hand, is the first to explore social grouping for the multi-camera tracking problem, which is more robust to changes in camera characteristics, viewpoints, and illumination conditions.

Head pose estimation. Head pose and gaze estimation is a long-studied area in computer vision and human computer interaction (HCI). It enables various applications such as human attention tracking and area or object of focus detection [27][1]. Most work focuses on head image classification where images possess reasonable resolutions and face landmarks are visible. Murphy-Chutorian and Trivedi [30] give an excellent review on diverse approaches towards this problem. Recent advances in this area include using part-based model [53]. In this work, we focus on head pose estimation in the common high-angle surveil-

lance video, also known as head direction estimation [10] and coarse gaze estimation [3]. Compared to traditional pose estimation work, the visual features of head images are usually very weak considering their small sizes, thus methods requiring face landmarks are not applicable. This problem is usually modeled as a regression problem (though discretized classes sometimes serve as an intermediate step, due to the ease of dealing with discrete labels over real-valued angles [10] [38]. Our work follows this approach), where angle difference between prediction and annotation, instead of classification accuracy, is measured, because of the difficulty of accurate class labeling [13] and the contiguity of nearby classes/angles in the feature space. Most work still focuses on feature extraction and estimation based on head images alone: Robertson and Reid [38] explore skin color feature. Tosato et al. [44] explore covariance features. The histogram of gradient (HoG) [12] is popular recently [3][10]. Support Vector Machine (SVM), SVM Regressor, Neural Networks, Decision Trees, and Nearest Neighbor classifiers are among classifiers/regressors applied [44][38][31]. The recent representative work by Benfold and Reid [3] employs structured learning, proposing a CRF for head pose estimation. Chamveha et al. [9] employ spectral clustering for scene adaptation. Chen and Odobez [10] couple head direction estimation with body pose in a general kernel learning framework. However, all these existing work only consider individuals. We consider general social tendency and repellence beyond individuals.

Group discovery. Social discovery has also drawn much attention in the computer vision community recently [47][18][8][40]. Ge et al. [18] infer social groups given a tracking result. By contrast, we perform grouping and tracking jointly. Chamveha et al. [8] use attention cue to help discovering groups, while we perform grouping first to aid head pose estimation. This is because we note that in challenging scenarios, head pose estimation can be more difficult than group discovery (Ge et al. [18] also note that trajectory information alone is enough to yield substantial agreement with human for the grouping task).

Socially-aware computer vision. Social context has been explored in a number of computer vision problems. For tracking, Pirsiavash et al. [32] proposed a more effective dynamic model based on social information, Pirsiavash et al. [33] and Yamaguchi et al. [47] infer grouping for better trajectory prediction and behavior prediction respectively. Bazzani et al. [2] focuses on tracking groups and Chen et al. [11] considers local group consistency. Ours is the first to consider social grouping context for the data association-based multi-target tracking and head pose estimation problem.

3 Social Grouping for Multi-target Tracking and Head Pose Estimation

We first introduce our notation and the probabilistic formulation of utilizing social grouping for multi-target tracking and head pose estimation as two maximum a posteriori (MAP) problems.

3.1 Notation

The input of our system is a set of n tracklets (possibly including false alarms) $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$ within a time interval [0, T], extracted by methods described in Sec. 6.2. Each tracklet τ_i is a sequence of short descriptions of a single target across the time interval [t_i^{start}, t_i^{finish}]. Such descriptions include the position and size of target (for the tracking problem), and the position and size of the pedestrian head (for the head pose estimation problem). In particular we let $a_i(t)$ be the camera (discrete camera labels) and $l_i(t)$ be the position (discrete pixel coordinates in the image) of τ_i at time t. We abuse $l_i(t)$ to denote both pedestrian and head positions.

The task of multi-target tracking is to determine which tracklets correspond to the same target, which can be represented as a binary correspondence matrix ϕ:

$$\phi_{i,j} = \begin{cases} 1 & \text{if tracklet } j \text{ immediately follows tracklet } i, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

with the added constraints that $\sum_j \phi_{i,j} = 1$ and $\sum_i \phi_{i,j} = 1$, indicating each tracklet should only follow and be followed by one other tracklet (except for the first and last tracklets of each track, addressed by virtual starting and ending tracklets in Sec. 4.4.1). We let Φ be the set of valid correspondence matrices.

For social grouping evaluation, we model it as a clustering problem and assume people form K groups, where K is unknown. Within each group, there is a group mean trajectory (a sequence of image coordinates) G_k, with $G = \{G_1, G_2, \ldots, G_K\}$. ψ denotes a binary social grouping assignment matrix:

$$\psi_{i,k} = \begin{cases} 1 & \text{if tracklet } i \text{ is assigned to group } k, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Again there is an added constraint that $\sum_k \psi_{i,k} = 1$ and we let Ψ be the set of valid social grouping matrixes.

For group head pose estimation, we will process each group independently at every time point so we drop the time stamp here. Let C denote the number of individuals in a group, Y denote the head directions of everyone in the group, Υ denote the head directions of all head images in the scene, X denote any existing unary evidence for individuals (such as image values or walking direction; there are M such features), and L denote the pedestrians' head locations. Let y_j and l_j be the head direction and location of the jth person, and x_j^i be the ith unary evidence for the jth person. Thus $Y = \{y_1, \ldots, y_C\}$, $L = \{l_1, \ldots, l_C\}$, $X^i = \{x_1^i, \ldots, x_C^i\}$, and $X = \{X^1, \ldots, X^M\}$. Information of X and L can be extracted from tracklet descriptions.

3.2 The Probabilistic Model Formulation

The inference of tracking, group discovery, and head pose estimation given inputs can be modeled as two maximum a posteriori (MAP) problems:

$$(\phi^*, \psi^*, G^*) = \underset{\phi \in \Phi, \psi \in \Psi, G}{\arg\max}\ P(\phi, \psi, G|\tau) \tag{3}$$

and

$$\Upsilon^* = \arg\max_{\Upsilon} P(\Upsilon|\phi, \psi, G, \tau). \quad (4)$$

In our work, the input to the second problem is the output of the first problem. Thus a single forward filtering of these two steps would output all desired information (tracking, group discovery, head pose estimation).

4 Coupling Social Grouping with Multi-target Tracking

We model the first MAP problem, $P(\phi, \psi, G|\tau)$, as

$$\begin{aligned} P(\phi, \psi, G|\tau) &\propto P(\phi, \psi, G, \tau) \\ &= P(G)\ P(\tau, \psi|G)\ P(\phi|\tau, \psi, G) \\ &= P(G)\ P(\tau, \psi|G)\ P(\phi|\tau, \psi), \end{aligned} \quad (5)$$

assuming group trajectories do not affect tracklet linking given grouping assignments. Next we explain each component of this model and the optimization algorithm.

4.1 Social Grouping as K-means Clustering

$P(\tau, \psi|G)$ is the data likelihood function of the probabilistic interpretation of clustering algorithms such as K-means clustering. We have

$$P(\tau, \psi|G) \propto \prod_{i,k|\psi_{ik}=1} P(\tau_i|G_k), \quad (6)$$

assuming trajectories for each individual are independent from each other given group mean trajectories (a similar assumption is made in general K-means clustering). $P(\tau_i|G_k)$ is the likelihood that tracklet i comes from group k, which we decompose across time as

$$P(\tau_i|G_k) = \prod_{t=t_i^{start}}^{t_i^{finish}} P(a_i(t)|G_k)\ P(l_i(t)|a_i(t), G_k). \quad (7)$$

$P(a_i(t)|G_k)$ is the probability that group k appears at camera $a_i(t)$, a parameter of the model for group k which we denote as $b_{k,a}(t)$. $P(l_i(t)|a_i(t), G_k)$ is the probability that at time t, a member of the group in camera $a_i(t)$ will appear at position $l_i(t)$, which we model as a Gaussian centered around the mean $u_{k,a}(t)$, the position for group k in camera a at time t, also a parameter of the model for group k. We use a fixed variance for all such Gaussians.

Notice that here we provide a general formulation for the multi-camera scenario. When it is the single-camera case, Eq. 7 can be significantly simplified ($P(a_i(t)|G_k)$ can be dropped).

4.2 Socially Constrained Multi-target Tracking

$P(\phi|\tau, \psi)$ measures the probability of tracklet linking (or track handover in the multi-camera case) given the social group information. Compared to traditional tracking methods, this adds a group constraint that if two tracklets are linked (they are the same person), they belong to the same group (one group per person):

$$\begin{aligned} P(\phi|\tau, \psi) = &\prod_{i|\forall m, \phi_{m,i}=0} P_{init}(\tau_i) \prod_{j|\forall m, \phi_{j,m}=0} P_{term}(\tau_j) \\ &\times \prod_{i,j|\phi_{i,j}=1} \begin{cases} P_{link}(i,j) & \text{if } \forall k, \psi_{i,k} = \psi_{j,k}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (8)$$

where $P_{init}(\tau_i)$ is the likelihood of τ_i being an initial tracklet, and $P_{term}(\tau_j)$ the likelihood of τ_j being the last tracklet. $P_{link}(i, j)$ is the likelihood that tracklet j is the first instance following tracklet i. These probabilities are the affinity model; any standard cues from the literature can be used (see Sec. 6.3).

4.3 A Simple Social Group Model

We model the probability of social groups as

$$P(G) \propto e^{-\kappa|G|}, \quad (9)$$

penalizing large numbers of social groups to avoid overfitting (such as placing each person in a separate group). Note that other heuristics are also applicable. Our choice is intuitive and results in a simple linear penalty in the optimization space, with its effectiveness validated in experiments.

4.4 Joint Optimization of Social Grouping and Multi-target Tracking

This section introduces the joint optimization of tracking and social grouping ($P(G)$, $P(\tau, \psi|G)$, and $P(\phi|\tau, \psi)$ in Eq. 5) as a constrained nonlinear optimization framework, which we call SGB (Social Grouping Behavior) algorithm.

We first reformulate the joint optimization of social grouping and multi-target tracking in the negative log space and achieve clean formulations. Then we introduce an effective optimization framework that can result in simple existing methods.

4.4.1 Optimization Reformulation

We perform the joint optimization of tracking and social grouping in the negative log-likelihood space (a minimization problem). Ignoring an additive constant from the proportionality in Eq. 9,

$$-\ln P(G) = \kappa|G|. \quad (10)$$

This term is in charge of selecting the number of groups and serves as the outer loop of optimization. Ignoring a similar additive constant, for $P(\tau, \psi|G)$ (Eq. 6), we have $-\ln P(\tau, \psi|G) = \sum_{i,k|\psi_{ik}=1} D(\tau_i, G_k) =$

$$\sum_{i,k|\psi_{ik}=1} \sum_{t=t_i^{start}}^{t_i^{finish}} -\alpha \ln b_{k,a_i(t)}(t) + \beta \left| l_i(t) - u_{k,a_i(t)}(t) \right|^2 \quad (11)$$

from Eq. 7 where α and β are weights relating to the variance of the Gaussian. For simplicity, we define $D(\tau_i, G_k)$

to be the "distance" of tracklet i from group k as above. In the single-camera case, the distribution $b_{k,a}(t)$ is degenerate and drops out of the equation.

$P(\phi|\tau,\psi)$ (Eq. 8) can be transformed to an assignment problem by defining a $2n \times 2n$ tracklet linking matrix

$$H = \left(\begin{array}{c|c} H^{link}_{n\times n} & H^{term}_{n\times n} \\ \hline H^{init}_{n\times n} & \infty_{n\times n} \end{array} \right) \tag{12}$$

with $H^{link}_{i,j} = -\ln P_{link}(i,j)$, $H^{init}_{i,i} = -\ln P_{init}(\tau_i)$, $H^{term}_{i,i} = -\ln P_{term}(\tau_i)$ and infinity ($-\ln 0$) elsewhere (including all diagonal elements). The virtual tracklets are introduced to handle track initializations and terminations. Eq. 8 is 0 if any assignments violate the constraint that linked tracklets must be in the same social group. Therefore, if we add this as a constraint: $\forall i,j,k\ \phi_{i,j}(\psi_{i,k} - \psi_{j,k}) = 0$, the resulting equation can be written in terms of H:

$$-\ln P(\phi|\tau,\psi) = \sum_{i,j} \phi_{i,j} H_{i,j} \tag{13}$$

Our optimization's outer loop tries different numbers of social groups ($P(G)$). Inside (optimizing $P(\tau,\psi|G)$ and $P(\phi|\tau,\psi)$), we can drop Eq. 10 and minimize the sum of Eq. 13 and Eq. 11 with the above constraint:

$$\begin{aligned} \min_{\phi\in\Phi,\psi\in\Psi,G} \quad & \sum_{i,j} \phi_{i,j} H_{i,j} + \sum_{i,k} \psi_{i,k} D(\tau_i, G_k) \\ \text{s.t.} \quad & \forall i,j,k \quad \phi_{i,j}(\psi_{i,k} - \psi_{j,k}) = 0. \end{aligned} \tag{14}$$

We call Eq. 14 the primal problem.

4.4.2 A Two-stage Alternating Minimization Algorithm

We use a two-stage iterative alternative optimization algorithm to solve the constrained nonlinear optimization problem in Eq. 14. The Lagrangian is

$$\begin{aligned} L(\phi,\psi,G,\mu) = & \sum_{i,j} \phi_{i,j} H_{i,j} + \sum_{i,k} \psi_{i,k} D(\tau_i, G_k) \\ & + \sum_{i,j,k} \mu_{i,j,k} \phi_{i,j} (\psi_{i,k} - \psi_{j,k}), \end{aligned} \tag{15}$$

where the μs are the Lagrange multipliers. The dual of this problem is

$$\begin{aligned} & \max_{\mu}\ q(\mu) \\ \text{where} \quad & q(\mu) = \min_{\phi\in\Phi,\psi\in\Psi,G} L(\phi,\psi,G,\mu). \end{aligned} \tag{16}$$

The resulting correspondence ϕ of the optimization is the output of the method. For a fixed μ, let

$$(\phi^\mu, \psi^\mu, G^\mu) = \arg\min_{\phi\in\Phi,\psi\in\Psi,G} L(\phi,\psi,G,\mu). \tag{17}$$

To solve Eq. 16, we use a quasi-Newton strategy with limited-memory BFGS updates and Wolfe line search conditions guided by the subgradient [39]:

$$\left.\frac{\partial q}{\partial \mu_{i,j,k}}\right|_{\mu} = \phi^\mu_{i,j}(\psi^\mu_{i,k} - \psi^\mu_{j,k}). \tag{18}$$

To calculate the subgradient, we use a two-stage block coordinate-minimization algorithm to solve Eq. 17. The first stage minimizes over ϕ (the tracklet correspondence result) from Eq. 15 with ψ and G fixed:

$$\phi^\mu = \arg\min_{\phi\in\Phi} \sum_{i,j} \phi_{i,j} [H_{i,j} + \sum_k \mu_{i,j,k}(\psi_{i,k} - \psi_{j,k})]. \tag{19}$$

This amounts to adding a penalty term to the matrix scores (compare with Eq. 13). So Eq. 19 is a standard assignment problem and can be efficiently solved by the Hungarian algorithm (or any algorithm designed for tracklet linking).

The second stage minimizes Eq. 15 over ψ and G, with ϕ fixed: $(\psi^\mu, G^\mu) =$

$$\arg\min_{\psi\in\Psi,G} \sum_{i,k} \psi_{i,k} [D(\tau_i, G_k) + \sum_j (\mu_{i,j,k}\phi_{i,j} - \mu_{j,i,k}\phi_{j,i})]. \tag{20}$$

This amounts to a standard K-means clustering problem. If the "centers," G, are fixed, the assignments, ψ, are made to minimize the augmented distance. When the assignments are fixed, the centers can be placed to minimize their distances to the captured points. Several initial group assignments are tried, as K-means converges to local minimum. The output of the one with the minimum value for Eq. 16 for one specific $|G|$ is maintained. At the end, we add the linear penalty of $|G|$ indicated by Eq. 10 and the outer loop (over $|G|$) selects the solution with the minimal negative log-likelihood score. See Alg. 1 for details.

Our method can be viewed as approximate max-product on the graph $G - \psi - \phi$ (in which the constraint forms the potential between ψ and ϕ). Direct variable elimination does not work, as it would require transmitting a *distribution* over all tracklet-tracklet-group triples. Dual decomposition [42] also results from a Lagrangian formulation, but is different from ours. We employ combinatorial optimization methods inside of max-product (our K-means and Hungarian algorithms) which has been explored in other max-product formulations [16].

Algorithm 1: SGB Algorithm

Data: Tracklet set τ
Result: Tracking ϕ_{Final}, Grouping ψ_{Final}

```
for K ← 1 to K_m do
    for i ← 1 to N do
        μ ← 0, φ^{K,i} ← 0
        initialize ψ^{K,i} and G^{K,i} randomly
        while Not local maximum for Eq. 16 do
            μ ← subgradient ascent: Eqs. 17 and 18
            while φ^{K,i} or ψ^{K,i} changes do
                Update φ^{K,i}: Eq. 19
                while ψ^{K,i} changes do
                    Update ψ^{K,i}: Eq. 20
                    Update G^{K,i} according to ψ^{K,i}
        Cost^{K,i} ← primal cost (φ^{K,i}, ψ^{K,i}, G^{K,i}): Eq. 14
(K*, i*) ← arg min_{K,i} Cost^{K,i} + βK
φ_Final ← φ^{K*,i*}, ψ_Final ← ψ^{K*,i*}
```

5 Socially-aware Head Pose Estimation

This section introduces the estimation of head poses given grouping information and tracking result ($P(\Upsilon|\phi,\psi,G,\tau)$). We formulate this problem as inference in a Conditional Random Field (CRF), discuss how we build the social interaction factor, and provide exact convex learning and inference procedures.

5.1 A Conditional Random Field Formulation

$P(\Upsilon|\phi,\psi,G,\tau)$ is the probability of head pose labeling in the video. In this work, we model the head poses of a group as a generative graph labeling[1] problem for each group at each time instance:

$$P(\Upsilon|\phi,\psi,G,\tau) = P(\Upsilon|\phi,\psi,\tau) = \prod_k P(Y_k|X_k,L_k), \tag{21}$$

assuming group mean trajectories do not affect head pose estimation given grouping assignments. Concentrating on a single group (one $P(Y_k|X_k,L_k)$ term), we drop the k subscript. By assuming a uniform prior on head poses, each evidence source is independent given the head pose, and each unary evidence (x) only depends on the person's head pose (y), we have

$$P(Y|X,L) = \frac{1}{Z}P(Y,L)\prod_i\prod_j P(y_j|x_j^i), \tag{22}$$

with Z being a normalization constant. We model pairwise social tendencies in the group for $P(Y,L)$. This problem can be modeled as a CRF as shown in Fig. 3. By using a log-linear model and ignoring the normalization constant, we get:

$$\begin{aligned}\ln P(Y|X,L) &\propto \sum_i \left\langle w_1^i, \Lambda_1^i(X^i,Y)\right\rangle + \langle w_2, \Lambda_2(Y,L)\rangle \\ &= \langle w, \Lambda(X,Y,L)\rangle\,,\end{aligned} \tag{23}$$

where

$$\Lambda_1^i(X^i,Y) = \sum_j \lambda_1^i(x_j^i, y_j), \tag{24}$$

and

$$\Lambda_2(Y,L) = \sum_{j_1 \prec j_2} \lambda_2(y_{j_1}, y_{j_2}, l_{j_1}, l_{j_2}). \tag{25}$$

$\prec$ is an ordering: we enumerate all unique pairs in a group. The subscript in λ_2 denotes a pairwise term. $\lambda_2(\cdot)$ is the feature vector for a pair of people that jointly models head pose labeling Y and locations L, with details described in Sec. 5.2, w_2 is the weight vector for these features, and $\langle\cdot,\cdot\rangle$ is the dot product. Evidence from unary factors (i.e. λ_1 and Λ_1) is represented similarly. $\Lambda(X,Y,L)$ is the feature vector composed of features from Λ_2 and Λ_1^i for all i (from 1 to M). w is a vector of parameters to be estimated (composed of the weights from w_2 and w_1^i for all i). This formulation allows exact convex learning and inference.

1. We use label and head pose direction interchangeably.

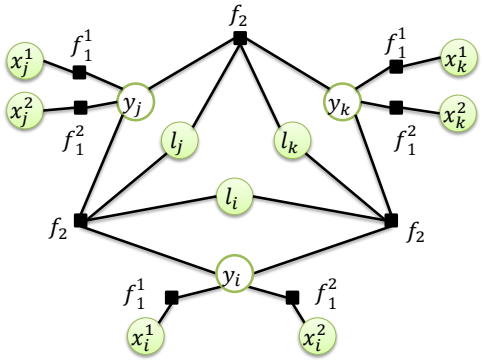


Fig. 3: A factor graph showing how variables and cliques interact in the CRF. A graph of three head images and only two unary features are shown for simplicity. If there are more people in a group or more unary features, this graph can be straight-forwardly augmented.

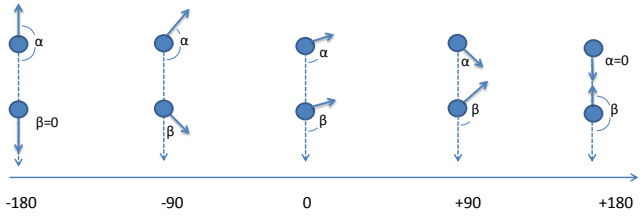


Fig. 4: Structure-aware head pose angle difference. Nodes are head images and dark blue arrows are head directions. Relative positions within group members are considered. The difference is simply $\beta-\alpha$. A positive number implies social attraction.

5.2 Building Group Interaction Models

We study the pairwise head pose interaction patterns in social groups for Eq. 25, which is key for using social grouping information to improve head pose estimation performance. We define the structure-aware head pose angle difference as illustrated in Fig. 4. We will use $SA(j_1,j_2)$, short for $SA(y_{j_1},y_{j_2},l_{j_1},l_{j_2})$, to denote this angle between the head directions of person j_1 and j_2. This angle takes into account the relative positions of the two people. Using structure information allows us to differentiate between social attraction and divergence when the absolute angle difference is the same. Given social groups, we collect such angle differences from only 200 pedestrian pairs from training data (the model data), identify two modes by thresholding velocity (a dataset dependent parameter in pixel/frames similar to that in Chamveha et al. [9]), and build the histograms shown in Fig. 5.

The resulting histograms are intuitive: (1) As shown in Fig. 5 (left), when people walk, they tend to look in the same direction (where they are heading generally or where an object of interest is), but there is more social attraction than divergence, as people tend to make eye contact with each other. We choose to model it with two exponential distributions on both sides of zero degrees. (2) As shown in Fig. 5 (right), when people are relatively stationary, they tend to look directly at each other (angle difference around +180 degree), or be attracted to common objects of interest

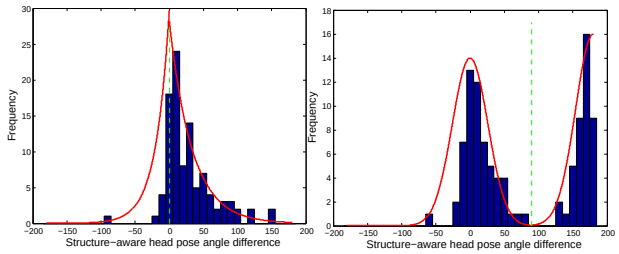


Fig. 5: Two social interaction modes with structure-aware head direction angle difference. Left: dynamical social interaction mode, fitted with two exponentials on either side of 0 degree. Right: static social interaction mode, fitted with two Gaussians on either side of 90 degrees. The specific distributions (exponential and Gaussian) are chosen due to their expressive power in this application and simplicity to express in the negative log space. The fitted distributions are rescaled and are for illustration only; their actual parameters are learned from training data.

(around 0 degrees, for example, when people scan shop windows). Though this is arguably a mixture of Gaussian, we model it with two Gaussians, separating at 90 degrees, for simplicity. The goal of learning is then to learn the rates of the exponentials and variances of Gaussians (feature weights in the negative log space).

These general forms can be converted into features so that the weights in Eq. 25 correspond to the rates and variances above. Given the group head pose interaction models, the feature vector of dynamical interaction mode for two head images (two exponentials on either side of 0 degree) is $\lambda_2^{moving}(y_{j_1}, y_{j_2}, l_{j_1}, l_{j_2}) =$

$$\begin{bmatrix} |SA(j_1,j_2)|\ I[SA(j_1,j_2) \geqslant 0] \\ |SA(j_1,j_2)|\ I[SA(j_1,j_2) < 0] \end{bmatrix}. \quad (26)$$

$I[\cdot]$ is the indicator function indicating the submode of social interaction for the pair $j_1 - j_2$. If the mode is off, the corresponding feature is 0.

The feature vector of the static interaction mode is $\lambda_2^{static}(y_{j_1}, y_{j_2}, l_{j_1}, l_{j_2}) =$

$$\begin{bmatrix} (SA(j_1,j_2) - 180)^2\ I[SA(j_1,j_2) \geqslant 90] \\ (SA(j_1,j_2))^2\ I[SA(j_1,j_2) < 90] \end{bmatrix}. \quad (27)$$

Similar to the feature vector in Eq. 26, these two features indicate which Gaussian submode is active and the corresponding feature value.

The dynamical interaction feature and static interaction feature can be unified as $\Lambda_2(y_{j_1}, y_{j_2}, l_{j_1}, l_{j_2}) =$

$$\begin{bmatrix} \lambda_2^{moving}(y_{j_1}, y_{j_2}, l_{j_1}, l_{j_2})\ I[\text{moving}] \\ \lambda_2^{static}(y_{j_1}, y_{j_2}, l_{j_1}, l_{j_2})\ I[\text{not moving}] \end{bmatrix}. \quad (28)$$

For example, if people are moving (estimated from tracking result), the dynamical interaction (moving) mode is on, and all features in λ_2^{static} become 0.

5.3 CRF Parameter Learning

Our CRF modeling allows exact convex discriminative learning. Note that we are interested in a regression problem, as the loss function models angle difference. However, using discrete and fine (32 bins) class labels make exact learning possible.

Let $X^{(m)}$ denote all unary features, $L^{(m)}$ denote head locations, and $Y^{(m)}$ denote the ground-truth labeling of group instance m. Further, let $\Lambda^{(m)}(Y) = \Lambda(X^{(m)}, Y, L^{(m)})$; thus $\Lambda^{(m)}(Y^{(m)}) = \Lambda(X^{(m)}, Y^{(m)}, L^{(m)})$ indicates a ground-truth feature-label configuration from training data. We conduct discriminative learning [43] of $P(Y|X, L)$ in the negative log space. Given N training examples, each of which is a graph labeling and related features, the objective function of training is $g(w) =$

$$\frac{1}{N}\sum_{m=1}^{N} \ln \sum_{Y} \left(\frac{P(Y|X^{(m)}, L^{(m)})}{P(Y^{(m)}|X^{(m)}, L^{(m)})} e^{l(Y^{(m)};Y)} \right) + \frac{\gamma}{2}||w||^2, \quad (29)$$

where $l(\cdot;\cdot)$ is the loss function for a group:

$$l(Y^{(m)}; Y) = \sum_j l'(y_j^{(m)}; y_j). \quad (30)$$

$l'(\cdot;\cdot) \in [0, 180]$ is the absolute difference between two directions. $\frac{\gamma}{2}||w||^2$ is a regularization term to avoid overfitting (γ is achieved via cross-validation in training).

After we apply Eq. 23, the objective function becomes

$$g(w) = \frac{1}{N}\sum_{m=1}^{N} \ln \sum_{Y} \Gamma^{(m)}(Y) + \frac{\gamma}{2}||w||^2 \quad (31)$$

$$\text{where } \Gamma^{(m)}(Y) = e^{l(Y^{(m)};Y) - \left\langle w, \Lambda^{(m)}(Y^{(m)}) - \Lambda^{(m)}(Y) \right\rangle}, \quad (32)$$

Eq. 31 is convex with gradient

$$\gamma w - \frac{1}{N}\sum_{k=1}^{N} \frac{\sum_Y \Gamma^{(k)}(Y)(\Lambda^{(k)}(Y^{(k)}) - \Lambda^{(k)}(Y))}{\sum_Y \Gamma^{(k)}(Y)}. \quad (33)$$

Since the objective function and gradient are explicit, minimization can be done exactly with any convex programming package, and we again use the one from Schmidt [39].

The complexity is $O(Q^C)$, where Q is the number of quantized head pose directions and C is the number of people in a group. Sociology research [29] shows that in natural scenes, people generally form groups of fewer than 6 people. This is also validated in the dataset we use. If the scene is really crowded (such as a Marathon event), large groups can be divided into smaller ones, or our model is not suitable since social interaction can be quite noisy in such cases. Running time is discussed in Sec. 7.5.

5.4 Head Pose Estimation Inference

Given model parameters (feature weights learned in the previous section), we perform head pose estimation inference by outputting

$$\max_Y \langle w, \Lambda(X, Y, L) \rangle \ , \quad (34)$$

which is the maximization of the log of $P(Y|X, L)$.

We use a brute-force approach to try all combinations of head directions for exact inference. The complexity is the same as learning, with tractability discussed above.

6 Details on Lower-level Tasks

Our framework is general in that it can be built upon different choices of lower-level components, such as tracklet extraction methods, features to build the tracklet affinity matrix, and unary features used for head pose estimation. We give details of our choices for implementation.

6.1 Parameter Estimation for Tracking

Parameters for tracking and group discovery include the feature weights for tracking, and κ for group number selection. They are estimated by a coarse grid search in the first time window in each dataset, and are fixed afterwards. In practice, feature weights are first selected for tracking without social grouping. Then κ is selected by a simple binary search after adding the social grouping term.

6.2 Tracklet Extraction

Our framework only requires the tracklet extraction method employed to be reliable (commonly assumed in the literature). Namely there should be few within tracklet identity switches. In order to perform comparative experimental evaluation, when tracklets from authors of published work are available, we use them. Otherwise we build our tracklet extraction framework based on human detection responses, combining nearest neighbor association and template matching to extract conservative tracklets. Given detection responses, we link detection response pairs only at consecutive frames which have very similar color, size and position. Additionally, the newly added detection must be similar to the first detection in the tracklet, thus avoiding within-tracklet ID switches caused by gradual changes. We find this simple strategy produces almost zero ID switches within tracklets and good recall performance.

6.3 Basic Affinity Model

Social grouping behavior regularizes the tracking solution and alleviates the need for a highly tuned affinity model. However, the basic affinity model must produce reasonable measurements, $H_{i,j}$. For both single-camera and multi-camera tracking, we build the basic affinity model using appearance (app) cues and spatial-temporal (st, usually referred as motion in single-camera tracking) cues:

$$-\ln P_{link}(i,j) = -\ln p^{app}_{i,j} - \ln p^{st}_{i,j}. \tag{35}$$

For single-camera tracking, we use the Bhattacharyya distance between the average color histograms within the tracklets [41]. We employ the HSV color space and get a 24-element feature vector after concatenating 8 bins for each channel. The motion model is a simple linear motion smoothness measure [25].

For multi-camera tracking, we use the BTF model and the Parzen window technique for spatial-temporal information in Javed et al. [20]. $P_{init}(T_i)$ and $P_{term}(T_i)$ are set to be a single constant (from training) for simplicity. There is also the time constraint that tracklet linking is only possible when tracklet j takes place later than tracklet i and within a maximum allowed frame gap t_{max}.

6.4 Spatial-temporal K-means Clustering

We describe how to implement the two steps of K-means clustering: group update (with group assignments given) and tracklet assignment (with group parameters given).

Recall that we modeled the group mean trajectory for G_k as, at each time t, a distribution over which camera a member of the group appears in, $b_{k,\cdot}(t)$, and a mean position within each camera a that a group member would appear, $u_{k,a}(t)$. Track assignment (finding ψ given a fixed G) is simple: for each tracklet τ_i, compute $D(\tau_i, G_k)$ from Eq. 11 for each group G_k and select the one that minimizes the negative log-likelihood.

For the update of G_k with the assignment ψ fixed, we must find the parameter assignments to $b_{k,\cdot}$ and $u_{k,\cdot}$ that maximize the likelihood. The log-likelihood is a sum across time, so the maximization can be done independently at each time point. $b_{k,a}(t)$ is a multinomial parameter and therefore its maximum likelihood estimate is proportional to the number of tracklets that are assigned to group k at time t in camera a.

$u_{k,a}(t)$ is the conditional mean for group k at time t in camera a. Therefore, its maximum likelihood parameter is the average position of all tracks assigned to group k at time t in camera a. If at any point there are no tracklets for group k and camera a, we use linear interpolation or extrapolation to generate a mean. If no tracklets in camera a are ever assigned to group k, we place $u_{k,a}(t)$ in the middle of the image for all t.

6.5 Unary Terms in CRF

Features from existing work can be used to construct unary features in our head pose estimation framework. We use two unary features. First, walking direction is shown to be effective in some datasets. As proposed by Benfold and Reid [3] and validated in our work, head pose direction is distributed approximately as a Gaussian with the walking direction as the mean. Thus in our negative log-likelihood framework, the unary feature of walking direction is

$$\lambda_1^{walking}(y_j, x_j^{walking}) = (y_j - x_j^{walking})^2. \tag{36}$$

We also build a two-level HoG vector to model visual features of head images, following Chen and Odobez [10]. Then we train a multi-class SVM with probability estimates [45]. Besides predicting labels, this allows us to estimate the probability of a visual vector belonging to each class. In this way, we have

$$\lambda_1^{HoG}(y_j, x_j^{HoG}) = -\log P(y_j|x_j^{HoG}). \tag{37}$$

where $-\log P(y_j|x_j^{HoG})$ can be directly obtained from the output of an SVM classifier with probability estimates.

TABLE 1: Datasets used for each task in the experiments.

Task	Datasets
Multi-target Tracking	PETS 2009, CAVIAR, TUD
Multi-camera Tracking	VideoWeb
Head Pose Estimation	PETS 2009, CAVIAR, TownCentre
Group Discovery	PETS 2009, PSUHub, TownCentre

TABLE 2: Comparison of the tracking result on the CAVIAR dataset: 75 ground truth (GT) tracks.

Method	Recall	Prec.	MT	ML	Frag	IDS
Particle filter	55.7%	60.4%	53.3%	10.7%	15	19
Basic affinity	81.1%	82.7%	77.3%	6.7%	9	12
MCMC[41]	84.5%	90.7%	84.0%	4.0%	6	8
SBM[52]	–	–	85.3%	4.0%	7	7
Our SGB	90.1%	95.1%	88.0%	2.6%	5	6

7 EXPERIMENT

We conduct comparative experiments with recent related methods on publicly available datasets for tracking, head pose estimation, and group discovery. Experimental results clearly show the benefits of utilizing social grouping context. The datasets we use is summarized in Tbl. 1

7.1 Single-camera Tracking Evaluation

We first evaluate how modeling social grouping behavior helps to improve single-camera multi-person tracking on the CAVIAR Test Case Scenarios dataset [7]. We use the videos selected by Song et al. [41], consisting of 12,308 frames for about 500 seconds. We retrieve tracklets from the same authors and use the same evaluation metrics by Li et al. [25]: the number of ground truth trajectories (GT), mostly tracked trajectories (MT), mostly lost trajectories (ML), fragments (Frag), ID switches (IDS), and recall and precision for detections. A comparison with several published results under the same configuration is shown in Tbl. 2. Our basic affinity model achieves reasonable results, while better results than competing methods can be achieved by employing our social grouping model with the simple affinity model.

Fig. 6 shows representative cases of the strong grouping information that allows us to improve tracking performance.

We further compare our model on the popular PETS 2009 and TUD-Stadtmitte datasets against a number of state-of-the-art methods using the same evaluation metrics. We obtained the publicly available detection results, ground truth data, and automatic evaluation tool from the authors of [50]. In addition to the former metrics, we also report the false alarm rate (FAF) for detections, and partially tracked trajectory ratio (PT) from the evaluation tool. In Tbl. 3 and Tbl. 4 we can see that our model outperforms several state-of-art methods, even though our model is built upon a simple basic affinity model. On the other hand, competing methods either solve complex optimization problems (Milan et al. [28] introduce six types of jumps in the optimization space) or build sophisticated affinity models (Kuo and Nevatia [21] use appearance features from the person identification literature). Of particular interest, for the PETS 2009 dataset, pedestrians were asked to travel across the scene multiple times. Even in such a scenario they formed groups and made social interactions, which is utilized by our model to help tracking. An example is shown in Fig. 8.

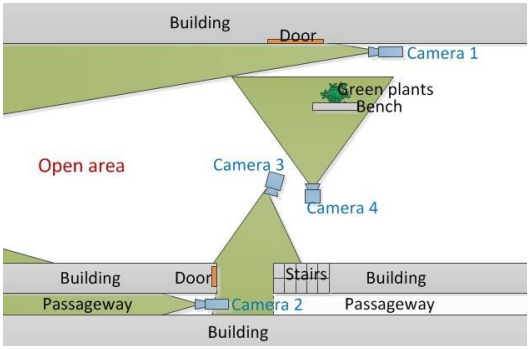


Fig. 7: Topology of the cameras in the experiments.

7.2 Multi-camera Tracking Evaluation

We test our method using two sets of videos on the publicly available VideoWeb dataset [14]. We choose Cam27, Cam20, Cam36 and part of Cam21 (indexed by 1–4) to establish the desired non-overlapping topology, shown in Fig. 7. Multi-camera tracking in this setting is very challenging for the following reasons. (1) We use 4 cameras, unlike most prior work that use 2–3. (2) This is an outdoor dataset with a cluttered environment and severe within-camera illumination change, which makes traditional methods that establish one single transformation between each camera pairs, such as BTFs, much less reliable. (3) Since this dataset is mainly designed for complex real-world activity recognition, there exist heavy interactions among individuals, unlike "designed" tracking datasets (for example the one in the work of Javed et el. [20]).

We compare our proposed multi-camera social grouping behavior tracking (MulSGB) to directly using the Bhattacharyya distance between RGB color histograms, Parzen window estimation for spatial-temporal information and the original color histogram for appearance (Parzen Window) and the BTF plus Parzen window estimation framework (Parzen Window + BTF) in the work of Javed et al. [20].

We gather 9 videos using all 4 cameras and 4 videos with camera 1–3. We use 5 videos from the first set for training and all the other videos for testing (note the second set of videos contains a subset of cameras of the first set so no additional training is needed). All other videos in the dataset either had no inter-camera motion or were missing data for more cameras. The data used have roughly 40,000 frames (25fps) for each of the four cameras for training and 80,000 frames for each camera for testing. For detection, we use a state-of-art pedestrian detector [17] to get detection responses and generate reliable intra-camera tracks using our introduced single-camera tracking framework. The same set of tracks are used for all comparing methods. We hand-labeled ground truth and measure the percentage of correctly linked pairs for the eight testing scenes (which consist of 244 single-camera tracks in total). Fig. 9 and Fig. 10 show the results for each set of videos.

We have the following observations. (1) Given the poor color histogram result, especially for the four-camera

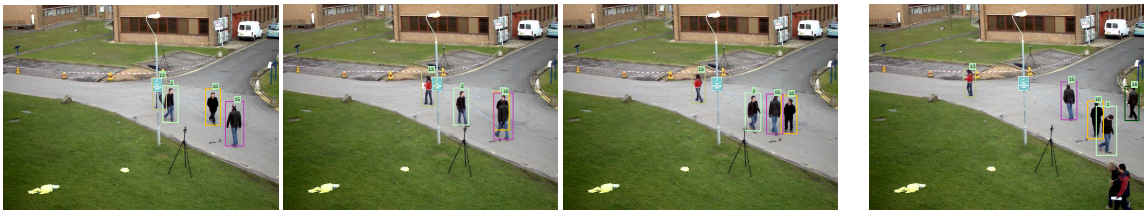
Frame 890 Frame 950 Frame 980 Frame 1010

(a) Even under heavy occlusions and interactions, the usual identity switch of 2 and 10 is avoided.

Frame 510 Frame 1245 Frame 1650 Frame 1682

(b) Long-term tracking of the couple (20,21) is possible under challenging conditions: small target, illumination change (frame 510), and false detection (frame 1245).

Fig. 6: Some representative tracking results for CAVIAR dataset.



Frame 477 Frame 483 Frame 489 Frame 509

Though there are heavy interactions between 10 and 15, social context from 2 helps to recover an ID switch.

Fig. 8: One representative tracking result for PETS dataset.

TABLE 3: Comparison of the tracking result on the PETS 2009 dataset.

Method	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
KSP [4]	83.8%	96.3%	0.160	23	73.9%	17.4%	8.7%	22	13
Energy Minimization [28]	92.4%	98.4%	0.070	23	91.3%	4.4%	4.4%	6	11
Online CRF [50]	93.0%	95.3%	0.268	19	89.5%	10.5%	0.0%	13	0
Nonlinear Motion [49]	91.8%	90.0%	0.053	19	89.5%	10.5%	0.0%	9	0
Our SGB model	97.2%	98.6%	0.077	19	94.7%	5.3%	0.0%	4	2

TABLE 4: Comparison of the tracking result on the TUD-Stadtmitte dataset.

Method	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
KSP [4]	63.1%	79.2%	0.650	9	11.1%	77.8%	11.1%	15	5
Energy Minimization [28]	84.7%	86.7%	0.510	9	77.8%	22.2%	0.0%	3	4
PRIMPT [21]	81.0%	99.5%	0.028	10	60.0%	30.0%	10.0%	0	1
Online CRF [50]	87.0%	96.7%	0.184	10	70.0%	30.0%	0.0%	1	0
Our SGB model	95.2%	98.5%	0.085	10	90.0%	10.0%	0.0%	4	3

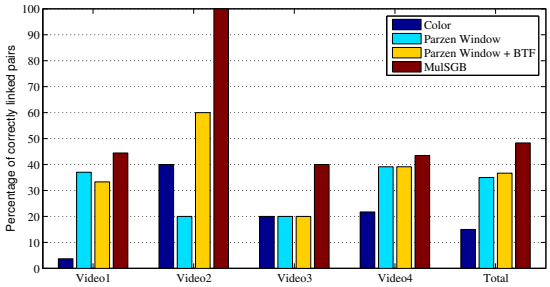


Fig. 9: Percentage of correctly linked pairs on the four video sequences with four cameras. The videos consist of 27, 5, 5 and 23 (60 in total) ground truth linked pairs respectively.

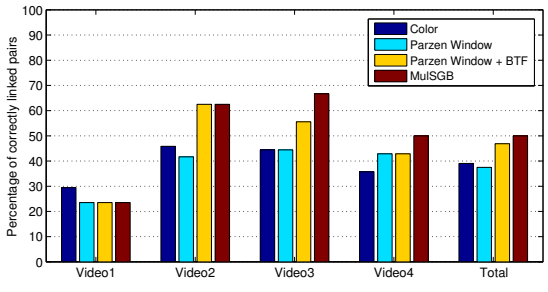


Fig. 10: Percentage of correctly linked pairs on the four video sequences with three cameras. The videos consist of 17, 24, 9 and 14 (64 in total) ground truth linked pairs respectively.

Frame 5809 (Cam 3) Frame 6222 (Cam 3) Frame 6295 (Cam 2) Frame 6689 (Cam 2)

Fig. 11: Example tracking result with our model, where G indicates group number. Because people form groups and show proximity to group members, social grouping provides powerful contextual information to improve multi-camera tracking. Other methods tend to identify a new person (Frame 6295 target 1) and output an identity switch (target 3 and 5) on this sequence, because traditional evidences are unreliable.

setting (demonstrating the difficulty of the dataset), the overall performance is good, as our MulSGB model indeed improves tracking performance over competing methods. (2) The example in Fig. 11 shows a representative example where social grouping helps tracking, while other methods fail under this challenging sequence. (3) Since our social grouping model serves as a regularizer, the basic affinity model upon which we built social grouping model is sometimes a bottleneck. For example, we observe no improvement upon the baseline model for two sequences in Fig. 10. We observed that in such cases, although the optimization usually heads toward a good solution, it could not recover wrong links since the basic model provides very unlikely handover possibility between the correct pairs. For example, when the illumination condition changes between the testing set and training set, the learned BTF may even hurt the performance comparing to pure color histogram comparison, as is the case for video1 in Fig. 10.

7.3 Head Pose Estimation Evaluation

We evaluate how social interaction improves head pose estimation in challenging videos, using the TownCentre dataset [3], CAVIAR, and PETS 2009. We use mean absolute angle difference (MAAD) stated in degrees as the evaluation metric, as is commonly done in related work. We quantized head pose into 32 directions, which is finer than most existing work (such as 8 directions [10][38]). This helps alleviating errors from coarse quantization when comparing angles. Competing methods that require discretization use the same setting.

We compare our method with models using visual features only (HoGSVM) and walking direction only (Walking). We also compare our method with a model with both visual and motion features. We call this model the BR (Benfold and Reid) setting [3]. Our implemented BR baseline does not incorporate temporal information. However, the resulting CRF can be solved exactly. We feel these two factors largely compensate each other as we get comparable results as those by Benfold and Reid [3]. Temporal information might be incorporated in our framework if approximate inference algorithms were applied. We also compare with two state-of-the-art methods: Orozco et al. [31] build a mean image for each class and represent each image as a distance map to these references. We use our own implementation with KL-Divergence as the distance measure (best reported measure in the paper). Tosato et al. [44] design a new visual feature and have publicly available implementation. Note that the small-sized head images make the comparison to landmark detection based work (e.g. [53]) impossible.

TABLE 5: Comparison of the head pose estimation results on the TownCentre, CAVIAR and PETS 2009 dataset. Numbers are reported on MAAD.

Method	TownCentre	CAVIAR	PETS
HoGSVM	31.20	28.80	32.64
Walking	23.89	72.01	58.28
DisMap [31]	33.12	30.20	31.54
WARCO [44]	31.12	25.70	28.65
BR Setting [3]	22.87	27.00	31.85
Ours	21.83	24.65	28.78

We use head images from people that are not in groups to train the multi-class SVM. Note we only report results for people identified in groups. For people that are not identified in groups, our model would output exactly the same result by using individual features alone. For the TownCentre dataset, about 30% of the people are identified in groups. For PETS 2009, over 40% of the people are in social groups. For the CAVIAR dataset over 60% are in groups.

We first use the TownCentre dataset to test our proposed method. This dataset has been used in several recent papers. It involves people traveling in a shopping mall. Though this dataset is treated as high-resolution video in the tracking literature, head images are small due to the high camera angle. We use the result of head tracking from Benfold and Reid [3] and use our spatial-temporal clustering procedure in Sec. 6.4 to determine groups. We manually label head directions for every 15 frames. Due to annotation differences, the angle differences are not directly comparable. But the performance we get from our BR setting baseline implementation is comparable to that of Benfold and Reid [3], which reports an MAAD of 23.90.

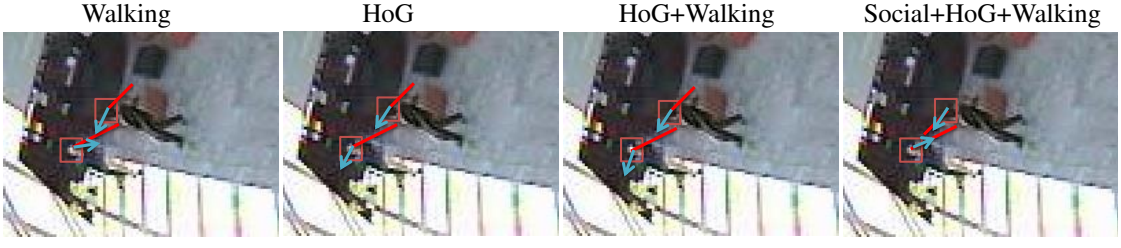
We gather 270 pairs of head images for this dataset. Whenever training is involved, 100 pairs are used for training and the others are used for testing. Since camera parameters are available for this dataset, we evaluate performance on the ground-plane. The results for different methods are shown in Tbl. 5.

As stated by Benfold and Reid [3], we also observe that walking direction provides a very good baseline in this dataset since most people are walking in the shopping

(a) Our model provides finer head direction estimate even when walking direction is reliable.

(b) Our model helps to correct head direction estimations in social groups with multiple people.

Fig. 12: Representative head direction estimation results for TownCentre. Red lines indicate human-labeled head direction.



(a) Our model corrects head direction for small head images as people interact.

(b) One case that our model is not able to fully recover false estimations.

Fig. 13: Some representative head direction estimation result for CAVIAR dataset.

mall. It can generate a better result than using only visual features. But even in such scenario, our model improves upon the best non-social method. As people walk together, their head directions tend to be attracted by other group members. Using social information regularizes out outliers that do not conform to such social constraints. Note that the performance gain from our social model is as large as the gain from combining two non-social information sources (comparing to using walking direction alone). We show two qualitative examples in Fig. 12.

We also compare performances on the CAVIAR dataset and PETS 2009 dataset. We annotate 5 video sequences[2] in CAVIAR and the entire PETS dataset at every 5 frames for head locations and head direction manually to focus on head pose estimation. For CAVIAR we gather 241 pairs of data, 100 of which are used for training and the others for testing. For PETS we gather 194 pairs of data and use half of them for training. Note for these two datasets we directly assign person ID and group ID based on our tracking model. That is, we do not assume ground truth identity or group member labeling and we evaluate head pose estimation performance in the complete system.

Compared to the TownCentre dataset, head images in these two datasets are of lower resolutions but possess lower variance because there are fewer people. CAVIAR involves more people standing still; the static mode of our social interaction model is more frequently activated and walking directions can be very noisy. People in PETS also show more freedom while walking so walking direction is again not as reliable as that in TownCentre. For these two datasets, we evaluate performance on the image plane.

We summarize the results in Tbl. 5. The performance gains by incorporating social context are more significant on these two datasets. They are much larger than the gain from combining the two non-social information sources (comparing to using visual feature alone.) This is because walking direction is often no longer a reliable feature and visual features are still weak. Yet, when people are relatively static, they tend to make more social contacts so our model helps more. Also, when walking, pedestrians' head direction severely deviates from walking direction,

2. FightChase, MeetSplit3rdGuy, FightOneManDown, MeetWalkTogether1, FightRunAway1

such deviations are usually motivated by group members or objects of interest, which is modeled in our formulation. We note that the reference-set based approach [31] does not perform very well due to its classification (instead of regression) formulation and the sparsity of training data. Our model performs comparatively with or better than the state-of-the-art method [44]. Some examples are shown in Fig. 13. We also show a case where our social model is not able to recover from false head pose estimations: Fig. 13(b). This is because our social model can be viewed as a regularizer, and it will not help much when the baseline model provides very bad evidence (for example, assigning very low probability to the true label).

7.4 Group Discovery Evaluation

Group discovery is provided by the group assignment matrix of our model. The simple spatial-temporal clustering approach is robust as a global consistency measure, while existing methods typically use features such as velocity, which can be unreliable with noisy detections or standing-still people. We show that our group discovery component can produce reasonable result compared to other designed approaches. The fact that our group discovery model is coupled with the tracking process (while other methods typically assume and are built upon perfect tracking result) makes our grouping approach more practical. When trajectories are available, the spatial-temporal clustering approach can be directly applied. We evaluate both cases.

Following Ge et al. [18], we use the following evaluation method: Each pedestrian is coded into one of two categories: alone or in a group. This is called the dichotomous coding scheme. A trichotomous coding scheme classifies each pedestrian into alone, in a group of two, or in a group of three or more. Match rate indicates the percentage of persons that are classified correctly. Furthermore, to test the statistical significance of the agreement between the human annotations and the output of the algorithm, Cohen's Kappa test [24] is used. Kappa score ranges from -1 to 1, and Landis and Koch [24] characterize values smaller than 0 as indicating no agreement and $(0, 0.2]$ as slight, $(0.2, 0.4]$ as fair, $(0.4, 0.6]$ as moderate, $(0.6, 0.8]$ as substantial, and $(0.8, 1]$ as almost perfect agreement.

Since we are not aware of group discovery results or annotations on the datasets we conduct tracking experiments on, or any available implementations of relating work, we are not able to conduct comparative experiments on these datasets. We thus annotate grouping in the PETS 2009 dataset. Our method produces 87% matching rate and a κ value of 0.75 for both dichotomous and trichotomous coding scheme (there are no trichotomous groups in the ground truth.) 55 trajectories are identified in time windows of 100 frames. (The same person in different time windows are treated as different persons [18].) We can achieve substantial agreement with human annotator on this dataset. If we focus on predicted pairs of people in social groups, for the 11 groundtruth pairs, our system achieves 91% recall and 71% precision.

TABLE 6: Comparison of the group discovery result on the PSUHub dataset.

	Match Rate	κ
dichotomous [18]	84%	0.74
trichotomous [18]	75%	0.63
dichotomous [ours]	83%	0.58
trichotomous [ours]	76%	0.49

We also compare our method with Chamveha et al [8] on the Towncentre dataset. Since their implementation is not available, we report the same measure, group accuracy (whether two people are in a group or not, compared with human annotation), as reported in the paper on the same dataset. We achieve an accuracy of 78.2% while they report 81.8%. The results are comparable and their method is based on the ground truth trajectories.

We further test our spatial-temporal clustering method against Ge et al. [18] on their publicly available PSUHub dataset and compare with their results. The dataset provides 2476 pedestrian trajectories in 177 time windows without images. We show the results in Tbl. 6.

We achieve comparative matching rates to a method designed solely for group discovery. Our model is inferior in terms of Kappa test, but we still get moderate agreement with ground truth. Note that our model is very simple to implement with only one parameter (weight for group size penalization, which is fixed across each dataset), while we are aware of at least four free parameters in Ge et al. [18]. Also, our method tends to group strangers that follow common path. Such pragmatic social groups still help tracking and head pose estimation. (Strangers may still follow common path, look at where they are heading to, or look at common object of interest.) Furthermore, the coupling of our clustering method with tracking makes it more practical when full trajectories are not available.

7.5 Running Time

We use a standard desktop and all our code is implemented in Matlab without specific optimization or parallelization. For the tracking problem, given tracklets and the affinity matrix H, the running time of our optimization depends on the implementation of the second-order gradient based method and scales with the number of tracklets. For the datasets in this paper, it takes 1 to 10 seconds to converge to a local maximum for each run on a time window. Though multiple runs with different random initializations are necessary to find a better optimum, our optimization is trivial to parallelize for each run. For head pose estimation, our implementation for training takes about one minute to converge to the global optimum with 100 pairs of data. Testing typically takes fewer than 5 seconds to finish (since no gradient descent is involved). Group discovery given full trajectories takes less than one second for each time window for the PSUHub dataset.

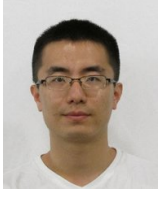
8 Conclusion

We show a general framework of coupling the novel social grouping context with important computer vision tasks including multi-target tracking and head pose estimation. Certain sub-components in our framework are naturally coupled and thus can be joint optimized. We then provide

effective solvers for those components based on nonlinear optimization and conditional random field. We conduct extensive experiments to show that social grouping context helps tracking and head pose estimation. Our social grouping model alone can also produce reasonable results.

REFERENCES

[1] J. Aghajanian and S. Prince. Face pose estimation in uncontrolled environments. In *BMVC*, 2009.
[2] L. Bazzani, M. Cristani, and V. Murino. Decentralized particle filter for joint individual-group tracking. In *CVPR*, 2012.
[3] B. Benfold and I. Reid. Unsupervised learning of a scene-specific coarse gaze estimator. In *ICCV*, 2011.
[4] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. PAMI*, 2011.
[5] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011.
[6] A. A. Butt and R. T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *CVPR*, 2013.
[7] CAVIAR. Caviar dataset. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.
[8] I. Chamveha, Y. Sugano, Y. Sato, and A. Sugimoto. Social group discovery from surveillance videos: A data-driven approach with attention-based cues. In *BMVC*, 2013.
[9] I. Chamveha, Y. Sugano, D. Sugimura, T. Siriteerakul, T. Okabe, Y. Sato, and A. Sugimoto. Head direction estimation from low resolution images with scene adaptation. *CVIU*, 2013.
[10] C. Chen and J. Odobez. We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video. In *CVPR*, 2012.
[11] X. Chen, Z. Qin, L. An, and B. Bhanu. An online learned elementary grouping model for multi-target tracking. In *CVPR*, 2014.
[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
[13] M. Demirkus, J. Clark, and T. Arbel. Robust semi-automatic head pose labeling for real-world face video sequences. *Multimedia Tools and Applications*, 2014.
[14] G. Denina, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, and B. Varda. Videoweb dataset for multi-camera activities and non-verbal communication. *Distributed Video Sensor Networks*, 2010.
[15] T. D'Orazio, P. Mazzeo, and P. Spagnolo. Color brightness transfer function evaluation for non-overlapping multi-camera tracking. In *ICDSC*, 2009.
[16] J. Duchi, D. Tarlow, G. Elidan, and D. Koller. Using combinatorial optimization within max-product belief propagation. In *NIPS*, 2006.
[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. PAMI*, 2010.
[18] W. Ge, R. Collins, and C. Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE Trans. PAMI*, 2011.
[19] J. F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multi-object tracking with merged measurements. In *ICCV*, 2011.
[20] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. Modeling inter-camera space-time and appearance relationships for tracking across non overlapping views. In *CVIU*, 2008.
[21] C.-H. Kuo, , and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011.
[22] C.-H. Kuo, C. Huang, and R. Nevatia. Inter-camera association of multi-target tracks by on-line learned appearance affinity models. In *ECCV*, 2010.
[23] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010.
[24] J. Landis and G. Koch. The measurement of observer agreement for categorical data. In *Biometrics*, 1977.
[25] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009.
[26] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *CVPR*, 2004.
[27] M. Marin-Jimenez, A. Zisserman, M. Eichner, and V. Ferrari. Detecting people looking at each other in videos. In *IJCV*, 2014.
[28] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE Trans. PAMI*, 2014.
[29] M. Moussaid, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behavior of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5, 2010.
[30] E. Murphy-Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. PAMI*, 31(4):607–626, 2009.
[31] J. Orozco, S. Gong, and T. Xiang. Head pose classification in crowded scenes. In *BMVC*, 2009.
[32] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.
[33] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010.
[34] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.
[35] B. Prosser, S. Gong, and T. Xiang. Multi-camera matching using bi-directional cumulative brightness transfer functions. In *BMVC*, 2008.
[36] Z. Qin and C. R. Shelton. Improving multi-target tracking via social grouping. In *CVPR*, 2012.
[37] Z. Qin, C. R. Shelton, and L. Chai. Social grouping for target handover in multi-view video. In *ICME*, 2013.
[38] N. Robertson and I. Reid. Estimating gaze direction from low-resolution faces in video. In *ECCV*, 2006.
[39] M. Schmidt. minfunc. http://www.di.ens.fr/~mschmidt/Software/minFunc.html.
[40] J. Sochman and D. Hogg. Who knows who - inverting the social force model for finding groups. In *ICCV Wrkshps*, 2011.
[41] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*, 2010.
[42] D. Sontag, A. Globerson, and T. Jaakkolar. Introduction to dual decomposition for inference. In *Optimization for Machine Learning*, 2011.
[43] C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 2012.
[44] D. Tosato, M. Spera, M. Cristani, and V. Murino. Characterizing humans on riemannian manifolds. *IEEE Trans. PAMI*, 2013.
[45] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.*, 2004.
[46] Z. Wu, T. H.Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *CVPR*, 2011.
[47] K. Yamaguchi, A. C. Berg, T. Berg, and L. Ortiz. Who are you with and where are you going? In *CVPR*, 2011.
[48] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In *CVPR*, 2011.
[49] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR*, 2012.
[50] B. Yang and R. Nevatia. Multi-target tracking by online learning a crf model of appearance and motion patterns. In *IJCV*, 2014.
[51] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 2006.
[52] S. Zhang, A. Das, C. Ding, and A. Roy-Chowdhury. Online social behavior modeling for multi-target tracking. In *CVPR Wrkshps*, 2013.
[53] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *CVPR*, 2012.



Zhen Qin Zhen Qin recieved his B.E. degree from Beijing University of Posts and Telecommunications in 2010 and Ph.D. from University of California, Riverside 2015. His research interests are computer vision and machine learning.His research interests are computer vision and machine learning.His research interests are computer vision and machine learning.

Christian R. Shelton Christian R. Shelton received his B.S. degree from Stanford in 1996 and Ph.D. from MIT 2001. After begin a postdoctoral scholar at Stanford, he joined the faculty at the University of California at Riverside where he is currently an Associate Professor of Computer Science. His research interests are in statistical approaches to artificial intelligence.