Who did What at Where and When: Simultaneous Multi-Person Tracking and Activity Recognition

Wenbo Li, Ming-Ching Chang, Siwei Lyu
Computer Science Department, University at Albany, State University of New York
{wli20,mchang2,slyu}@albany.edu
Ming-Ching Chang is the corresponding author.

Abstract—We present a bootstrapping framework to simultaneously improve multi-person tracking and activity recognition at individual, interaction and social group activity levels. The inference consists of identifying trajectories of all pedestrian actors, individual activities, pairwise interactions, and collective activities, given the observed pedestrian detections. Our method uses a graphical model to represent and solve the joint tracking and recognition problems via multi-stages: (1) activity-aware tracking, (2) joint interaction recognition and occlusion recovery, and (3) collective activity recognition. We solve the where and when problem with visual tracking, as well as the who and what problem with recognition. High-order correlations among the visible and occluded individuals, pairwise interactions, groups, and activities are then solved using a hypergraph formulation within the Bayesian framework. Experiments on several benchmarks show the advantages of our approach over state-of-art methods.

Index Terms—Group activity, collective activity recognition, pairwise interaction, multi-person tracking, hypergraph, high-order correlation.

I. INTRODUCTION

Multi-person activity recognition is a major component of many applications, e.g., video surveillance and traffic control. The problem entails the inference of the activities, the actors and their motion trajectories, as well as the time dynamics of the events. This task is challenging, since the activities are analyzed from both the spontaneous individual actions and the complex social dynamics involving groups and crowds [1]. We aim to address not only the where and when problem by visual tracking, but also the who and what problem by activity recognition.

While advanced methods for person detection are becoming more reliable [2, 3], most existing activity recognition approaches rely on visual tracking following a tracking-by-detection paradigm. These methods either fail to consider social interactions while inferring activities [4, 5, 6] or have difficulties recognizing the structural correlations of actions and interactions [7, 8, 9]. In particular, there are two major challenges: (i) ineffective tracking due to frequent occlusions in groups and crowds, and (ii) the lack of a suitable methodology to infer the complex but salient structures involving social dynamics and groups.

In this paper, we address both challenges using a bootstrapping framework to simultaneously improve the two tasks of multi-person tracking and social group activity recognition. We take state-of-the-art person detections [2, 3] as input to perform initial multi-person tracking. We then recognize stable group structures including the temporally cohesive individual activities (such as walking) and pairwise interactions (such as walking side-by-side, see Fig[1]) to robustly infer collective social activities (such as street crossing in a group) in multiple stages. Auxiliary inputs such as body orientation detections can be considered within the stages as well. The recognized activities and salient grouping structures are used as priors to recover occluded detections and false associations to improve performance.

We explicitly explore the correlations of pairwise interactions (of two individuals) and their group activities (within the group of more individuals) during the optimization. Observe in Fig[1] that group activities generally consist of correlated pairwise interactions, which we have exploited in the multi-stage inference steps. In our method, the multi-target tracking and the recognition of individual/group activities are jointly optimized, such that consistent activity labels characterizing the dynamics of the individuals and groups can be obtained. The individual and group activities are formulated using a dynamic graphical model, and high-order correlations are...
represented using hypergraphs. The simultaneous pedestrian tracking and multi-person activity recognition problems are then to be solved using an efficient cohesive cluster search in the hypergraphs.

The main contribution of this work is two-fold. First, we propose a new framework that can jointly solve the two tasks of real-time simultaneous tracking and activity recognition. Explicit modeling of the correlations among the individual activities, pairwise interactions, and collective activities leads to a consistent solution. Second, we propose a hypergraph formulation to infer the high-order correlations among social dynamics, occlusions, groups, and activities in multi-stages. Simultaneous tracking and activity recognition are formulated as a bootstrapping framework, which can be solved efficiently using the search of cohesive clusters in the hypergraphs. This hypergraph solution is general that it can be extended to include additional scenarios or constraints in new applications.

Experiments on several benchmarks show the advantages of our method with improvements in both activity recognition and multi-person tracking. Our method is easily deployable to real-world applications, since: (i) camera calibration is not required; (ii) online video streams can be processed by considering a time window in a round; (iii) the computation can be performed in real-time (about 20 FPS, not including the input detection steps).

II. RELATED WORKS

There exists a tremendous amount of multi-person tracking and activity recognition works. See [10], [11] for survey. Our work is most related to the collective activity recognition, which are organized into the following three categories — recognition based on (i) detection, (ii) tracking, and (iii) simultaneous tracking and recognition.

A. Collective Activity Recognition based on Detection

A hierarchical model is used in [12] to recognize collective activities by considering the person-person and group-person contextual information. The work of [13] uses hierarchical deep neural networks with a graphical model to recognize collective activities based on the dependencies of individual activities. This work is further extended in [9], where the individual and collective activities are iteratively recognized using RNN with refinements. Multi-instance learning is used in [14] to recognize collective activities by inferring the cardinality relations of individual activities. A recurrent CNN is used in [15] for the joint target detection and activity recognition.

B. Collective Activity Recognition based on Tracking

In this category, individual target trajectories are used as the input to recognize collective activities. Collective activities are recognized in [16] using random forests for the spatio-temporal volume classification. A two-stage deep temporal neural network is used in [4], where the first stage recognizes individual activities, and the second stage aggregates individual observations to recognize collective activities. In [17], the key constituents of activities and their relationships are used to recognize collective activities. A graphical model is developed in [18] to capture high-order temporal dependencies of video features. The and-or graph [19] is applied for video parsing and activity querying, where the detectors and trackers are launched upon receiving queries. A RNN architecture is designed in [20] to model high-order social group interaction contexts.

C. Simultaneous Tracking and Activity Recognition

Only very few works deal with the problem of simultaneous multi-person tracking and activity recognition. In [5], per-frame and per-track cues extracted from an appearance-based tracker are combined to capture the regularity of individual actions. A network flow-based model is used in [6] to link detections while inferring collective activities. However, these two methods did not consider pairwise interactions for activity recognition. In [7], [8], the tracking and activity recognition are formulated as a joint energy maximization problem, which is solved by belief propagation with branch-and-bound. However, high-order correlations among individual and pairwise activities are not considered, which limits the activity recognition performance.

III. METHOD

We start with notation definition in our method. Given an input video sequence, consider the most recent time window \( T = [t - \tau, t] \) in an online fashion, and denote previous time frames \([1, t - \tau - 1]\) as \( T'.\) Let \( D_T\) represent a set of target detections obtained using person detectors e.g. [2], [3]. Let \( X_{T'}\) represent the set of existing target trajectories. Let \( A_T, I_T, \) and \( C_T\) represent the set of recognized individual activities, pairwise interactions, and collective activities, respectively. Given \( D_T\), our approach aims to simultaneously solve the multi-person tracking and activity recognition problems, by inferring the following four terms within \( T:\) (i) target trajectories \( X_T = \{x_1, \ldots, x_b\}, \) where \( b\) is the number of observed targets, (ii) individual activity labels \( A_T = \{a_1, \ldots, a_b\}, \) (iii) pairwise interaction labels \( I_T = \{i_{1,2}, i_{1,3}, \ldots, i_{2,3}, \ldots, i_{b-1,b}\}, \) and (iv) collective activity labels \( C_T = \{c_{i-\tau}, \ldots, c_I\}, \) where \( c_f\) represents the collective activity with the most involved targets in the \( f\)-th frame. After a time window is processed, the method will extend target tracklets, update activity labels, and move on to the next time window: \( X_{t+1} = [X_{T'}, X_T], A_{t+1} = [A_{T'}, A_T], I_{t+1} = [I_{T'}, I_T], \) and \( C_{t+1} = [C_{T'}, C_T].\) To simplify notions, we omit the temporal indices to represent the variables within \([t - \tau, t]\) as \( X, A, I, C,\) and represent previous variables as \( X', A', I', C', \) i.e. \( X' = X_{T'}, A' = A_{T'}, I' = I_{T'}, C' = C_{T}..\)

A. Problem Formulation

We aim to infer accurate trajectories of all targets \((X)\) and their individual activities \((A)\), pairwise interactions \((I)\) and collective activities \((C)\) from the observed detections \((D)\). Relationship between these variables can be expressed as the joint distribution \( \Pr(X, A, I, C|D)\) as a dependency graph in
TABLE I
Notations for video activities, problem formulation and visual tracking.

| symbol | description |
|--------|-------------|
| $\chi$ | a target trajectory |
| $\alpha$ | an individual activity label, e.g. $\{\text{standing, walking, running}\}$ |
| $l$ | a pairwise interaction label, e.g. approaching (AP), facing-each-other (FE), standing-in-a-row (SR), ... |
| $c$ | a collective activity label, e.g., CROSSING, WALKING, GATHERING, ... |
| $b$ | number of observed targets (tracklets) |
| $T$ | video time window of length $\tau$ prior to time $t$, i.e., $T = [t-\tau, t]$ |
| $D$ | person detections (bounding boxes) |
| $X$ | target trajectories, $X_T = \{x_1, \ldots, x_h\}$ |
| $A$ | individual activity classes, $A_T = \{a_1, \ldots, a_b\}$ |
| $I$ | pairwise interaction classes, $I_T = \{i_1, 1.2, i_1.3, \ldots, i_1.b\}$ |
| $C$ | collective activity classes, $C_T = \{c_1, \ldots, c_c\}$ |
| $T', X', A', P, C'$ | existing entities prior to time window $T$, $X' = X_T$, $A' = A_T$, $P = I_T$, $C' = C_T$ |
| $n_A$ | number of individual activity classes, $n_A = 2$ in the CAD and Augmented-CAD datasets, $n_A = 5$ in the New-CAD dataset |
| $n_I$ | number of interaction classes, which is also the number of sub-hypergraphs used in our method, $n_I = 8$ in the CAD and Augmented-CAD datasets, $n_I = 9$ in the New-CAD dataset |
| $n_C$ | number of collective activity classes, $n_C = 5$ in CAD, $n_C = 6$ in Augmented-CAD, $n_C = 6$ in New-CAD datasets |

Problem Formulation

- $\Pr$ a joint distribution
- $\varphi_1, \varphi_2, \varphi_3$ clique potential functions in the Markov random field
- $X^+, A^+, I^+, C^+$ updated terms of $X, A, I, C$ after an optimization stage, respectively
- $p_{\text{dist}}$ the distance likelihood term for estimating the interaction between two targets
- $p_{\text{conn}}$ the group connectivity term for estimating the interaction between two targets
- $p_{\text{dir}}$ the direction term for estimating the interaction between two targets
- $p_{\text{face}}$ the facing direction likelihood term for estimating the interaction between two targets
- $\bar{x}$ a candidate tracklet
- $\bar{x}'$ a (putative) individual activity of a candidate tracklet
- $A$ the set of (putative) individual activities for all candidate tracklets
- $\theta_a$ the appearance similarity for tracklet linking
- $\tau_a$ time threshold for appearance-based tracklet linking
- $\oplus$ operator $\oplus$ represents the association of two tracklets
- $h$ the number of hypothetical tracklets to generate from an existing tracklet $x_i'$, $h = 9$

Fig[1] Based on the conditional independence assumption of $X, A, I, C$ in the graphical model, $\Pr(X, A, I, C|D)$ can be decomposed into three terms:

$$f_1(X, D) \cdot f_2(X, A, I) \cdot f_3(X, A, I, C). \quad (1)$$

(i) $f_1(X, D)$ is the confidence of target tracking, where the calculation will be given in §III-C (ii) $f_2(X, A, I)$ models the inter-dependencies among target trajectories, individual activity and pairwise interaction labels, which is further expressed as a Markov random field (red cycle in Fig.[1]):

$$f_2(X, A, I) \sim \varphi_1(X, A) \cdot \varphi_2(A, I) \cdot \varphi_3(I, X), \quad (2)$$

where $\varphi_1, \varphi_2$ and $\varphi_3$ are three clique potential functions capturing the inter-correlations between each variable pair. Derivation of these clique potentials will be given in §III-C and §III-D (iii) $f_3(X, A, I, C)$ reflects an important assumption that collective activities can be effectively modeled by robust inference of target trajectories, individual activities and pairwise interactions, where §III-D will provide details.

The inference of the joint tracking and recognition is then formulated as seeking $\arg \max \log \Pr(X, A, I, C|D) = X, A, I, C$

$$\arg \max_{X, A, I, C} \left\{ \log f_1(X, D) + \log \varphi_1(X, A) + \log \varphi_2(A, I) \right\}$$.  

However, standard iterative optimization such as block coordinate descent is not practical due to that: (i) the coupling of variables $X, A, I, C$ is still complicated; (ii) each of these variables represents a superset of time-dependent variables, so their joint optimization will be very inefficient; (iii) a real-time processing method is desired. We adopt a heuristic approximate solution using multi-stage updating scheme, which first jointly updates $X, A$, and then updates $I$, followed by the update of $C$. Our strategy is based on an important hypothesis that inferring pairwise interactions $I$ is crucial in resolving the entire optimization, because $I$ is the knob governing the representations in-between $X, A$ and $C$.

Our updating scheme shares spirit with the standard Gibbs sampling and MH-MCMC method for the inference in probabilistic graphical models. The updating scheme takes the following three stages:

Stage 1 activity-aware tracking (§III-C), where individual target trajectories and activity labels are updated using:

$$\arg \max_{X, A} \log f_1(X, D) + \log \varphi_1(X, A). \quad (3)$$

Stage 2 joint interaction recognition and occlusion recovery (§III-D), where the interaction labels together with the target trajectories and activities are updated using:

$$\arg \max_{X', A^+, I^+} \log \varphi_2(A^+, I) + \log \varphi_3(I, X^+). \quad (4)$$


**TABLE II**

| symbol | description |
|---|---|
| $\mathcal{H}$ | hypergraph $\mathcal{H} = (V, E, W)$ |
| $\mathcal{H}_T$ | tracking hypergraph $\mathcal{H}_T = (V_T, E_T, W_T)$ |
| $\mathcal{H}_R$ | activity recognition hypergraph $\mathcal{H}_R = (V_R, E_R, W_R)$ |
| $V$ | the vertex set of a hypergraph |
| $E$ | the hyperedge set of a hypergraph |
| $W$ | the hyperedge weights of a hypergraph |
| $W_a$ | the appearance hyperedge weight, working with control parameter $\lambda_a = 30$ |
| $W_d$ | the facing-direction hyperedge weight, working with control parameter $\lambda_d = 1$ |
| $W_g$ | the geometric similarity hyperedge weight, working with control parameter $\lambda_g = 0.5$ |
| $m$ | the hyperedge degree, i.e., the number of incident vertices of the hyperedge |
| $e^m$ | a $m$-degree hyperedge, $e^m = \{v_1^m, \ldots, v_m^m\}$ |
| $C$ | a hyperedge cluster, which is a vertex set with interconnected hyperedges |
| $n_i$ | number of vertices in a hypergraph cluster $C$, $\kappa = |C|$ |
| $E_C$ | the set of all incident hyperedges of a cluster $C$ |
| $\Psi$ | weighting function operated on a hypergraph cluster $C$ |
| $\gamma$ | the indicator vector to denote the vertex selection from $V \subset \mathcal{H}$ to be included in $C$ |
| $\epsilon$ | $\epsilon = \frac{1}{\lambda}$ used in weight normalization |
| $\delta$ | $\delta = \frac{1}{2\lambda_g}$ used in weight normalization |
| $p_{ij}$ | image coordinate vector between two positions at $i$ and $j$ |
| $\mathcal{H}_C$ | a sub-hypergraph indexed by $\beta$, i.e., $\mathcal{H}_\beta$ |
| $E_\beta$ | the hyperedges of the sub-hypergraph $\mathcal{H}_\beta$ corresponding to the $\beta$-th interaction class |
| $W_\beta$ | the hyperedge weights of the sub-hypergraph $\mathcal{H}_\beta$ corresponding to the $\beta$-th interaction class |
| $\mathcal{G}$ | graph $\mathcal{G} = (V, E, W)$ |
| $V$ | the vertex set of a graph; $V$ is associated with $X'$ in this paper |
| $E$ | the edge set of a graph |
| $W$ | the edge weights of a graph |
| $e_{ij}$ | a graph edge connecting two vertices $v_i$ and $v_j$ |
| $p_{corr}$ | the correlation between the activities of two targets $x_i$ and $x_j$ used to calculate weight $W(e_{ij})$ |
| $q$ | a function to calculate the correlation between the activities of two targets |
| $d$ | Euclidean distance between two targets in the image coordinate. |
| $\phi_{ij}$ | the angle between the facing direction of $x_i$ and the relative vector from $x_i$ to $x_j$. |
| $G_s$ | sparse graph by discarding edges with small weights from $\mathcal{G}$ |

**Stage 3 collective activity recognition** ($\S$ III-F), where the collective activity labels are updated using:

$$
C^* = \arg \max_{C} \log f_3(X^t, A^t, I^t, C).
$$

We will show in $\S$ III-C and $\S$ III-D that we model high-order correlations among $X$, $A$ and $I$ using two respective hypergraphs. The clique potentials $\phi_1, \phi_2, \phi_3$ in Stage 1 and Stage 2 can be derived as the optimization of maximal weight search over the two hypergraphs, in order to infer $X, A, I$. Stage 3 infers $C$ using a probabilistic formulation based on the inferred $X, A, I$.

Notations for video activities, problem formulation and tracking are summarized in Table I where graph and hypergraph related notations are summarized in Table II.

**B. Cohesive Cluster Search on the Hypergraph**

We define an undirected hypergraph $\mathcal{H} = (V, E, W)$, where $V = \{v_r\}$ denotes the vertex set of $\mathcal{H}$, where $r$ denotes vertex index. An undirected hypergraph with $m$-incident vertices is defined as $e^m = \{v_1^m, \ldots, v_m^m\}$, where $m$ is the degree of the hyperedge. The set of all $m$-degree hyperedges is denoted as $E = \{e^m\}$. The weights of hyperedges are denoted as $W : E \rightarrow \mathbb{R}$, i.e., each hyperedge is associated with a weight.

We use the hypergraphs to represent both (1) the detection-tracklet association for tracking $(X', X)$, and (2) the correlations among individual activities $A$ and pairwise interactions $I$. The joint problem of multi-person tracking (with possible refinements) and group activity recognition can be solved using a standard cohesive cluster search on the hypergraph $[21]$. A cluster $C$ within a hypergraph is a vertex set with interconnected hyperedges. We use $\kappa = |C|$ to denote the number of vertices in $C$, and $E_C$ to denote the set of all incident hyperedges of $C$. A cluster is cohesive if its vertices are interconnected by a large amount of hyperedges with dense weights. Denote $\Psi(\cdot)$ the **weighting function** that measures the weight of a cluster. For a vertex $v_r \in V$, the cohesive cluster search optimization is to determine a large cluster $C(v_r)$ with dense weights:

$$
C(v_r)^* = \arg \max_{C(v_r): \mathcal{C}(v_r) \subset V} \Psi(C(v_r)) \text{ s.t. } C(v_r) \subset \mathcal{V}.
$$

We use indicator vector $y = (y_1, \ldots, y_{|V|})$, $y_p \in \{0, 1\}$ to denote the selection of vertices from $\mathcal{H}$ to be included in $C$:
(T1) Candidate Tracklet Association

(2) Tracklet Linking (Best Candidate Selection)

(T2) Tracklet Linking (Candidate Selection)

(T2.1) Group Structure Estimation

(T2.2) Construction of $H_r$

(T2.3) Tracking Optimization over $H_r$

Fig. 2. Stage 1 activity-aware tracking. Given five targets $x_1', \ldots, x_5'$, and new tracklets $x_1, \ldots, x_5$, step (T1) optimizes the association of candidate tracklets with existing target tracks. Step (T2) determines the best candidate assignments for tracklet linking in three steps. (T2.1) estimates the group structure using graph $G_s$, where the edges represent the correlations of activities between individuals. (T2.2) constructs hypergraph $H_r$ with hyperedges $e_1, \ldots, e_5$ based on the estimated group structure. (T2.3) solves the candidate tracklet linking and infers the possible occlusions in an optimization over $H_r$.

$y_p = 1$ for $v_p \in C$, and $y_p = 0$ otherwise. The selection is constrained such that up to $\kappa$ vertices including $v_i$ are enclosed in $C$, such that $\sum_{p=1}^{V} y_p = \kappa$, and $y_i = 1$.

The design of $\Psi(\cdot)$ affects the resulting cluster $C$ from the search. Typical $\Psi(\cdot)$ can be the total weight of all incident hyperedges. However, direct maximization of the total weight leads to a large cluster that is not necessarily cohesive. Instead, we maximize a normalized weight, which is the total weight divided by the cardinality of all incident hyperedges. This normalization also enables continuous optimization. For $C$ with $\kappa$ vertices and $m$-degree hyperedges, this normalizer is $\kappa^m$. Our weighting function $\Psi(C(v_r))$ is:

$$\sum_{e \in E_S} \frac{W(e^m)}{\kappa^m} = \sum_{v_p, \ldots, v_q \in V} \left( \frac{m}{\kappa} y_{v_p} \ldots \frac{m}{\kappa} y_{v_q} \right).$$

where $p, q$ denote vertex indices.

It is intuitive to enforce that $C$ must contain at least one hyperedge, thus $\kappa$ must be $\geq m$. Let $\delta_p = \frac{y_p}{\kappa}$ and $\epsilon = \frac{1}{\kappa}$. The conditions $\sum_{p=1}^{V} y_p = \kappa$ is then $\sum_{p=1}^{V} \delta_p = 1$. We relax the constraint of $y_p \in \{0, 1\}$ to be $\delta_p \in [0, \epsilon]$, so $\delta$ is a continuous variable for optimization. Eq.(6) is re-written as:

$$\max_{\delta} \sum_{v_p, \ldots, v_q \in V} \left( W(v_p, \ldots, v_q) \cdot \delta_{v_p} \ldots \delta_{v_q} \right)$$

s.t. $\sum_{p=1}^{n} \delta_p = 1$, $\delta_p \in [0, \epsilon]$, $\delta_r = \epsilon$.

We solve Eq.(8) using the pairwise update algorithm in [21].

C. Activity-Aware Tracking

Stage 1 of our method simultaneously recognizes individual activities and links tracklets in the following two steps (see Fig.2 for a schematic overview):

- (T1) Generate candidate tracklets $\bar{X}$ from new detections $D$ that maximizes $\log f_1(X, D)$ in Eq.(3).
- (T2) Link tracklets $X'$ with $\bar{X}$ by maximizing the appearance, motion, and geometric consistencies that maximizes $\log \varphi_1(X, A)$ in Eq.(3).

1 All candidate tracklets and their labels are denoted with a bar $\bar{\cdot}$.
2 We use trajectory prediction based on motion extrapolation in step (R1) of §III-D to determine if the target is still within the scene.
between individual activities using an undirected complete graph $\mathcal{G} = \{V, E, W\}$ with $V = X'$, $\forall x_i', x_j' \in E$. Edge weight $W(e_{ij})$ reflects the correlation between activities $a_i$ and $a_j$ of $x_i'$ and $x_j'$, respectively. We define $p_{corr}$ to reflect the correlation between activities of two targets similar to [23]:

$$p_{corr}(x_i, x_j) = g(d_{ij}, \phi_{ij}, \|v_i\|, \|v_j\|, a_i, a_j, \eta(x_i), \eta(x_j)),$$

(9)

where $d_{ij}$ represents Euclidean distance between the targets. As shown in Fig. 4, $\phi_{ij}$ represents the angle between the facing direction of $x_i$ and the relative vector from $x_i$ to $x_j$, and $v_i$ represents the velocity of $x_i$. For a target $x_i$, if $a_i$ is recognized as “standing”, we use the classifier in [24] to calculate the body orientation $\eta(x_i)$ out of 8 quantizations. Otherwise, $\eta(x_i)$ estimates motion direction from the trajectory. Edge weights of $\mathcal{G}$ are calculated according to Eq. (9) and refined using further grouping cues as in [23]. Fig. 4 visualizes the correlation defined by Eq. (9). The probability is higher on the side of a person than in the front or back, which is an implementation of Hall’s proxemics social norms [25]. We discard edges with weights lower than 0.3 to obtain a sparse graph denoted as $\mathcal{G}_s$ for computation speedup.

**(R2.2) Construct hypergraph $H_e = \{V_e, E_e, W_e\}$ using $\mathcal{G}_s$ to capture the high-order correlations between activities within a group.** A vertex $v_p \in V_e$ represents a hypothesis of linking a tracked target with its candidate tracklet, i.e., $v_p = x_i' \oplus \bar{x}_{i,k}$ where “$\oplus$” represents the association of two tracklets. A $m$-degree hyperedge $e_m \in E_e$ represents the combination of $m$ tracklet linking hypotheses in an assignment.

The linking of tracklets $X'$ with $\bar{X}$ can be considered as an assignment problem with the following two **tracklet assignment constraints**: (i) a target cannot be linked with two or more candidate tracklets, and (ii) a candidate tracklet cannot be linked with two or more targets. We enforce these constraints in the construction of hyperedges in $H_T$. Specifically, $v_p, v_q \in V_e$, where $v_p = x_i' \oplus \bar{x}_{i,k}$ and $v_q = x_j' \oplus \bar{x}_{j,l}$, if and only if $e_{ij} = (x'_i, x'_j) \in \mathcal{G}_s$, $v_p$ and $v_q$ can co-exist in a hyperedge in $H_T$.

We further consider motion and behavior consistencies and their correlations (via $\mathcal{G}_s$) in determining the hyperedge weights. Specifically, we consider three affinities that determine the hyperedge weights: the **appearance ($W_a$)** of each tracklet, the **facing-direction ($W_f$)** and the **geometric similarity ($W_g$)** between tracked targets.

The appearance affinity between a target $x_i'$ and a candidate tracklet $\bar{x}_{i,k}$ is computed using the appearance features of tracklets as [3]:

$$W_a(e^m) = \sum_{x_i' \oplus \bar{x}_{i,k} \in e^m} |x'_i - x_{i,k}|.$$

(10)

We assume that activity states (such as walking direction) do not change abruptly in-between small linked tracklets. In other words, difference between facing directions of two targets should be small for linked tracklets:

$$W_f(e^m) = \sum_{x_i' \oplus \bar{x}_{i,k} \in e^m} \cos(\eta(x'_i), \eta(\bar{x}_{i,k})).$$

(11)

Our method aims to run on surveillance videos without calibration. To ensure smooth tracking, we use a geometric affinity term $W_g$ to ensure that relative angles between two targets does not change abruptly:

$$W_g(e^m) = \sum_{x_i' \oplus \bar{x}_{i,k} \in e^m} \sum_{x_j' \oplus \bar{x}_{j,l} \in e^m} \cos(p'_{ij}, \bar{p}_{ij}),$$

(12)

where $p'_{ij}$ and $\bar{p}_{ij}$ represent the relative image coordinate vectors between tracked targets and candidate tracklets. Final affinity value of a hyperedge $e^m$ is computed by $W(e^m) = \lambda_a W_a(e^m) + \lambda_f W_f(e^m) + \lambda_g W_g(e^m)$, where $\lambda_a$, $\lambda_f$, $\lambda_g$ are set as $\lambda_a = 30$, $\lambda_f = 1$, $\lambda_g = 0.5$. 

**Fig. 3. Stage 2 joint interaction recognition and occlusion recovery** in a road-crossing scenario, where $x_1$, $x_2$, $x_3$ and $x'_4$ are walking side-by-side across a road, while $x_4$, $x_0$, $x_T$ are standing side-by-side waiting. Step (R1) considers the linking of the occluded target $x'_4$ to three hypothetical tracklets $\bar{x}_{4,1}$, $\bar{x}_{4,2}$, $\bar{x}_{4,3}$. Step (R2) constructs hypergraph $H_R$ for the inference in two steps. (R2.1) evaluates each pairwise interaction by calculating a confidence score, where wrongly assigned labels are depicted in red. (R2.2) constructs hyperedges based on the recognized pairwise interactions, where each hyperedge characterizes the likelihood of a pairwise interaction. Step (R3) optimizes the inference over $H_R$ to jointly recognize interaction labels and resolve the tracklet linking and occlusion recovery.

**Fig. 4. Social group affinity between a pair of individuals is calculated based on:** (a) distance, angle, and motion (velocity magnitude & direction), (b) visualizes such a measure at (0, 0) with direction vector (1, 1) arrow in a color map depicting the probability kernel between 0 (blue) and 1 (red).
Our approach is motivated from the observation that pairwise interactions $I$ within a group can provide rich contextual cues to recognize the activities (as in Fig 1) and recover possible occlusions. Stage 2 of our method jointly resolves the two problems of (1) recognizing pairwise interactions $I$ and (2) occlusion recovery to improve tracking. We again use a hypergraph representation to explore the high-order correlations among the interactions $I$, such that a similar cluster search scheme can be applied for optimization. Specifically, we construct the (activity) recognition hypergraph $(\mathcal{H}_R)$ based on the inferred target locations $X^*$ and individual activities $A^*$. The optimization over $\mathcal{H}_R$ maximizes the clique potential function $\log \varphi(A^*, I) \varphi_3(I, X^*)$ in Eq. (3), as $\varphi_2(A^*, I) \varphi_3(I, X^*) \sim \sum_{\mathcal{C} \subseteq \mathcal{R}^m} \Psi(C^m_\alpha)$, where $C^m_\alpha$ represents a cohesive cluster obtained from $\mathcal{H}_R$.

Stage 2 of our method jointly recognizes $I$ and recovery occlusions in the following three main steps (see Fig 3 for a schematic overview):

- **(R1) Generate hypothetical tracklets $\hat{X}$ for occlusion recovery from given existing $X'$ and $A'$.
- **(R2) Construct hypergraph $\mathcal{H}_R$ based on $X^*, A^*, \hat{X}$ to infer high-order correlations among their pairwise interactions $I$.
- **(R3) Optimize recognition and recovery over $\mathcal{H}_R$ to simultaneously recognize interaction $I$ and link occluded targets with suitable hypothetical tracklets.

**R1. Generate hypothetical tracklets $\hat{X}$.** For each possibly occluded target $x'_i \in X'$, we generate a few hypothetical tracklets $\hat{x}_i = \{\hat{x}_{i,1}, \ldots, \hat{x}_{i,h}\}$ based on trajectory predictions, where $h$ is empirically set to 9. For a moving target $x'_i$ with $a'_i = \text{walking}$, we generate $\hat{x}_i$ via motion extrapolation. For a stationary target $x'_i$ with $a'_i = \text{standing}$, we add a small perturbation to $\hat{x}_i$.

**R2. Construct hypergraph $\mathcal{H}_R = \{V_R, E_R, W_R\}$, such that high-order correlations among interactions $X$ and $X$ are captured for the purposes of simultaneous activity recognition and occlusion recovery. Thus, $V_R = X \cup \hat{X}$. Each hyperedge in $E_R$ characterizes the likelihood of a pairwise interaction $i \in I$. For example in Fig 3, $x_1, x_2, x_3$ are connected by 3 hyperedges, which correspond to interactions “WS”, “RS”, “SS”, respectively. See § IV for a complete list of interaction class defined in public datasets [24, 7]. We denote $n_I$ the number of interaction classes.

We calculate the hyperedge weights in $\mathcal{H}_R$ in two steps: **(R2.1) evaluates each pairwise interaction with a confidence score. (R2.2) constructs hyperedges in $\mathcal{H}_R$ using the average score from all involved targets.**

**(R2.1) Recognize pairwise interaction activities.** We calculate a confidence score for each possible pairwise interaction $\hat{1}_{ij}$ between the targets $x_i, x_j$ using a simple effective rule-based probabilistic approach as in [23]. Specifically, the confidence score of $\hat{1}_{ij}$ belonging to the $\beta$-th class is calculated by multiplying the following six component probabilities: distance (ds), group connectivity (gc) calculated in (9), individual activity agreement (aa), distance change type (dc), facing direction (df), and frontness/sidedness (fs):

$$p(\hat{1}_{ij} = \beta | x_i, x_j, a_i, a_j) = p_{ds}(\beta | x_i, a_i, a_j) \cdot p_{gc}(\beta | a_i, a_j) \cdot p_{aa}(\beta | a_i, a_j) \cdot p_{dc}(\beta | x_i, x_j) \cdot p_{df}(\beta | x_i, x_j) \cdot p_{fs}(\beta | x_i, x_j).$$

Detailed formulation of the above component probabilities and formulation are provided in Tables III and IV.

**(R2.2) Construct hyperedges in $\mathcal{H}_R$.** We consider interactions among both real and hypothetical targets during the optimization. We avoid the inclusion of multiple hypothetical tracklets of a target into a hyperedge. For each hyperedge $e^m = \{x^m_1, \ldots, x^m_m\} \in E_R$ for the $\beta$-th interaction class, we calculate the edge weight by averaging the confidence scores of the involved targets:

$$W(e^m) = \frac{1}{m(m-1)} \sum_{i,j} p(\hat{1}_{ij} = \beta | x^m_i, x^m_j, a^m_i, a^m_j).$$

**(R3) Optimize recognition and recovery over $\mathcal{H}_R$ cohesive cluster search on each sub-hypergraph $\mathcal{H}_\beta$ respectively (as described in § III-B). This optimizes the assignment of interaction labels and the linking of probable hypothetical tracklets. Similar to (T2.3), for each vertex $v_r \in \mathcal{H}_R$, we search for candidate cohesive clusters with confidence scores. We ensure that the resulting cohesive clusters are sound and redundancy-free, also not violating the task assignment constraints. Optimization results are used to update $X^*, A^*, I$ into $X^\dagger, A^\dagger, I^\dagger$, respectively as in Eq.(4).
TABLE III
Component probabilities for the pairwise interaction activities. The parameters used in these component probabilities, e.g., the means and standard deviations are calculated from the training dataset.

| Component                        | Probability                                                                 |
|----------------------------------|-----------------------------------------------------------------------------|
| Distance                         | $p_{d_s}(\text{within-effective-range}|x_i,x_j) = \delta(\|d_{ij} - \mu_{ds}\| \leq b), d_{ij} \sim N(\mu_{ds}, \sigma_{ds}), \text{where } N \text{ denotes normal distribution} |
| Group connectivity               | $p_{gc}(\text{connect}|x_i,x_j) \text{, where } \{\text{connect, not-connect}\}$ |
|                                  | $p_{gc}(\text{connect}|x_i,x_j) = g(d_{ij}, \phi_{ij}, ||x_i||, ||x_j||), \text{where } g() \text{ is defined in Eq.}(\ref{eq:connect})$ |
|                                  | $p_{gc}(\text{not-connect}|x_i,x_j) = 1 - g(d_{ij}, \phi_{ij}, ||x_i||, ||x_j||)$ |
| Individual activity agreement    | $p_{aa}(a_i = \Lambda_{A1}, a_j = \Lambda_{A2}) = \sqrt{p(\Lambda_{A1}|x_i) \cdot p(\Lambda_{A2}|x_j)}, \text{where } \Lambda_{A1}, \Lambda_{A2} \in \{\text{standing, walking, running}\}$ |
| Distance-change type             | $p_{dc}(\text{connect}|x_i,x_j) \text{, where } \{\text{decreasing, unchanging, increasing}\}$ |
|                                  | $p_{dc}(\text{decreasing}|x_i,x_j) = 1 - \text{sigmoid}(d_{ij}^0, \mu_{dsu}, \sigma_{ssu}), \text{where } d_{ij}^0 = d_{ij} - d_{ij}'$ |
|                                  | $p_{dc}(\text{unchanging}|x_i,x_j) = \lambda \times \text{sigmoid}(d_{ij}^0, \mu_{dsu}, \sigma_{ssu}) + (1 - \lambda) \times (1 - \text{sigmoid}(d_{ij}^0, \mu_{dsu}, \sigma_{ssu}))$ |
|                                  | if $d_{ij}^0 < \mu_{dsu}, \lambda = 1; if d_{ij}^0 \geq \mu_{dsu}, \lambda = 0; \text{otherwise, } \lambda = 1 - \text{sigmoid}(d_{ij}^0, \mu_{dsu}, \sigma_{ssu})$ |
|                                  | $p_{dc}(\text{increasing}|x_i,x_j) = \text{sigmoid}(d_{ij}^0, \mu_{dsu}, \sigma_{ssu})$ |
| Facing direction                 | $p_{fr}(\text{DR}|x_i,x_j) \in \{0,1\}, \text{where } \{0,1\} \in \{\text{same, opposite, frequent-changing}\}$ |
| Frontness/sideness              | $p_{fs}(\text{frontness}|x_i,x_j) = \max(\text{cost}(p_{ij}, x_i), \text{cost}(p_{ij}, x_j))$ |
|                                  | $p_{fs}(\text{sideness}|x_i,x_j) = 1 - p_{fs}(\text{frontness}|x_i,x_j)$ |

TABLE IV
Probabilistic formulations for the pairwise interaction $p(i_{ij} = \beta)$. We define dancing-together (DT) as a new interaction activity class to deal with the new collective activity “dancing” in the Augmented-CAD.

| Pairwise Interaction $p(i_{ij} = \beta)$ | Associated Collective Activity (C) | Probabilistic Formulation |
|------------------------------------------|-----------------------------------|---------------------------|
| facing-each-other ($\beta = FE$)         | TALKING                           | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{connect}) \cdot p_{aa}(\text{standing, standing}) \cdot p_{dc}(\text{unchanging}) \cdot p_{dc}(\text{opposite}) \cdot p_{fr}(\text{frontness})$ |
| standing-in-a-row ($\beta = SR$)         | QUEUING                           | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{connect}) \cdot p_{aa}(\text{standing, standing}) \cdot p_{dc}(\text{unchanging}) \cdot p_{dc}(\text{same}) \cdot p_{fs}(\text{sideness})$ |
| standing-side-by-side ($\beta = SS$)     | WAITING                           | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{connect}) \cdot p_{aa}(\text{standing, standing}) \cdot p_{dc}(\text{unchanging}) \cdot p_{fr}(\text{frontness})$ |
| dancing-together ($\beta = DT$)          | DANCING                           | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{connect}) \cdot p_{aa}(\text{walking, walking}) \cdot p_{dc}(\text{unchanging}) \cdot p_{fr}(\text{frontness})$ |
| approaching ($\beta = AP$)               | GATHERING                         | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{not-connect}) \cdot p_{aa}(\text{walking, walking}) \cdot p_{dc}(\text{decreasing}) \cdot p_{fr}(\text{frontness})$ |
| walking-in-opposite-directions ($\beta = WO$) leaving ($\beta = LV$) | DISMISSAL                      | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{not-connect}) \cdot p_{aa}(\text{walking, walking}) \cdot p_{dc}(\text{increasing}) \cdot p_{fr}(\text{frontness})$ |
| walking-side-by-side ($\beta = WS$)      | CROSSING WALKING (TOGETHER)      | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{connect}) \cdot p_{aa}(\text{walking, walking}) \cdot p_{dc}(\text{unchanging}) \cdot p_{fr}(\text{frontness})$ |
| running-side-by-side ($\beta = RS$)      | JOGGING                           | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{connect}) \cdot p_{aa}(\text{running, running}) \cdot p_{dc}(\text{unchanging}) \cdot p_{fr}(\text{frontness})$ |
| running-one-after-the-other ($\beta = RR$)| CHASING                           | $p_{dc}(\text{within-effective-range}) \cdot p_{gc}(\text{connect}) \cdot p_{aa}(\text{running, running}) \cdot p_{dc}(\text{unchanging}) \cdot p_{fr}(\text{frontness})$ |

E. Collective Activity Recognition

Stage 3 of our method infers the collective activities $C^*$ for each individual in a group, based on an intuition that collective activity is characterized by pairwise interactions $I$ indexed by $\beta$ within the group. Fig.\ref{fig:example} illustrates several examples, and Table \ref{tab:collective} shows the cohesive pairwise interaction for each collective activity class. For each target $x_i$ within a group, we infer the most probable collective activity. The term $\log f_3(X^1, A^1, I^*, C)$ in Eq.(\ref{eq:collective}) can be maximized based on a probabilistic formulation similar to \ref{eq:pairwise}. Consider the $\beta$-th interaction for the $c$-th collective activity, $p(c|x_i)$ is calculated using:

$$
    p(c|x_i) = 1 - \prod_{v_j} \left(1 - p(i_{ij} = \beta|x_i,x_j)\right),
$$

where $p(i_{ij} = \beta|x_i,x_j)$ is obtained in Eq.(\ref{eq:collective}) after the optimization in (\ref{eq:collective}). The collective activity of $x_i$ is determined by $\arg\max_{c} p(c|x_i)$. We use the collective activity involving most participants as the label of the scene, to comply with the practice in major datasets \ref{tab:datasets}. If there are insufficient targets for interactive or collective activities (e.g. people leaving the scene), we keep existing labels for a short while.

IV. Experiments

1) Implementation: We implement our method in C++. Experiments are conducted on a machine with a i7-4800MQ CPU (2.8GHz) and 16GB RAM. We use the state-of-the-art person detections \ref{tab:detection} as input, and employ deep re-identification features \ref{tab:reid} as the appearance features for tracking. We set hyperedge degree $m = 3$ to balance the performance and speed. The whole pipeline runs in nearly real-time at approximately 20 FPS (not including the detection time). Note that input detectors can be executed in parallel for real-world applications.

2) Datasets: We perform evaluation on three popular collective activity recognition datasets, which are termed CAD \ref{tab:datasets}, Augmented-CAD \ref{tab:datasets}, and New-CAD \ref{tab:datasets}. Pedestrians in CAD and New-CAD are annotated with target IDs that can be used as ground truth for tracking evaluation.

CAD \ref{tab:datasets} comprises 44 video clips with annotations for $n_{c} = 5$ collective activities (CROSSING, WAITING, QUEUING, WALKING, TALKING), $n_{v} = 8$ pairwise interactions: approaching (AP), leaving (LV), passing-by (PB), facing-each-other (FE), walking-side-by-side (WS), standing-in-a-row (SR), walking-in-opposite-directions (WO), and running-side-by-side (RS).


### TABLE V

Activity recognition evaluation results in terms of accuracy and comparison with state-of-the-art methods. The CAD results are split into two tables: one for comparing methods only producing collective activity results, and the other for all three activities.

| Method | Collective | Interaction | Individual |
|--------|------------|-------------|------------|
| OCA   | MCA        | OCA         | MCA        | OCA        | MCA        |
| LCC 16 | -          | 70.9        | -          | -          | -          |
| CPT 15 | -          | 75.1        | -          | -          | -          |
| FM 6   | -          | 70.9        | -          | -          | -          |
| LOG 12 | 79.7       | 78.4        | -          | -          | -          |
| MCTS 19 | -          | 88.9        | -          | -          | -          |
| LLC 14 | -          | 75.1        | -          | -          | -          |
| HIRF 18 | -          | 92.0        | -          | -          | -          |
| DSM 13 | -          | 80.6        | -          | -          | -          |
| CK 14  | 83.4       | 81.9        | -          | -          | -          |
| HDT 14 | -          | 81.5        | -          | -          | -          |
| SIE 19 | -          | 81.2        | -          | -          | -          |
| RMI 20 | -          | 89.4        | -          | -          | -          |
| UTR-S [1, 8] | 79.0       | 79.6       | 56.2       | 50.8       | -          |
| UTR-R [2, 8] | 79.4       | 80.2       | 55.5       | 56.6       | -          |
| Baseline | 87.8      | 88.0       | 65.4       | 48.4       | 87.3       |
| Ours w/o H_T | 91.8      | 91.8       | 74.3       | 55.6       | 87.7       |
| Ours w/o H_R | 88.0      | 87.8       | 71.4       | 56.5       | 87.3       |
| Ours w/o H_T, H_R | 87.9      | 87.6       | 67.3       | 53.7       | 87.4       |
| Ours | 92.5       | 92.4       | 78.1       | 57.6       | 87.4       |

| Method | Collective | Interaction | Individual |
|--------|------------|-------------|------------|
| OCA   | MCA        | OCA         | MCA        | OCA        | MCA        |
| LCC 16 | -          | 82.0        | -          | -          | -          |
| CPT 15 | -          | 85.8        | -          | -          | -          |
| FM 6   | -          | 83.7        | -          | -          | -          |
| LOG 12 | -          | 90.1        | -          | -          | -          |
| MCTS 19 | -          | 90.2        | -          | -          | -          |
| LLC 14 | -          | 89.4        | -          | -          | -          |
| HIRF 18 | -          | 86.9        | -          | -          | -          |
| DSM 13 | -          | 85.2        | -          | -          | -          |
| CK 14  | -          | 84.9        | -          | -          | -          |
| Ours | 95.1       | 94.3        | -          | -          | -          |

### TABLE VI

Tracking evaluation and comparison with state-of-the-art methods. ↑ and ↓ represent “the-higher-the-better” and “the-lower-the-better”, respectively. Bold highlights best results.

| Dataset | Method | Recall ↑ | Prec ↑ | FAR ↓ | MT (%) ↑ | ML (%) ↓ | FP ↓ | FN ↓ | IDS | FM ↓ | MOA ↑ | MOTP ↑ |
|---------|--------|----------|--------|-------|----------|----------|------|------|-----|------|-------|-------|

**CAD**

| Method | Recall ↑ | Prec ↑ | FAR ↓ | MT (%) ↑ | ML (%) ↓ | FP ↓ | FN ↓ | IDS | FM ↓ | MOA ↑ | MOTP ↑ |
|--------|----------|--------|-------|----------|----------|------|------|-----|------|-------|-------|
| H^T [24] | 84.3 | 83.8 | 0.90 | 74.5 | 5.4 | 23195 | 22368 | 474 | 722 | 67.6 | 67.5 |
| JPD [25] | 84.1 | 83.5 | 0.79 | 70.1 | 5.4 | 20339 | 22660 | 480 | 701 | 69.6 | 66.7 |
| DCEM [26] | 81.4 | 84.3 | 0.53 | 72.1 | 5.4 | 13617 | 69127 | 801 | 1025 | 41.2 | 62.5 |
| Ours w/o H_T | 84.0 | 86.3 | 0.74 | 74.0 | 5.2 | 18091 | 22717 | 355 | 623 | 70.4 | 66.7 |
| Ours w/o H_R | 84.0 | 85.1 | 0.82 | 74.0 | 5.2 | 21002 | 22684 | 319 | 638 | 69.0 | 67.6 |
| Ours w/o H_T, H_R | 84.0 | 84.8 | 0.83 | 74.7 | 5.2 | 21362 | 22717 | 360 | 630 | 68.7 | 67.6 |
| Our RMI [20] | 84.1 | 86.6 | 0.72 | 74.9 | 5.2 | 18061 | 22691 | 287 | 619 | 70.3 | 67.6 |

| Method | Recall ↑ | Prec ↑ | FAR ↓ | MT (%) ↑ | ML (%) ↓ | FP ↓ | FN ↓ | IDS | FM ↓ | MOA ↑ | MOTP ↑ |
|--------|----------|--------|-------|----------|----------|------|------|-----|------|-------|-------|
| H^T [24] | 87.4 | 88.3 | 0.37 | 81.5 | 1.6 | 7883 | 8572 | 117 | 232 | 55.6 | 64.7 |
| JPD [25] | 87.6 | 88.6 | 0.36 | 82.6 | 1.4 | 7660 | 8413 | 65 | 198 | 76.3 | 62.3 |
| DCEM [26] | 68.5 | 89.5 | 0.28 | 41.9 | 1.3 | 6051 | 21458 | 220 | 283 | 59.3 | 62.4 |
| Ours w/o H_T | 87.9 | 89.0 | 0.35 | 83.9 | 0.8 | 7331 | 8594 | 52 | 271 | 76.5 | 64.7 |
| Ours w/o H_R | 87.9 | 88.9 | 0.35 | 83.9 | 0.8 | 7411 | 8226 | 63 | 198 | 76.9 | 64.7 |
| Ours w/o H_T, H_R | 88.1 | 88.1 | 0.36 | 83.9 | 0.8 | 7726 | 8096 | 59 | 202 | 76.1 | 64.7 |
| Ours RMI [20] | 88.2 | 88.7 | 0.36 | 84.7 | 0.8 | 7808 | 8308 | 60 | 202 | 76.9 | 64.7 |

**New CAD**

### 3) Experimental Setup:

For evaluating activity recognition, we follow common protocols as in [2] for CAD and New-CAD, and protocol of [16] for Augmented-CAD. For evaluating tracking, we ensure fair comparison by running all tracking code using identical input detections.

### 4) Evaluation Metrics:

For activity recognition, we adopt the metrics used in [7], i.e., overall classification accuracy (OCA) and mean-per-class-accuracy (MCA) as in Table V. Note that the match-error-correction-rate used in [7] only reflects tracking fragmentation and identity switch. Instead, we adopt the more widely-used CLEAR MOT as tracking metrics to provide further insights for analysis.

### 5) Compared Methods:

We compare our method with 13 state-of-the-art activity recognition methods [18, 19, 7, 8, 9, 13, 14, 16, 6, 20] and a few baseline methods created by ourselves. These baseline methods accept the tracking results of [22] as input, and recognize

Standing-side-by-side (SS), no-interaction (NA), and n_A=2 individual activities: standing and walking.

Augmented-CAD [16] is created by augmenting the CAD dataset. Collective activity WALKING is removed due to its ambiguities in definition, and 2 new collective activities DANCING, and JOGGING are included. For the newly introduced video clips, there are no annotations for interaction activities, individual activities, nor target identities.

New-CAD [7] comprises 33 video clips with annotations for n_C=6 collective activities: GATHERING, TALKING, DISMISSAL, WALKING-TOGETHER, CHASING, QUEUING, n_I=9 pairwise interactions: approaching (AP), walking-in-opposite-directions (WO), facing-each-other (FE), standing-in-a-row (SR), walking-side-by-side (WS), walking-one-after-the-other (WR), running-side-by-side (RS), running-one-after-the-other (RR), no-interaction (NA), and n_A=3 individual activities: standing, walking, running.
Fig. 5. Recognized collective activity examples in the CAD dataset [24]: (a,b) CROSSING, (c,d) WAITING, and (e,f) QUEUING. Two individual activities walking (W) and standing (S) are detected on each target, while \( n_I = 8 \) possible pairwise interactions: approaching (AP), leaving (LV), passing-by (PB), facing-each-other (FE), walking-side-by-side (WS), standing-in-a-row (SR), standing-side-by-side (SS), no-interaction (NA) are detected among the pairs of targets. A top-down view of each scene is illustrated on the right.

A. Results and Analysis

1) Activity Recognition: Table VI shows the evaluation results in terms of accuracy of our method and other state-of-the-art methods on the three activity recognition datasets. Part of our recognition results are shown in Fig. 5 and Fig. 6. The efficacy of our method for individual activity recognition is demonstrated by the high accuracy score of approximately 90\%. Our method outperforms the state-of-the-art methods by a noticeable margin for collective activity recognition. The performance of our method is significantly better than [7], [8], which are, to the best of our knowledge, the only works that evaluate pairwise interactions. Our method also shows positive correlations between the recognition accuracy of \( C \) and \( I \). Overall, WALKING is difficult to recognize due to its ambiguity w.r.t. CROSSING. Nonetheless our method can still recognize it well due to our probabilistic formulation.

2) Tracking: Table VI shows the comparison of our method and other state-of-the-art tracking methods on the CAD and New-CAD. On the CAD dataset, our method achieves the best performance in most measures, e.g., MOTA, fragmentation (FM), identity switch (IDs), false positives (FP), mostly tracked targets (MT), precision (Prcn), and recall (Rcll). This is due to the incorporation of the high-order correlations in \( \mathcal{H}_T \) and \( \mathcal{H}_R \). Specifically, we use \( \mathcal{H}_T \) to model high-order correlations of individual activities, which improves the tracklet association. The use of \( \mathcal{H}_R \) to model high-order correlations of interaction activities improves occlusion recovery. This is further confirmed that after replacing \( \mathcal{H}_T \) or \( \mathcal{H}_R \) with ordinary graphs, the tracking performance decreases in most measures. New-CAD is less challenging than CAD in terms of tracking, because there are fewer occlusions and crossing occasions. Thus many compared methods yield performances closer toward saturation. However, our method still achieves the highest score in several measures, i.e., MOTA, mostly tracked targets (MT), and recall (Rcll). Both [22] and our method rely on the cohesive cluster search on the hypergraph, but our method consistently outperforms [22] by a significant margin because of (i) the modeling of high-order correlations of individual activities, and (ii) successful occlusion recovery. We visualize the tracking result comparisons for several sequences in Fig. 7. For each sequence, we show two frames across a long period of time, where the subjects have experienced heavy occlusions. It is clear that our method (especially the one with hypergraph optimizers) produces the fewest ID changes, thus it is more robust than the competing methods for occlusion handling.
V. Conclusion

We present a novel multi-stage framework for solving the joint tasks of multi-person tracking and group activity recognition. This approach can effectively address not only the where and when problem by visual tracking, but also the who and what problem by recognition. By explicit modeling of correlations among individual activities, pairwise interactions, and collective activities using hypergraphs, we can effectively improve recognition and tracking. Our method can track targets with occlusion recovery, identify correlated pairwise interactions, and recognize group collective activities. Experimental evaluations demonstrate that our method outperforms state-of-the-art methods in both tasks of tracking and activity recognition. Our method runs in nearly real-time (not counting input detections), and is applicable to a variety of real-world applications including video surveillance and situational awareness. Implementation code will be released upon the publication of this work.

REFERENCES

[1] A. Vinciarelli, M. Pantic, and H. Bourlard, “Social signal processing: Survey of an emerging domain,” Image Vision Comput., vol. 27, no. 12, pp. 1743–1759, 2009.

[2] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” in European Conference on Computer Vision, 2016, pp. 354–370.

[3] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, “POI: multiple object tracking with high performance detection and appearance feature,” in European Conference on Computer Vision Workshops, 2016, pp. 36–42.

[4] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, “A hierarchical deep temporal model for group activity recognition,” in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1971–1980.

[5] S. Khamis, V. I. Morariu, and L. S. Davis, “Combining per-frame and per-track cues for multi-person action recognition,” in European Conference on Computer Vision, 2012, pp. 116–129.

[6] ——, “A flow model for joint action recognition and identity maintenance,” in IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1218–1225.

[7] W. Choi and S. Savarese, “A unified framework for multi-target tracking and collective activity recognition,” in ECCV, 2012, pp. 215–230.

[8] ——, “Understanding collective activities of people from videos,” IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 36, no. 6, pp. 1242–1257, 2014.

[9] Z. Deng, A. Vahdat, H. Hu, and G. Mori, “Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition,” in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4772–4781.

[10] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis: A review,” ACM Computing Surveys, vol. 43, no. 3, pp. 16:1–16:43, 2011.

[11] W. Luo, X. Zhao, and T. Kim, “Multiple object tracking: A review,” CoRR, vol. abs/1409.7618, 2014.

[12] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, “Discriminative latent models for recognizing contextual group activities,” IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 34, no. 8, pp. 1549–1562, 2012.

[13] Z. Deng, M. Zhai, L. Chen, Y. Liu, S. Muralidharan, J. M. Roshkhar, and G. Mori, “Deep structured models for group activity recognition,” in British Machine Vision Conference, 2015, pp. 179.1–179.12.

[14] H. Hajimirsadeghi, W. Yan, A. Vahdat, and G. Mori, “Visual recognition by counting instances: A multi-instance cardinality potential kernel,” in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2596–2605.

[15] T. M. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, “Social scene understanding: End-to-end multi-person action localization and collective activity recognition,” pp. 4315–4324, 2017.

[16] W. Choi, K. Shahid, and S. Savarese, “Learning latent activity models for recognizing contextual group activities,” in British Machine Vision Conference, 2015, pp. 179.1–179.12.

[17] B. Antic and B. Ommer, “Learning latent constituents for recognition of group activities in video,” in European Conference on Computer Vision, 2014, pp. 33–47.

[18] M. R. Amer, P. Lei, and S. Todorovic, “HiRF: Hierarchical random field for collective activity recognition in videos,” in European Conference on Computer Vision, 2014, pp. 572–585.

[19] M. R. Amer, S. Todorovic, A. Fern, and S. Zhu, “Monte carlo tree search for scheduling activity recognition,” in IEEE International Conference on Computer Vision, 2013, pp. 1353–1360.
[20] M. Wang, B. Ni, and X. Yang, “Recurrent modeling of interaction context for collective activity recognition,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3048–3056.

[21] H. Liu, L. J. Latecki, and S. Yan, “Robust clustering as ensembles of affinity relations,” in Advances in Neural Information Processing Systems, 2010, pp. 1414–1422.

[22] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li, “Multiple target tracking based on undirected hierarchical relation hypergraph,” in IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1282–1289.

[23] M. Chang, N. Krahnstoever, and W. Ge, “Probabilistic group-level motion analysis and scenario recognition,” in IEEE International Conference on Computer Vision, 2011, pp. 747–754.

[24] W. Choi, K. Shahid, and S. Savarese, “What are they doing?: Collective activity classification using spatio-temporal relationship among people,” in ICCV Workshops, 2009, pp. 1282–1289.

[25] E. T. Hall, The Hidden Dimension. Anchor Books, 1966.

[26] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. R. Dick, and I. D. Reid, “Joint probabilistic data association revisited,” in IEEE International Conference on Computer Vision, 2015, pp. 3047–3055.

[27] A. Milan, K. Schindler, and S. Roth, “Multi-target tracking by discrete-continuous energy minimization,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 10, pp. 2054–2068, 2016.
Fig. 7. Example tracking results. For each sequence, we show two frames across a long period of time, where the subjects have experienced heavy occlusions. The number of target ID changes are shown in each experiment. Colorful diamonds visualize the corresponding target ID changes.