Understanding Collective Activities of People from Videos

Wongun Choi and Silvio Savarese

Abstract—This paper presents a principled framework for analyzing collective activities at different levels of semantic granularity from videos. Our framework is capable of jointly tracking multiple individuals, recognizing activities performed by individuals in isolation (i.e., atomic activities such as walking or standing), recognizing the interactions between pairs of individuals (i.e., interaction activities) as well as understanding the activities of group of individuals (i.e., collective activities). A key property of our work is that it can coherently combine bottom-up information stemming from detections or fragments of tracks (or tracklets) with top-down evidence. Top-down evidence is provided by a newly proposed descriptor that captures the coherent behavior of groups of individuals in a spatial-temporal neighborhood of the sequence. Top-down evidence provides contextual information for establishing accurate associations between detections or tracklets across frames and, thus, for obtaining more robust tracking results. Bottom-up evidence percolates upwards so as to automatically infer collective activity labels. Experimental results on two challenging data sets demonstrate our theoretical claims and indicate that our model achieves enhances tracking results and the best collective classification results to date.

Index Terms—Collective activity recognition, tracking, tracklet association

1 Introduction

A common paradigm in activity recognition research is to analyze the actions (e.g., *walking*, *jogging*, *dancing*) performed by single individuals in isolation (which we refer as to *atomic activities*) [26], [29], [36]. These are characterized by looking at the behavior of individuals independently of what other individuals are doing in the surrounding scene. In many application scenarios (such as surveillance, video analysis and retrieval), however, it is critical to analyze the collective behavior of individuals (which we refer as to *collective activities*) as well as the way such individuals interact [7], [21], [35]. For instance, the collective activity *gathering* involves multiple individuals walking (an atomic activity), looking and facing each other (FE) (an interaction) and/or moving in a coherent spatial temporal structure toward a certain spatial location (Fig. 1).

Reasoning about collective activities, interactions, and atomic activities, is a very challenging problem. Such an activity recognition system may require that all of (or some of) the following tasks are performed reliably and accurately: i) identifying stable and coherent trajectories of each individual (tracks); ii) Estimating each individual's properties such as human pose and their atomic actions; iii) Discovering the interaction between pairs of individuals; iv) Recognizing the collective activity present in the scene. As shown in [12], [41], tracking multiple individuals at the same time (as well as estimating relevant properties) is extremely difficult because of self-occlusions, detection faults, illumination changes, camera shake or movement, etc. This often leads to fragmented trajectories (*tracklets*) which are not descriptive enough to enable the construction of reliable interaction models. Moreover, assigning atomic activity labels to individuals that coexist in close proximity can be very problematic. This problem has been mostly ignored in the literature.

In this paper we explore the intuition that contextual information provided by the collective behavior of multiple interacting individuals can make the tracking and recognition process more accurate and robust than if these problems are solved in isolation. For instance, if one knows that the people in the scene are *gathering* (Fig. 1), one can infer that a number of people should be *approaching* (interaction), moving toward a point of convergence, facing each other while performing a walking action. This gives strong constraints on each individual spatial-temporal trajectory, which enable the construction of more accurate tracks. Contextual information about collective behavior is provided by a newly proposed descriptor which we have called crowd context. In turn, if better trajectories are obtained, the interaction and the collective behavior can be estimated more accurately. Following this intuition, we argue that track association, atomic activity recognition, and collective activity recognition must be performed in a coherent fashion, so that each component can help the other.

2 Related Work

Target tracking is a well studied problems in computer vision, but it is far from being solved. In challenging scenes such as that in Fig. 13, tracks are not complete, but are fragmented into tracklets. It is the task of the tracker to associate tracklets in order to assemble complete tracks. Tracks are often fragmented due to occlusions, ambiguities in the appearance properties of the

- W. Choi is with NEC Laboratories, E-mail: wongun@nec-labs.com.
- S. Savarese is with the Department of Computer Science, Stanford University, 353 Serra Mall, Gates Building, Stanford, CA 94305-9020. E-mail: ssilvio@stanford.edu.

Manuscript received 20 Jan. 2013; revised 27 Sept. 2013; accepted 20 Oct. 2013. Date of publication 3 Nov. 2013; date of current version 12 May 2014.
Recommended for acceptance by G. Mori.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2013.220

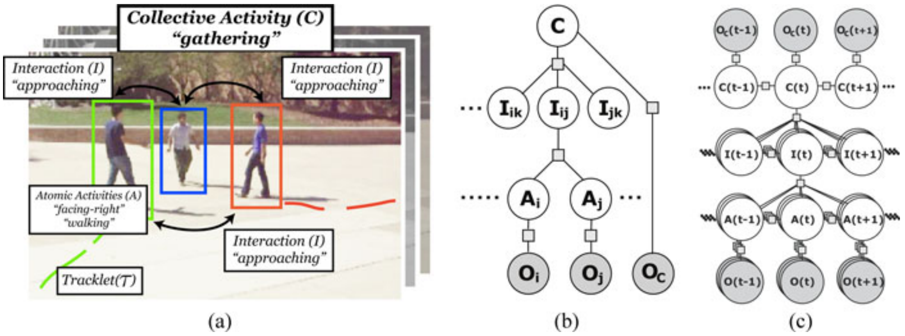


Fig. 1. In this paper, we aim at jointly and robustly tracking multiple targets and recognizing the activities that such targets are performing. (a): The collective activity *“gathering”* is characterized as a collection of interactions (such as *“approaching” (AP)*) between individuals. Each interaction is described by pairs of atomic activities (e.g., *“facing-right”* and *“facing-left”*). Each atomic activity is associated with a spatial-temporal trajectory (tracklet τ). We advocate that high level activity understanding helps obtain more stable target trajectories. Likewise, robust trajectories enable more accurate activity understanding. (b): The hierarchical relationship between atomic activities (A), interactions (I), and collective activity (C) in one time stamp is shown as a factor graph. Squares and circles represent the potential functions and variables, respectively. Observations are the tracklets associated with each individual along with their appearance properties O_i as well as crowd context descriptor O_c [7], [8] (Section 3.1). (c): A collective activity at each time stamp is represented as a collection of interactions within a temporal window. An interaction is described by a pair of atomic activities within specified temporal window (Section 3.2). Non-shaded nodes are associated with variables that need to be estimated and shaded nodes are associated with observations.

targets and sharp camera movements. Recent algorithms address this through the use of detection responses [12], [41], and the idea of contextualizing adjacent tracks using pairwise interaction models [5], [19], [23], [30], [37], [42]. The interaction models, however, are typically limited to a few hand-designed interactions, such as attraction and repulsion. Methods such as [33] leverage the consistency and physical behavior of flows of large crowds of individuals, but do not attempt to associate tracklets or understand the actions of individuals. Zhang et al. [44] and Pirsiavash et al. [31] formulate the problem of multi-target tracking into a min-cost flow network based on linear/dynamic programming. Although both methods model interactions between people, they still rely on heuristics to guide the association process via higher level semantics.

A number of methods have recently been proposed for action recognition by extracting sparse features [11], correlated features [36], discovering hidden topic models [29], or feature mining [26]. These works consider only a single person, and do not benefit from the contextual information available from recognizing interactions and activities. Ryoo and Aggarwal [34] model the pairwise interactions between people, but their representation is limited to using local motion features. Several works address the recognition of group activities in sport events such as in baseball or football games by learning a storyline model [14], reasoning the social role of individual players [20], modelling the trajectories of people with Bayesian networks [16], temporal manifold structures [25], and non-stationary kernel hidden Markov models [39]. All these approaches, however, assume that the trajectories are available (known). Recently, Ni et al. [28] recognize group activities by considering local causality information from each track, each pair of tracks, and groups of tracks. In our own work in [7], we classify collective activities by extracting descriptors from people and the surrounding area, and in [8] we extend it by learning the structure of the descriptor from data. Ryoo and Aggarwal [35] model a group activity as a stochastic collection of individual activities. None of these works exploit the contextual information provided by collective activities to help identify targets or classify atomic activities. Lan et al. [21] and Amer et al. [1] use a hierarchical model to jointly classify the collective activities of all people in a scene, but they are restricted to modelling contextual information in a single frame, without seeking to solve the track identification problem. A number of works [9], [18], [27], [32], [45] has been proposed to recognize group behavior in videos, but they do not seek to characterize the behavior of each individual.

Our contributions are five-fold: (i) we propose a graphical model that merges for the first time the problems of collective activity recognition and multiple target tracking into a single coherent framework [6]. The model coherently combines bottom-up information stemming from detections or fragments of tracks (tracklets) with top-down evidence (Section 3); (ii) we characterize top-down evidence by using a newly proposed descriptor [7], [8] that captures spatio-temporal relationship among people (Section 4); (iii) we introduce a novel path selection algorithm that leverages target interactions for guiding the process of associating targets (Section 5); (iv) we propose a novel inference procedure based on the branch-and-bound (BB) formulation for solving the target association problem using the contextual information coming from the interaction models (Section 7); (v) we present an extensive quantitative evaluation on challenging data sets, showing superiority to the state-of-the-art (Section 9).

3 Modeling Collective Activity

Our model accomplishes collective activity classification by simultaneously estimating the activity of a group of people (*collective activity* C), the pairwise relationships between individuals (*interactions activities* I), and the specific activities of each individual (*atomic activities* A) given a set of observations O (see Fig. 1). A collective activity describes the overall behavior of a group of more than two people, such as *gathering*, *talking*, and *queuing*. Interaction activities

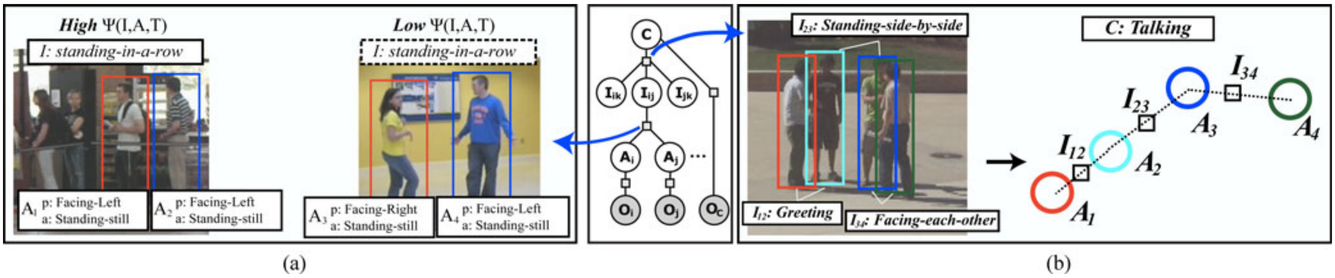

High Ψ(I,A,T)
I: standing-in-a-row
A1 p: Facing-Left a: Standing-still
A2 p: Facing-Left a: Standing-still
Low Ψ(I,A,T)
I: standing-in-a-row
A3 p: Facing-Right a: Standing-still
A4 p: Facing-Left a: Standing-still
C
Iik
Iij
Ijk
Ai
Aj
...
Oi
Oj
Oc
I23: Standing-side-by-side
I12: Greeting
I34: Facing-each-other
C: Talking
I34
I23
I12
A1
A2
A3
A4
(a)
(b)

Fig. 2. (a) Each interaction is represented by a number of atomic activities A_i that are characterized by an action and pose label. For example, with interaction $I =$ *standing-in-a-row (SR)*, it is likely to observe two people with both pose $p =$ *facing-left* and activity $a =$ *standing-still*, whereas it is less likely that one person has $p =$ *facing-left* and the other $p =$ *facing-right*. (b): Collective activity C is represented as a collection of interactions I. For example, with $C =$ *talking* collective activity, it is likely to observe the interaction $I_{34} =$ *facing-each-other*, and $I_{23} =$ *standing-side-by-side (SS)*. The consistency of $C, I_{12}, I_{23}, I_{34}$ generates a high value for $\Psi(C, I)$.

model pairwise relationships between two people which can include *approaching*, *facing-each-other* and *walking-in-opposite-directions (WO)*. The atomic activity collects semantic attributes of a tracklet, such as poses (*facing-front*, *facing-left*) or actions (*walking*, *standing*). Feature observations $O = (O_1, O_2, \dots O_N)$ operate at a low level, using tracklet-based features to inform the estimation of atomic activities. Collective activity estimation is helped by observations O_C, which use features such as spatio-temporal local descriptors (Section 4) to capture the coherent behavior of individuals in a spatial-temporal neighborhood. At this time, we assume that we are given a set of tracklets $\tau_1, \dots, \tau_N$ that denote all targets' spatial location in 2D or 3D. These tracklets can be estimated using methods such as [5]. Tracklet associations are denoted by $T = (T_1, T_2, \dots, T_M)$ and indicate the association of tracklets. We address the estimation of T in Section 5.

The information extracted from tracklet-based observations O enables the recognition of atomic activities A, which assist the recognition of interaction activities I, which are used in the estimation of collective activities C. Concurrently, observations O_c provide evidence for recognizing C, which are used as contextual clues for identifying I, which provide context for estimating A. The bi-directional propagation of information makes it possible to classify C, A, and I robustly, which in turn provides strong constraints for improving tracklet association T. Given a video input, the hierarchical structure of our model is constructed dynamically. An atomic activity A_i is assigned to each tracklet τ_i (and observation O_i), an interaction variable I_{ij} is assigned to every pair of atomic activities that exist at the same time, and all interaction variables within a temporal window are associated with a collective activity C.

3.1 The Model

The graphical model of our framework is shown in Fig. 1. Let $O = (O_1, O_2, \dots O_N)$ be the N observations (visual features within each tracklet) extracted from video V, where observation O_i captures appearance features $s_i(t)$, such as histograms of oriented gradients (HoG [10]), and spatio-temporal features $u_i(t)$, such as a bag of video words (BoV [11]). t corresponds to a specific time stamp within the set of frames $\mathcal{T}_V = (t_1, t_2, \dots, t_Z)$ of video V, where Z is the total number of frames in V. Each observation O_i can be seen as a realization of the underlying atomic activity A_i of an individual. Let $A = (A_1, A_2, \dots, A_N)$. A_i includes pose labels $p_i(t) \in \mathcal{P}$, and action class labels $a_i(t) \in \mathcal{A}$ at time $t \in \mathcal{T}_V$. $\mathcal{P}$ and $\mathcal{A}$ denote the set of all possible pose (e.g., *facing-front*) and action (e.g., *walking*) labels, respectively. $I = (I_{12}, I_{13}, \dots, I_{N-1N})$ denotes the interactions between all possible (coexisting) pairs of A_i and A_j, where each $I_{ij} = (I_{ij}(t_1), \dots I_{ij}(t_Z))$ and $I_{ij}(t) \in \mathcal{I}$ is the set of interaction labels such as *approaching*, *facing-each-other* and *standing-in-a-row*. Similarly, $C = (C(t_1), \dots, C(t_Z))$ and $C(t_i) \in \mathcal{C}$ indicates the collective activity labels of the video V, where $\mathcal{C}$ is the set of collective activity labels, such as *gathering*, *queueing*, and *talking*. In this work, we assume there exists only one collective activity at a certain time frame. Extensions to modelling multiple collective activities will be addressed in the future. T describes the target (tracklet) associations in the scene as explained in Section 3.

We formulate the classification problem in an energy maximization framework [24], with overall energy function $\Psi(C, I, A, O, T)$. The energy function is modelled as the linear product of model weights w and the feature vector ψ :

$$\Psi(C, I, A, O, T) = w^T \psi(C, I, A, O, T) \tag{1}$$

$\psi(C, I, A, O, T)$ is a vector composed of $\psi_1(\cdot), \psi_2(\cdot), \dots, \psi_m(\cdot)$ where each feature element encodes local relationships between variables and w, which is learned discriminatively, is the set of model parameters. High energy potentials are associated with configurations of A and I that tend to co-occur in training videos with the same collective activity C. For instance, the *talking* collective activity tends to be characterized by interaction activities such as *greeting*, *facing-each-other* and *standing-side-by-side*, as shown in Fig. 2.

3.2 Model Characteristics

The central idea of our model is that the atomic activities of individuals are highly correlated with the overall collective activity, through the interactions between people. This hierarchy is illustrated in Fig. 1. Assuming the conditional independence implied in our undirected graphical model, the overall energy function can be decomposed as a summation of seven local potentials: $\Psi(C, I)$, $\Psi(C, O)$, $\Psi(I, A, T)$, $\Psi(A, O)$, $\Psi(C)$, $\Psi(I)$, and $\Psi(A)$. The overall energy function can easily be represented as in Eq. (1) by rearranging the potentials and concatenating the feature elements to construct the feature vector ψ. Each local potential corresponds to a node (in the case of unitary terms), an edge (in the case of pairwise terms), or a high order potential seen on the graph in Fig. 1c: 1) $\Psi(C, I)$ encodes the correlation between collective activities and interactions (Fig. 2b). 2) $\Psi(I, A, T)$ models the correlation between interactions and atomic activities (Fig. 2a). 3) $\Psi(C)$, $\Psi(I)$ and $\Psi(A)$ encode the temporal smoothness prior in each of the variables. 4) $\Psi(C, O)$ and $\Psi(A, O)$ model the compatibility of the observations with the collective activity and atomic activities, respectively.

Collective-Interaction $\Psi(C, I)$: The function is formulated as a linear multi-class model [40]:

$$\Psi(C, I) = \sum_{t \in \mathcal{T}_V} \sum_{a \in \mathcal{C}} w^a_{ci} \cdot h(I, t; \triangle t_C) \mathbb{I}(a, C(t)), \tag{2}$$

where w_i is the vector of model weights for each class of collective activity, $h(I, t; \triangle t_C)$ is an $\mathcal{I}$ dimensional histogram function of interaction labels around time t (within a temporal window $\pm\triangle t_C$), and $\mathbb{I}(\cdot, \cdot)$ is an indicator function, that returns 1 if the two inputs are the same and 0 otherwise.

Collective Activity Transition $\Psi(C)$: This potential models the temporal smoothness of collective activities across adjacent frames. That is,

$$\Psi(C) = \sum_{t \in \mathcal{T}_V} \sum_{a \in \mathcal{C}} \sum_{b \in \mathcal{C}} w^{ab}_c \mathbb{I}(a, C(t)) \mathbb{I}(b, C(t+1)). \tag{3}$$

Interaction Transition $\Psi(I) = \sum_{i,j} \Psi(I_{ij})$: This potential models the temporal smoothness of interactions across adjacent frames. That is,

$$\Psi(I_{ij}) = \sum_{t \in \mathcal{T}_V} \sum_{a \in \mathcal{I}} \sum_{b \in \mathcal{I}} w^{ab}_i \mathbb{I}(a, I_{ij}(t)) \mathbb{I}(b, I_{ij}(t+1)). \tag{4}$$

Interaction-Atomic $\Psi(I, A, T) = \sum_{i,j} \Psi(A_i, A_j, I_{ij}, T)$: This encodes the correlation between the interaction I_{ij} and the relative motion between two atomic motions A_i and A_j given all target associations T (more precisely the trajectories of T_k and T_l to which τ_i and τ_j belong, respectively). The relative motion is encoded by the feature vector ψ and the potential $\Psi(A_i, A_j, I_{ij}, T)$ is modeled as:

$$\Psi(A_i, A_j, I_{ij}, T) = \sum_{t \in \mathcal{T}_V} \sum_{a \in \mathcal{I}} w^a_{ai} \cdot \psi(A_i, A_j, T, t; \triangle t_I) \mathbb{I}(a, I_{ij}), \tag{5}$$

where $\psi(A_i, A_j, T, t; \triangle t_I)$ is a vector representing the relative motion between two targets within a temporal window ($t - \triangle t_I$, $t + \triangle t_I$) and w^a_{ai} is the model parameter for each class of interaction. The feature vector is designed to encode the relationships between the locations, poses, and actions of two people. See Section 3.3 for details. Note that since this potential incorporates information about the location of each target, it is closely related to the problem of target association. The same potential is used in both the activity classification and the multi-target tracking components of our framework.

Atomic Prior $\Psi(A)$: Assuming independence between pose and action, the function is modelled as a linear sum of pose transition $\Psi_p(A)$ and action transition $\Psi_a(A)$. This potential function is composed of two functions that encode the smoothness of pose and action. Each of them is parameterized as the co-occurrence frequency of the pair of variables similar to $\Psi(I_{ij})$.

Observations $\Psi(A, O) = \sum_i \Psi(A_i, O_i)$ and $\Psi(C, O)$: these model the compatibility of atomic (A) and collective (C) activity with observations (O). Details of the features for atomic activities and collective activities are explained in Sections 9 and 4, respectively.

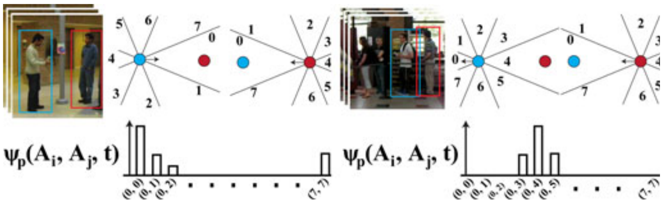


Fig. 3. Illustration of target centric coordinate and histogram ψ_p. *Left-top* and *Right-top* illustrate typical example of *facing-each-other* and *standing-in-a-line* interaction. Given the 3D location (circle) and pose (arrow) of target A_i and A_j, each one's location in terms of the other's view point is obtained as a discretized angle (numbers on the figure); e.g., in the left example, both red and blue people are in the 0th bin of the other's view point. The histograms ϕ_p of each example (*bottom*) are built by counting number of co-occuring discretized angle in a temporal window. With the *facing-each-other* interaction, it is highly likely to observe two people located in $0, 1, 7$th bin of the other's view, producing a pattern similar to the one shown in the *bottom-left*. On the other hand, with the *standing-in-a-line* interaction, it is more likely to observe one in $0, 1, 7$th bin while the other is in $3, 4, 5$th bin, generating a pattern similart to the one shown in the *bottom-right*.

3.3 Interaction Feature

We model the interaction feature as a combination of three types of relative motion features, ψ_l, ψ_p, and ψ_a. Each of the feature vector encodes relative motion (distance and velocity), one's location in another's viewpoint, and co-occurring atomic action. All of them are represented as a histogram so as to capture a non-parametric statistics of interactions.

- ψ_l is a feature vector that captures the relative position of a pair of people. In order to describe the motion of one respect to the other, ψ_l is represented as a histogram of velocity and location difference between the two within a temporal window ($t - \triangle t, t + \triangle t$).
- ψ_p encodes a person's location with respect to the other's viewpoint. First, we define the ith target centric coordinate system for each time t by translating the origin of the system to the location of the target i and rotating the x-axis along the viewing direction (pose) of the target i. At each time stamp t in the temporal window, the angle of each target within the others' coordinate system is computed and discretized angle is obtained (see Fig. 3) in order to describe the location of one person in terms of the viewpoint of the other. Given each location bin, histogram ψ_p is built by counting the number of occurrences of the bin number pair to encode the spatial relationship between two targets within a temporal window ($t - \triangle t, t + \triangle t$).
- ψ_a models co-occurrence statistics of atomic actions of the two targets within a temporal window ($t - \triangle t$, $t + \triangle t$). It is represented as a $|\mathcal{A}| \times (|\mathcal{A}| + 1)/2$ dimensional vector of $(a_i(t), a_j(t))$ histogram.

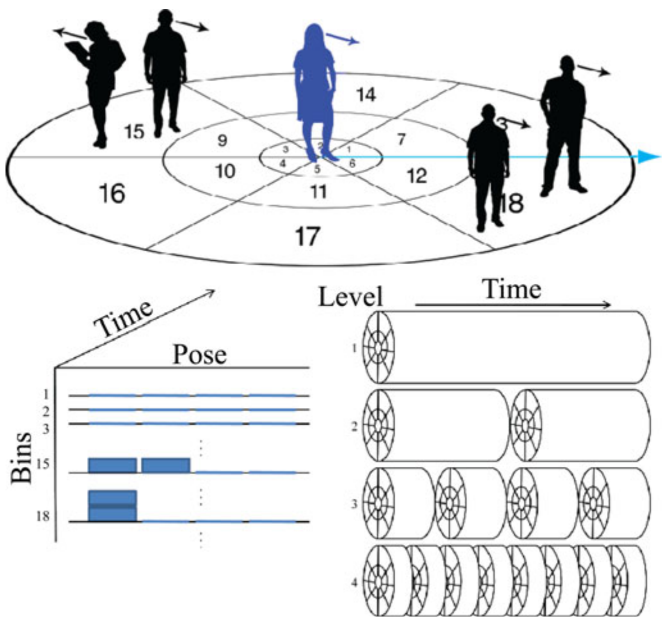


Fig. 4. Spatio-Temporal Local Descriptor. (a) Space around anchor person (blue) is divided into multiple bins. The pose of the anchor person (blue arrow) locks the "orientation" of the descriptor which induces the location of the reference bin "1". (b) Example of STL descriptor-the descriptor is a histogram capturing people and pose distribution in space and time around the anchor person. (c) Classification of STL descriptor is achieved by decomposing the histogram in different levels along the temporal axis.

Note that the first two features ψ_l, ψ_p are dependent on the trajectories of the two targets. Thus, change in association will result in a higher or lower value of an interaction potential.

4 Crowd Context

In this section, we introduce the definition of the *crowd context* and describe its mathematical formulation given a set of spatio-temporal trajectories. The concept of crowd context is originally introduced in our works [7], [8]. The crowd context captures *the coherent behavior of individuals in time and space* that are performing a certain collective activity. Such contextual information encodes the direct observation cue for the collective activities ($\Psi(C, O)$ in Section 3).

A naive way for characterizing the crowd context is by encoding the spatial-temporal dependencies of individuals in a neighborhood of the video sequence. This can be done by using the spatial-temporal-local (STL) descriptor introduced in [7]. The STL descriptor is in essence a fixed-dimensional vector (Fig. 4) and is associated to each person. For each time stamp, the STL descriptors are used to classify the collective activity using a standard support vector machine (SVM) [4] classifier. Temporal smoothness is enforced by applying a markov chain model across each time stamp. Though the method shows promising results, such rigid descriptors require the parameters that control the structure of the descriptor to be manually tuned, which can be extremely difficult in presence of large intra-class variability. Such limitation can be addressed by introducing a new scheme called randomized spatio temporal volume (RSTV) which is used to automatically learn the best structure of the descriptor. In the following sections, we review the rigid STL descriptor first and the extended RSTV later.

4.1 Rigid STL Descriptor

In this section, we describe how to extract an STL descriptor for each individual (track) in each time stamp given a set of tracklets $\{\tau_1, \tau_2, \ldots, \tau_N\}$, where $\tau_i = \{l_i, p_i, t_i\}$ is an individual tracklet and $l_i = (x_i, y_i)$, p_i and t_i are sequences of x, y location, pose and time index, respectively. Note that the pose captures the orientation of an individual in this framework (e.g., left, front, right, and back).

Given a person i in certain time stamp t (the *anchor*), they determine the locations l_j^i and poses p_j^i of other individuals in the anchor's coordinate system, where the anchor's coordinate system has the origin at the anchor's (x, y) location and is oriented along the pose direction of the anchor (see Fig. 4 top). The space around each anchor i at time t is divided into multiple bins following a log-polar space partition similar to the shape context descriptor [2]. Moreover, for each spatial bin, P "pose" bins are considered where P is the number of poses that are used to describe a person orientation. Finally, the temporal axis is also decomposed in temporal bins around time stamp t. This spatial, temporal and pose sensitive structure is used to capture the distribution of individuals around the anchor i at time t and construct the STL descriptor. For each anchor i and time stamp t, an STL descriptor is obtained by counting the number of individuals that fall in each bin of the structure described above. Thus, the STL descriptor implicitly embeds the *flow* of people around the anchor over a number of time stamps. After accumulating the information, the descriptor is normalized by the total number of people that fall in the spatio-temporal extension of the descriptor.

There are a number of important characteristics of the STL descriptor. First, the descriptor is rotation and translation invariant. Since the relative location and pose of individuals are defined in the anchor's coordinate system, the descriptor yields a consistent representation regardless of the orientation and location of the anchor in the world. Moreover, the dimensionality of the descriptor is fixed regardless of the number of individuals that appear in the video sequence. This property is desirable in that it allows to represent an activity using a data structure that is not a function of the specific instantiation of a collective activity. Finally, by discretizing space and time into bins, the STL descriptor enables a classification scheme for collective activities that is robust to variations in the spatio-temporal location of individuals for each class of activity (intra-class variation).

Given a set of STL descriptors (each person in the video is associated to a STL descriptor) along with the associated collective activity labels, one can solve the collective activity classification problem by using a classification method such as SVM [4]. In order to capture various levels of temporal granularity, the authors of [7] adopt SVM classifier equipped with a temporal pyramid intersection kernel (see Fig. 4 bottom right). The temporal axis is divided into four hierarchical levels of temporal windows and intersection kernel is defined per each level. The finest temporal window allows to capture the detailed motion of individuals around the anchors; the highest level allows to encode the

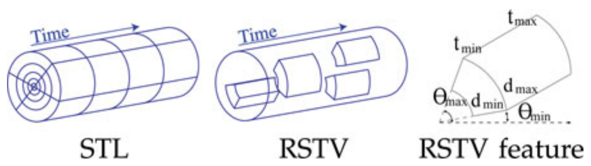


Fig. 5. STL counts the number of people in each spatio-temporal and pose bins that are divided following a hand defined parameterization (*left*). On the other hand, the RSTV learns what spatial bins are useful (shown as a trapezoid-like volume) in order to discriminate different collective activities and discards the regions (shown as empty regions) that are not helpful for such discrimination task (*middle*). A random spatio-temporal volume (feature) is specified by a number of parameters (*right*). Pose and velocity are omitted from the illustration.

overall distribution of people around the anchor over the observed period. This classification scheme is used as a baseline method in Section 9.

4.2 Learning the Crowd Context

The STL descriptor is limited in that the structure of the bins of the STL descriptor is predefined beforehand and parameters such as the minimum distance from the anchor or the maximum support of the descriptor are defined once for all. In particular, by assuming that the spatial support has fixed size, the STL descriptor does not have the ability to adaptively filter out background activities or activities that differ from the dominant one.

In order to avoid above mentioned limitations, we introduce a novel scheme, called Randomized Spatio-Temporal Volume. The RSTV approach is based on the same intuition as STL that crowd context can be captured by counting the number of people with a certain pose and velocity in fixed regions of the scene, relative to an anchor person. However, RSTV extends this intuition and considers variable spatial regions of the scene with a variable temporal support. The full feature space contains the evidence extracted from the entire videos: the location of each individual in anchor's coordinates as well as the velocity and pose of each individual per video frame. This can be interpreted as a soft binning scheme where the size and locations of bins are estimated by a random forest so as to obtain the most discriminative regions in the feature space. Over these regions, the density of individuals is inspected, which can be used for classification. Fig. 5 compares the rigid STL binning scheme and the flexible RSTV. RSTV is a generalization of the STL in that the rigid binning restriction imposed in the STL is removed. Instead, portions of the continuous spatio-temporal volume are sampled at random and the discriminative regions for classification of a certain activity are retained. RSTV provides increasing discrimination power due to increased flexibility.

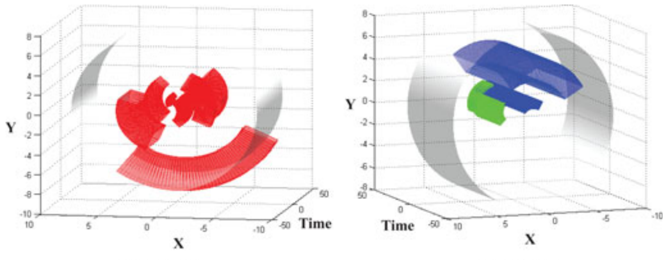
There are several benefits of the RSTV framework over rigid STL descriptor. 1) The RSTV automatically determines the discriminative features in the feature space that are useful for classification. Indeed, while STL proposes a rigid and arbitrary decomposition of the feature space, in RSTV the binning space is partitioned so as to maximize discrimination power. 2) Unlike STL, there are no parameters that are to be learned or selected empirically (e.g., support distance, number of bins). 3) It enables robustness to clutter. Indeed, unlike STL, the RSTV does not operate given fixed parameters such as radial support and number of spatial bins, but explores the possible space of parameters; thus the density feature, using which classification is performed, is only calculated over regions relevant to each different activity. Hence the classification evidence is pertinent to each activity and avoid clutter that possibly arises from hard-coded framework parameters that may be tuned to achieve optimal classification of a few activities, but not all. Notice that STL concept is similar to the Shape Context [2] descriptor, which is known to be susceptible to clutter due to non discriminative inclusion of all points within the radial support.

Learning RSTV with Random Forest: We use a Random Forest classifier to learn the structure of RSTV given training data. A Random forest [3] is an ensemble of many singular classifiers known as decision trees which is trained from a portion of the training data. The training set is subdivided into multiple *bags* by random sampling with replacement (*bagging*) in order to reduce the effect of over-fitting. Given each set, one random decision tree is trained following successively drawing and selection of a random feature that best discriminates the given training set [3].

The RSTV is trained based on the random forest classifier given a set of training data and associated activity labels (x_i, y_i) where each data point is defined for each person and time stamp. In following description, it is assumed that the trajectories and poses of all people are already transformed into the anchor's coordinate system to form data point x_i and associated activity label y_i. Given a random bag, a random decision tree is learned by recursively discovering the most discriminative features. The algorithm first randomizes over different volumes of the feature space and second randomizes over different decision thresholds given the feature subspace. The feature is defined as the number of people lying in a spatio-temporal volume that is specified by location (l^k), velocity (v^k), pose (p^k) and time (t) defined in the anchor's (k) coordinate system. A unique spatio-temporal volume is specified by a number of parameters: 1) minimum and maximum distance d_{min}, d_{max}, 2) minimum and maximum angle in the space θ_{min}, θ_{min}, 3) relative orientation/pose p, 4) temporal window t_{min}, t_{max} and 5) minimum and maximum velocity v_{min}, v_{max} (Fig. 5 right). In each node, a number M of such hyper-volume r_n and a scalar decision threshold d_n is drawn randomly multiple times. Given the feature pair (r_n, d_n), the training data is partitioned into two subsets I_r and I_l by testing $f(x; r_n) > d_n$, where $f(x; r_n)$ is a function that counts the number of people lying in the hyper volume r_n. Among the set of candidate features, the one that best discriminates the training data into two partitions is selected by examining the information gain (Eq. (6)).

$$\Delta E = -\frac{|I_l|}{|I|} E(I_l) - \frac{|I_r|}{|I|} E(I_r),\ where\ E(I) = -\sum_{i=1}^{C} p_i\ log_2(p_i), \tag{6}$$

I_l and I_r are the partition of set I divided by given feature, C is the number of activity classes, p_i is the proportion of collective activity class i in set I, and $|I|$ is the size of the set I.



(a) Waiting (b) Talking

Fig. 6. Example of learned RSTV regions. (a) & (b) illustrate a set of RSTV regions learned automatically by a single tree. Each colour indicates different pose of neighbouring individuals (up—red, down—blue and right—green). Each RSTV is oriented such that the anchor is facing in the upward z direction. Hence (a) indicates that while waiting, an anchor is surrounded on the left and right by people facing the same direction. RSTV in (b) illustrates that during talking the anchor and neighbour face each other and are in very close proximity. Note that each RSTV needs only capture some coherent portion of evidence since there exist many trees in the RF. x and z have units of meters while time is measured in frames.

Typical examples of learned RSTV structure is shown in Fig. 6. The detailed algorithm for learning RSTV is presented in Algorithm 1 and Algorithm 2.

Algorithm 1 RSTV learning

```
Require: $I = \{(x_i, y_i)\}$
  Randomly draw a bag $I_t$ for each tree
  for all random decision tree do
    At the root node, $root \leftarrow NodeLearn(I_t)$
  end for
```

Given the learned RSTV forests, one can classify a novel testing example x by passing down the example along each tree and taking the class that maximizes marginal posterior probability $P(y|x) = \sum_{tree} P_{tree}(y|x)$ over all trees. The posterior probability of a tree is defined as the corresponding p_y in the leaf node that the testing example reached in the decision tree.

5 Multiple Target Tracking

Our multi-target tracking formulation follows the philosophy of [38], where tracks are obtained by associating corresponding tracklets. Unlike other methods, we leverage the contextual information provided by interaction activities to make target association more robust. Here, we assume that a set of initial tracklets, atomic activities, and interaction activities are given. We will discuss the joint estimation of these labels in Section 6.

Algorithm 2 Recursive Node Learning (NodeLearn)

```
Require: $I_n$
  if $|I_n| < N_{min}$ then
    Compute distribution of classes $p_i$ over all $C$
    $node.isleaf \leftarrow TRUE$
    $node.p \leftarrow p_i$
    return node
  end if
  $\Delta E_{max} \leftarrow -INF$
  for $m = 0$ to $M$ do
    Randomly draw a feature pair $(r_n^m, d_n^m)$
    Compute information gain $\Delta E_m$
    if $\Delta E_{max} < \Delta E_m$ then
      $\Delta E_{max} \leftarrow \Delta E_m$
      $(r_n, d_n) \leftarrow (r_n^m, d_n^m)$
    end if
  end for
  Partition $I_n$ into $(I_l, I_r)$ using $(r_n, d_n)$
  $node.isleaf \leftarrow FALSE$
  $node.left \leftarrow NodeLearn(I_l)$
  $node.right \leftarrow NodeLearn(I_r)$
  $node.feature \leftarrow (r_n, d_n)$
  return node
```

As shown in Fig. 7, tracklet association can be formulated as a min-cost network problem [44], where the edge between a pair of nodes represents a tracklet, and the black directed edges represent possible links to match two tracklets. We refer the reader to [31], [44] for the details of network-flow formulations.

Given a set of tracklets $\tau_1, \tau_2, \ldots, \tau_N$ where $\tau_i = \{x_{\tau_i}(t_0^i), \ldots, x_{\tau_i}(t_e^i)\}$ and $x(t)$ is a position at t, the tracklet association problem can be stated as that of finding an unknown number M of associations $T_1, T_2, \ldots, T_M$, where each T_i contains one or more indices of tracklets. For example, one association may consist of tracklets 1 and 3: $T_1 = \{1, 3\}$. To accomplish this, we find a set of possible paths between two non-overlapping tracklets τ_i and τ_j. These correspond to match hypotheses $p_{ij}^k = \{x_{p_{ij}^k}(t_e^i + 1), \ldots, x_{p_{ij}^k}(t_0^j - 1)\}$ where the time stamps are in the temporal gap between τ_i and τ_j. The association T_i can be redefined by augmenting the associated pair of tracklets τ_i and τ_j with the match hypothesis p_{ij}. For example, $T_1 = \{1, 3, 1\text{-}2\text{-}3\}$ indicates that tracklets 1 and 3 form one track and the second match hypothesis (the solid edge between τ_1 and τ_3 in Fig. 7) connects them. Given

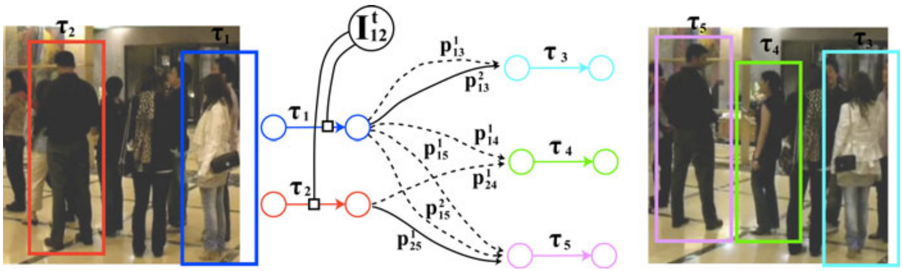


Fig. 7. The tracklet association problem is formulated as a min-cost flow network [31], [44]. The network graph is composed of two components: tracklets τ and path proposals p. In addition to these two, we incorporate interaction potential to add robustness in tracklet association. In this example, the interaction "standing-in-a-row" helps reinforce the association between tracklets τ_1 and τ_3 (or τ_2 and τ_5) and penalizes the association between τ_1 and τ_4 (or τ_1 and τ_5).

human detections, we can generate match hypotheses using the K-shortest path algorithm [43] (see Sections 4.2 and 5.1 for details).

Each match hypothesis has an associated cost value c_{ij}^k that represents the validity of the match. This cost is derived from detection responses, motion cues, and color similarity. By limiting the number of hypotheses to a relatively small value of K, we prune out a majority of the exponentially many hypotheses that could be generated by raw detections. If we define the cost of entering and exiting a tracklet as c_{en} and c_{ex} respectively, the tracklet association problem can be written as:

$$\begin{aligned}\hat{f} &= \operatorname*{argmin}_f c^T f\\ &= \operatorname*{argmin}_f \sum_i c_{en} f_{en,i} + \sum_i c_{ex} f_{i,ex} + \sum_{i,j}\sum_k c_{ij}^k f_{ij}^k\\ &\qquad s.t. f_{en,i}, f_{i,ex}, f_{ij}^k \in \{0,1\},\\ &\qquad f_{en,i} + \sum_j\sum_k f_{ji}^k = f_{i,ex} + \sum_j\sum_k f_{ij}^k = 1,\end{aligned} \tag{7}$$

where f represent the flow variables, the first set of constraints is a set of binary constraints and the second one captures the inflow-outflow constraints (we assume all the tracklets are true). Later in this paper, we will refer to $\mathbb{S}$ as the feasible set for f that satisfies the above constraints. Once the flow variable f is specified, it is trivial to obtain the tracklet association T through a mapping function $T(f)$. The above problem can be efficiently solved by binary integer programming, since it involves only a few variables, proportional to the number of tracklets N that is typically a few hundred, and there are $2N$ equality constraints. Note that the number of nodes in [31], [44] is usually in the order of tens or hundreds of thousands.

One of the novelties of our framework lies in the contextual information that comes from the interaction activity nodes. For the moment, assume that the interactions I_{12}^t between A_1 and A_2 are known. Then, selecting a match hypothesis f_{ij}^k should be related with the likelihood of observing the interaction I_{12}^t. For instance, the *red* and *blue* targets in Fig. 7 are engaged in the *standing-in-a-row* interaction activity. If we select the match hypothesis that links *red* with *pink* and *blue* with *sky-blue* (shown with solid edges), then the interaction will be compatible with the links, since the distance between *red* and *blue* is similar to that between *pink*/*sky-blue*. However, if we select the match hypothesis that links *red* with *green*, this will be less compatible with the *standing-in-a-row* interaction activity, because the *green*/*pink* distance is less than the *red*/*blue* distance, and people do not tend to move toward each other when they are in a queue. The potential $\Psi(I, A, T)$ (Section 3.2) is used to enforce this consistency between interactions and tracklet associations.

5.1 Hypothesis Generations

For any pair of tracklets τ_i, τ_j that are not co-present at the same time-stamp (thus can be linked), we generate K path hypotheses to associate the two tracklets into a unique track. Such hypotheses are obtained by finding K-shortest paths between the two tracklets in a detection graph (Fig. 8). The graph is built by connecting the residual detections between the two tracklets.

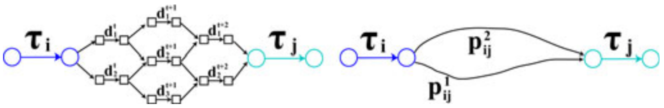


Fig. 8. Illustration of path hypothesis generation given detection residuals. *Left:* the graph is composed of detections in the temporal gap between τ_i and τ_j. Each detection is represent as a pair of square nodes that are linked by a detection response edge. The cost d associated with the edge encodes the detection confidence value. The detections in time $t+1$ that has enough overlap with the detections in time t are added to the graph. *Right:* given the detection residual graph above, we can obtain a concise set of path proposals using K-shortest path search method. Note that there can be exponential number of possible path in the first graph.

To illustrate, consider the example shown in Fig. 8. Beginning from the last frame (shown as $t-1$) of preceding tracklet τ_i, we find the residual detections at t that have sufficient amount of overlap with the bounding box of τ_i at $t-1$. We add these detections as a pair of nodes (shown as square nodes in Fig. 8) and a cost edge (link the two nodes) into the graph. These nodes are linked to the previous frame's tracklet by a directed edge. Subsequently, we add detections in time stamp $t+1$, by calculating the overlap between the added detection in time t and all residual detections in time $t+1$. We add detection nodes in all time stamps between τ_i and τ_j iteratively and finish the graph building process by considering the connectivity between τ_j and detections at $t+2$. The detections in $t+2$ that do not overlap sufficiently with the bounding box of τ_j at the first frame are discarded.

As noted in the graph, there is an exponentially large (and redundant) number of possible paths that link the two tracklets, which require extensive amount of computation. If we consider to take the interaction potential into account for tracklet association, it is required to compute an interaction feature for each possible path of targets. This can result into an infeasible amount of computation in target association. To avoid this issue, we use the K-shortest paths search method [43] that generates a concise set of path hypotheses to link the two tracklets (Fig. 8). In practice, we consider the detection confidence to obtain the cost for simplicity. One can add more cost features such as color similarity, motion smoothness, if desired. To avoid having no proposals when there are missing detections, we add one default hypothesis that links two tracklets in a shortest distance.

5.2 Match Features

Each path p_{ij}^k is associated to a cost value c_{ij}^k that measures the likelihood that the two tracklets τ_i, τ_j belong to the same target. We model this cost value as a linear weighted sum of multiple match features: color difference, height difference, motion difference and accumulated detection confidences of the path. In details,

$$c_{ij}^k = w_m^T d_k(\tau_i, \tau_j), \tag{8}$$

where w_m is a model weight and $d_k(\tau_i, \tau_j)$ is a vector that collects all the features. Each of the features is obtained as follows: i) color difference is obtained by the Bhattacharyya distance between color histograms of τ_i and τ_j, ii) height difference is encoded by computing the difference between average height of τ_i and τ_j, iii) motion difference is computed by absolute difference in the velocity of τ_i and τ_j, and iv) accumulated detector confidence is calculated by summing up the detection confidence in the path p_{ij}^k.

Given the match features, we obtain the cost of each path proposal by Eq. (8). In the case of target initiation and termination, we use the cost value c_{en}, c_{ex} to model the cost of initiating and terminating a target.

6 Unifying Activity Classification and Tracklet Association

The previous two sections present collective activity classification and multi-target tracking as independent problems. In this section, we show how they can be modelled in a unified framework. Let $\hat{y}$ denote the desired solution of our unified problem. The optimization can be written as:

$$\hat{y} = \operatorname*{argmax}_{f,C,I,A} \underbrace{\Psi(C,I,A,O,T(f))}_{Sec.3} - \underbrace{c^T f}_{Sec.5},\ s.t.\ f \in \mathbb{S}, \tag{9}$$

where f is the binary flow variables, $\mathbb{S}$ is the feasible set of f, and C, I, A are activity variables. As noted in the previous section, the interaction potential $\Psi(A,I,T)$ involves the variables related to both activity classification (A, I) and tracklet association (T). Thus, changing the configuration of interaction and atomic variables affects not only the energy of the classification problem, but also the energy of the association problem. In other words, our model is capable of propagating the information obtained from collective activity classification to target association and from target association to collective activity classification through $\Psi(A,I,T)$.

Algorithm 3 Iterative Belief Propagation

```
Require: Given association $\hat{T}$ and observation $O$.
  Initialize $C^0, I^0, A^0$
  while Convergence, k++ do
    $C^k \Leftarrow \text{argmax}_C\ \Psi(C, I^{k-1}, A^{k-1}, O, \hat{T})$
    for all $\forall i \in A$ do
      $A_i^k \Leftarrow \text{argmax}_A\ \Psi(C^k, I^{k-1}, A, A_{\setminus i}^{k-1}, O, \hat{T})$
    end for
    for all $\forall i \in I$ do
      $I_i^k \Leftarrow \text{argmax}_I\ \Psi(C^k, I, I_{\setminus i}^{k-1}, A^k, O, \hat{T})$
    end for
  end while
```

6.1 Inference

Since the interaction labels I and the atomic activity labels A guide the flow of information between target association and activity classification, we leverage the structure of our model to efficiently solve this complicated joint inference problem. The optimization problem Eq. (9) is divided into two sub problems and solved iteratively:

$$\{\hat{C}, \hat{I}, \hat{A}\} = \operatorname*{argmax}_{C,I,A} \Psi(C,I,A,O,T(\hat{f})) \tag{10}$$

$$\hat{f} = \operatorname*{argmin}_{f} c^T f - \Psi(\hat{I}, \hat{A}, T(f)), s.t. f \in \mathbb{S}. \tag{11}$$

Given $\hat{f}$ (and thus $\hat{T}$) the hierarchical classification problem is solved by applying iterative belief propagation. Fixing the activity labels A and I, we solve the target association problem by applying the Branch-and-Bound algorithm with a tight linear lower bound (see below for more details).

Iterative Belief Propagation. Due to the high order potentials in our model (such as the collective-interaction potential), the exact inference of the all variables is intractable. Thus, we propose an approximate inference algorithm that takes advantage of the structure of our model. Since each type of variable forms a simple chain in the temporal direction (see Fig. 1), it is possible to obtain the optimal solution given all the other variables by using belief propagation [13]. The iterative belief propagation algorithm is grounded in this intuition, and is shown in detail in Algorithm 3.

Target Association Algorithm. We solve the association problem by using the Branch-and-Bound method [22]. Unlike the original min-cost flow network problem, the interaction terms introduce a quadratic relationship between flow variables. Note that we need to choose at most two flow variables to specify one interaction feature. For instance, if there exist two different tails of tracklets at the same time stamp, we need to specify two of the flows out of seven flows to compute the interaction potential as shown in Fig. 7. This leads to a non-convex binary quadratic programming problem which is hard to solve exactly (the Hessian H, that contains information from interaction potentials (see Section 7), is not a positive semi-definite matrix).

$$\operatorname*{argmin}_{f} \frac{1}{2} f^T H f + c^T f,\ s.t.\ f \in \mathbb{S} \tag{12}$$

To tackle this issue, we use a Branch-and-Bound algorithm with a novel tight lower bound function given by $h^T f \leq \frac{1}{2} f^T H f,\ \forall f \in \mathbb{S}$ (Section 7).

7 Tracklet Association with Interaction Potential

The target association problem with the interaction potential can be written as:

$$\begin{aligned} \hat{f} = \operatorname*{argmin}_{f}\ & c^T f - \Psi(I, A, T(f)) \\ s.t.\ & f_{en,i}, f_{i,ex}, f_{ij}^k \in \{0,1\} \\ & f_{en,i} + \sum_j \sum_k f_{ji}^k = f_{i,ex} + \sum_j \sum_k f_{ij}^k = 1, \end{aligned} \tag{13}$$

where the constraints are summarized as: 1) binary flow constraints (the flow variable should be 0 or 1 integer value specifying that a path is valid or not) and 2) inflow-outflow constraints (the amount of flow coming into a tracklet should be the same as the amount of flow going out of it and the amount is either 0 or 1). The c vector is a cost vector that measures the likelihood of linking two tracklets c_{ij}^k or the cost to initiate/terminate a target c_{en}, c_{ex}. The second term $\Psi(I, A, T(f))$ encodes the interaction potential which is dependent on the trajectories derived from tracklet association.

7.1 The Non-Convex Quadratic Objective Function

Though the match cost $c^T f$ is represented as a linear function, the interaction potential involves quadratic relationship between flow variables. As discussed in Section 3, the interaction potential $\Psi(I, A, T(f))$ is composed of a sum of interaction potentials each of which is associated to a single interaction variable.

$$\Psi(I, A, T) = \sum_{i,j} \Psi(A_i, A_j, I_{ij}, T), \tag{14}$$

$$\Psi(A_i, A_j, I_{ij}, T) = \sum_{t\in\mathcal{T}_V} \sum_{a\in\mathcal{I}} w^a_{ai} \cdot \psi(A_i, A_j, T, t; \triangle t)\ \mathbb{I}(a, I_{ij}). \tag{15}$$

Since the feature function ψ is dependent on at most two flow variables, the overall objective function can be represented as a quadratic function.

Let us start by introducing a few definitions. We define the *head* and *tail* path of a tracklet τ_i as the path through which the flow comes into τ_i and the path through which the flow goes out from τ_i, respectively. The *head* path of τ_i can be between the entering path $f_{en,i}$ and the path connecting from any other tracklet τ_l, f^k_{li}. Similarly, the *tail* path of τ_i can be between the exiting path $f_{ex,i}$ and the path connecting to any other tracklet τ_m, f^k_{im}. A tracklet τ_i is called *intact* in a certain temporal support $t \in (t_1,\ t_2)$, if the trajectory of the target is fully covered by the tracklet within the temporal support (i.e, the tracklet is not fragmentized within the time gap). Otherwise, it is called *fragmentized* in a certain temporal support $t \in (t_1,\ t_2)$.

In order to calculate the interaction between two targets i and j at certain time stamp t, we need to specify the trajectory of A_i and A_j in all time stamps $t \in (t - \triangle t,\ t + \triangle t)$ (the temporal support of an interaction, Section 3.3), which can involve selecting at most two flow variables in our flow network.[1] If the both tracklets are *intact* within the temporal support of I^t_{ij}, the interaction potential does not get affected by tracklet association (we need to specify no flow variable to compute the interaction feature and thus it can be ignored). If only one of the tracklets is *fragmentized* and the other is *intact*, we need to specify only one *head* or *tail* path of the fragmentized tracklet. On the other hand, if the both τ_i and τ_j are fragmentized in the temporal support, we need to specify two flow variables to obtain the associated interaction feature (*head* or *tail* of τ_i and *head* or *tail* of τ_j).

Since the objective function can be specified as a sum of quadratic and linear functions of flow variable f, the problem can be re-written as follows:

$$\begin{aligned} \hat{f} &= \underset{f}{\operatorname{argmin}}\ \ c^T f - \Psi(I, A, T(f)) \\ &= \underset{f}{\operatorname{argmin}}\ \ c^T f + d^T f + f^T H f \\ &\quad\ s.t.\ f \in \mathbb{S}, \end{aligned} \tag{16}$$

$\mathbb{S}$ represent the feasible set for f that satisfies the constraints listed in Eq. (13), the linear part of interaction potential d can be obtained by accumulating the interaction potentials that involve only one selection of path (one of the two tracklets τ_i, τ_j is *intact* within the temporal support), and the Hessian H can be obtained by accumulating all interaction potentials that involve two selections of flow variables (both of τ_i, τ_j are fragmentized in the temporal support of the given interaction variable as in the example of Fig. 7). Note that H is not positive semi-definite (thus non-convex) and standard quadratic programming techniques are not applicable.

7.2 Branch-and-Bound

Since the objective function is non-convex, we employ a novel Branch-and-Bound algorithm to solve the complicated tracklet association problem. The Branch-and-Bound algorithm we describe here find the global minimum of the objective function over the space $\mathbb{S}$. Starting from the initial subproblem $\mathcal{Q} = \mathbb{S}$, we split the space into two subspaces $\mathcal{Q}_0, \mathcal{Q}_1$ by setting 0 and 1 to a certain flow variable f_i (ignoring/selecting a path). Given each subproblem (where some of flow variables are already set either 0 or 1), we find the lower bound and upper bound (of optimal solution) in the subproblem, $L(\mathcal{Q})$ and $U(\mathcal{Q})$. If the difference between L and U is smaller than a specified precision ϵ and $U(\mathbb{S})$ is smaller than the lower bound of any other subspace, we stop the iteration and yield the global solution. Otherwise, the algorithm iterate the steps of 1) selecting a subproblem, 2) splitting the subproblem, and 3) finding the lower and upper bound in the subproblem. This is summarized in Algorithm 4.

Algorithm 4 Branch and Bound (BB) Tracklet Association

```
$\mathcal{Q} = \mathbb{S}$
$L_0 = L(\mathcal{Q})$
$U_0 = U(\mathcal{Q})$
$\mathcal{L} = \{\mathcal{Q}\}$
while $U_k - L_k > \epsilon$, k++ $<$ maxIter do
    Select a subproblem $\mathcal{Q} \in \mathcal{L}_k$ for which $L(\mathcal{Q}) = L_k$.
    Split $\mathcal{Q}$ into $\mathcal{Q}_0$ and $\mathcal{Q}_1$
    Form $\mathcal{L}_{k+1}$ from $\mathcal{L}_k$ by removing $\mathcal{Q}$ and adding $\mathcal{Q}_0$ and $\mathcal{Q}_1$
    $L_{k+1} = \min_{\mathcal{Q}\in\mathcal{L}_{k+1}} L(\mathcal{Q})$
    $U_{k+1} = \min_{\mathcal{Q}\in\mathcal{L}_{k+1}} U(\mathcal{Q})$
end while
```

In following sections, we discuss about how we compute the lower and upper bound of a subproblem $\mathcal{Q}$ (Section 7.3) and which variable is to be split to provide subproblems $\mathcal{Q}_0$ and $\mathcal{Q}_1$ (Section 7.4).

7.3 Lower Bound

In this section, we discuss how we define the lower bound of the objective function. To make it efficient to solve, we find a linear lower bound function:

$$L(f) = (c + d + l)^T f \le (c + d)^T f + f^T H f, f \in \mathcal{Q}. \tag{17}$$

Since the whole interaction potential is represented as a sum of interaction potentials associated with a single interaction variable, it suffices to show that the $l^T f$ is less than or equal to $f^T H f$ within one interaction potential (associated to a single interaction variable I^k_{ij}); that is $l^T f \le f^T H f$, if $l^T_i f \le f^T H_i f$, $\forall i$ where i denotes an index

[1] Note that it can involve up to four selections of path proposals to fully specify the trajectories of A_i and A_j: *head* of A_i, *tail* of A_i, *head* of A_j and *tail* of A_j if the two tracklets are both fragmentized in both directions within the temporal support of an interaction. However, we ignore such cases since i) it rarely happens, ii) it make the algorithm to be over-complicated and iii) if the tracklets are too short there are not reliable information we can exploit in the first place.

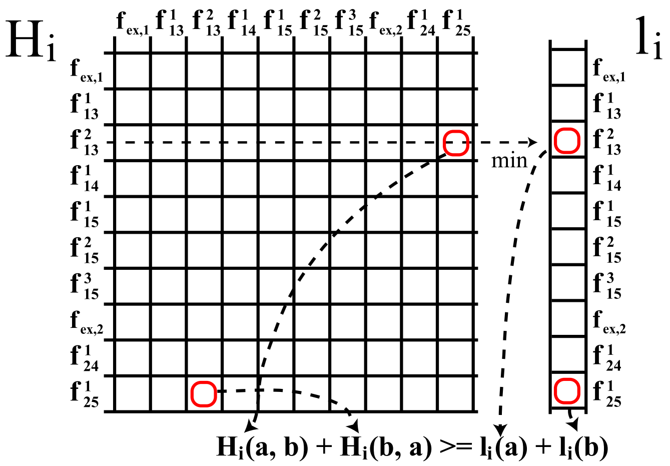

H_i
$f_{ex,1}$ f^1_{13} f^2_{13} f^1_{14} f^1_{15} f^2_{15} f^3_{15} $f_{ex,2}$ f^1_{24} f^1_{25}
l_i
min
$H_i(a, b) + H_i(b, a) >= l_i(a) + l_i(b)$

Fig. 9. Illustration of lower bound L computation for the interaction variable corresponding to Fig. 7. Each element of the Hessian H_i is obtained by computing the corresponding interaction potential $\Psi(A_i, A_j, I^t_{ij}, T)$ given the flow configuration. A linear lower bound $l^T f$ is derived from $f^T H f$ by taking the minimum of each row in the hessian H matrix. Note that only one configuration can be selected in the matrix H with symmetry since no two flow coming out from one tracklet τ_i or τ_j can be set simultaneously. The example shows the case when solid edges in Fig. 7 are selected.

that enumerates all interaction variables I^k_{ij}, H_i encodes the contextual information from a single interaction I^k_{ij}, l_i is a linear lower bound vector associated to each H_i, $l = \sum_i l_i$ and $H = \sum_i H_i$.

Thus, we decompose the Hessian H into summation of H_i and show that there exists a linear vector l_i that yields a lower bound of $f^T H_i f$. The matrix H_i can be obtained by computing the corresponding interaction potential $\Psi(A_i, A_j, I^t_{ij}, T(f))$ given each possible configuration of path flows, e.g., selecting the two solid paths shown in the Fig. 7.

$$H_i(a, b) = -\frac{1}{2}\Psi(A_i, A_j, I^t_{ij}, T(f)) \ where \ f_a = f_b = 1, \tag{18}$$

where a, b denote the indices of a flow variable (paths).

To obtain the lower bound of $f^T H_i f$, we take advantage of two properties: i) the variables are binary and ii) there must be one and only one inflow and outflow for each tracklet τ_i. These two facts can be easily derived from the basic constraints of the problem ($\mathbb{S}$). Given these, we notice that any two elements in H_i are always selected with symmetry (shown as red box in Fig. 9) and the values are added to produce $f^T H_i f = H_i(a, b) + H_i(b, a)$ where a and b are the indices of the selected variables in f. Thus, it is easy to show that,

$$\min_k H_i(a, k) + \min_k H_i(b, k) \leq H_i(a, b) + H_i(b, a). \tag{19}$$

From this, we obtain the lower bound vector l_i for H_i as

$$l_i(a) = \min_k H_i(a, k) \tag{20}$$

(see Fig. 9). The overall lower bound function l is obtained by summing up all lower bounds associated to each interaction variable: $l = \sum_i l_i$.

Given the lower bound function l, the lower bound of $\mathcal{Q}$ is obtained by applying binary integer programming on the lower bound with the given constraints of $\mathcal{Q}$, that is $\overline{f} = \mathrm{argmin}_f (c + c_I + l)^T f, \ s.t. \ f \in \mathcal{Q}$. The upper bound is set to be infinite if there is no feasible solution, or set to be the value of original objective function if the solution $\overline{f}$ we obtained is feasible.

7.4 Split Variable Selection

Though the presented lower bound can generate a rather tight lower bound in our problem, not all the flow variables in f have the same "tightness". Splitting on certain a flow variable f_a (setting it to one or zero) will give a higher uncertainty (larger gap between the lower bound and actual objective function) than splitting on others. To efficiently split the space and find the solution, we follow a strategy where the flow variable that is associated to the largest degree of ambiguity(gap) is selected in each iteration of the branch and bound procedure. In details, in order to measure the degree of ambiguity, we derive upper bound vector u_i from H_i by

$$u_i(a) = \max_k H_i(a, k). \tag{21}$$

Notice that we take the maximum of a given row in contrast to the minimum in lower bound case (Eq. (20)). Similar to the lower bound vector case, we can obtain full upper bound vector u by accumulating over different interaction variables. It is trivial to show that:

$$l^T f \leq f^T H f \leq u^T f. \tag{22}$$

Low degree of ambiguity takes place when the value of $l(a)$ is close to $u(a)$ and high degree of ambiguity takes place when the gap between $l(a)$ and $u(a)$ is large. Therefore, we choose the variable to be split by finding the variable that has largest difference, $\mathrm{argmax}_a u(a) - l(a)$.

8 Model Learning

Given the training videos, the model is learned in a two-stage process: i) learning the observation potentials $\Psi(A, O)$ and $\Psi(C, O)$. This is done by learning each observation potential $\Psi(\cdot)$ independently using multi-class SVM [40]. ii) learning the model weights w for the full model in a max-margin framework as follows. Given a set of N training videos (x^n, y^n), $n = 1, \ldots, N$, where x^n is the observations from each video and y^n is a set of labels, we train the global weight w in a max-margin framework. Specifically, we employ the cutting plane training algorithm described in [17] to solve this optimization problem. We incorporate the inference algorithm described in Section 6.1 to obtain the most violated constraint in each iteration [17]. To improve computational efficiency, we train the model weights related to activity potentials first, and then train the model weights related to tracklet association using the learned activity models. Since there exists different number of variables A, I and C in the training set, we balance the loss by using different values 1, 10, and 100 for A, I and C, respectively. Also, the observation potentials $\Psi(A, O)$ and $\Psi(C, O)$ are reweighted to be within $[-1, 1]$.

8.1 Model Analysis

We visualize a subset of model weights learned using our training algorithm in Fig. 10. The figure demonstrates that

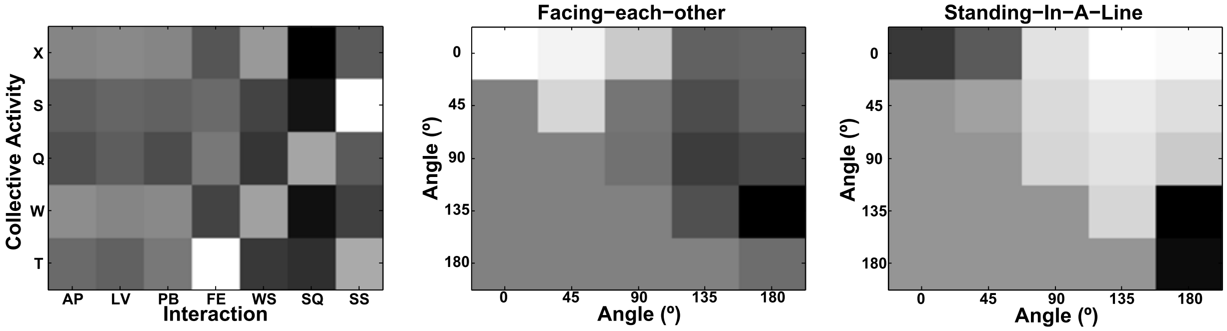


Fig. 10. Examples of learned weights using the proposed method. Lighter values indicate larger weights. *Left* Learned weights for $\Psi(C, I)$, with interactions on the *x*-axis and collective activities on the *y*-axis (see Section 9 for interaction and activity labels). As an example, note that the interaction "facing each other' has a very larger weight associated with "talking" (T) and a low weight associated with "walking." This agrees with our intuition that people are very likely to be facing each other when they are conversing, but will probably face the same direction when they are walking together. *Middle* Learned weights for $\psi_p(A, A, I)$, with the interaction "facing each other." The axes indicate the direction each person is facing. The highest weights are centered around the region corresponding to having two people in front of the other (0 degree in the person's view point). *Right* Learned weights for $\psi_p(A, A, I)$, with the interaction "standing in a queue." The highest weights correspond to people facing the same direction. *Middle* and *Right* demonstrate that our system has successfully learned that when people are facing each other they are most likely facing opposite directions, and when they are standing in a queue they will face the same direction.

our algorithm can capture meaningful relationships between variables at different levels in the hierarchy.

9 Experimental Validation

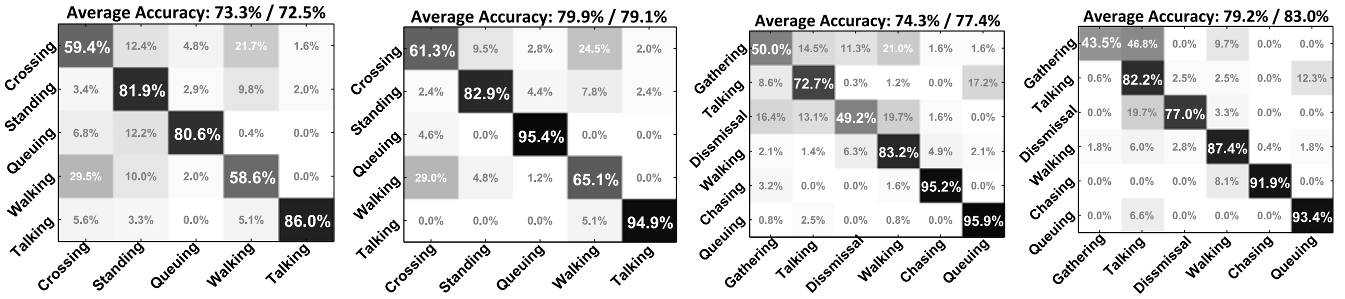
Implementation Details. Our algorithm assumes that the inputs O are available. These inputs are composed of collective activity features, tracklets, appearance feature, and spatio-temporal features as discussed in Section 3.1. Given a video, we obtain tracklets using a tracking method such as [5]. Once tracklets O are obtained, we compute two visual features (the histogram of oriented gradients (HoG) decriptors [10] and the bag of video words histogram [11]) in order to classify poses and actions, respectively. The HoG is extracted from an image region within the bounding box of the tracklets and the BoV is constructed by computing the histogram of video-words within the spatio-temporal volume of each tracklet. To obtain the video-words, we apply PCA (with 200 dimensions) and the k-means algorithm (100 codewords) on the cuboids obtained by [11]. Finally, the collective activity features are computed using the STL descriptor as discussed in Section 4 computed over tracklets and pose classification estimates. We construct our STL descriptor with following parameters: 8 meters for maximum radius and 60 frames for the temporal support. Since we are interested in labelling one collective activity per one time slice (i.e., a set of adjacent time frames), we take the average of all collected STL in the same time slice to generate an observation for C. In addition, we append the mean of the HoG descriptors obtained from all people in the scene to encode the shape of people in a certain activity. Instead of directly using raw features from HoG, BoV, and STL, we train multiclass SVM classifiers [17] for each of the observations to keep the size of parameters within a reasonable bound. In the end, each of the observation features is represented as a $|\mathcal{P}|$, $|\mathcal{A}|$, and $|\mathcal{C}|$ dimensional features, where each dimension of the features is the classification score given by the SVM classifier. In the experiments, we use the SVM response for C as a baseline method (Table 1 and Fig. 11).

Given tracklets and associated pose/action features O, a temporal sequence of atomic activity variables A_i is assigned to each tracklet τ_i. For each pair of coexisting A_i and A_j, I_{ij} describes the interaction between the two. Since I is defined over a certain temporal support ($\triangle t_I$), we subsample every 10th frames to assign an interaction variable. Finally, one C variable is assigned in every 20 frames with a temporal support $\triangle t_C$. We present experimental results using different choices of $\triangle t_I$ and $\triangle t_C$, (Table 2). Given tracklets and observations (O and O_C), the classification and

TABLE 1
Comparison of Collective and Interaction Activity Classification for Different Versions of Our Model Using the Data Set [7] (Left Column) and the Newly Proposed Dataset (Right Column)

	Dataset [7]				New Dataset			
Method	Ovral (C)	Mean (C)	Ovral (I)	Mean (I)	Ovral (C)	Mean (C)	Ovral (I)	Mean (I)
without O_C	38.7	37.1	40.5	37.3	59.2	57.4	49.4	41.1
no edges between C and I	67.7	68.2	42.8	37.7	67.8	54.6	42.4	32.8
no temporal chain	66.9	66.3	42.6	33.7	71.1	68.9	41.9	46.1
no temporal chain between C	74.1	75.0	54.2	48.6	77.0	76.1	**55.9**	**48.6**
full model ($\triangle t_C = 20, \triangle t_I = 25$)	**79.0**	**79.6**	**56.2**	**50.8**	**83.0**	**79.2**	53.3	43.7
baseline	72.5	73.3	-	-	77.4	74.3	-	-

The models we compare here are: i) Graph without O_C*. We remove observations provided by the STL descriptor (Section 4) for the collective activity. ii)* Graph with no edges between C and I*. We cut the connections between variables C and I and produce separate chain structures for each set of variables. iii)* Graph with no temporal edges*. We cut all the temporal edges between variables in the graphical structure and leave only hierarchical relationships. iv)* Graph with no temporal chain between C variables*. v) Our full model shown in Fig. 1d and vi) baseline method. The baseline method is obtained by taking the max response from the collective activity observation (O_C).*



(a) baseline (b) ours (c) baseline (d) ours

Fig. 11. (a) and (b) show the confusion table for collective activity using baseline method (SVM response for C) and proposed method on data set [7], respectively. (c) and (d) compare the two methods on newly proposed data set. In both cases, our full model improves the accuracy significantly over the baseline method. The numbers on top of each table show *mean-per-class* and *overall* accuracies.

target association take about a minute per video in our experiments.

Data Sets and Experimental Setup We present experimental results on the public data set [7] and a newly proposed data set. The first data set is composed of 44 video clips with annotations for five collective activities (*crossing*, *waiting*, *queuing*, *walking*, and *talking*) and eight poses (*right*, *right-front*, . . ., *right-back*). In addition to these labels, we annotate the target correspondence, action labels and interaction labels for all sequences. We define the eight types of interactions as *approaching*, *leaving* (LV), *passing-by* (PB), *facing-each-other*, *walking-side-by-side* (WS), *standing-in-a-row*, *standing-side-by-side* and *no-interaction* (NA). The categories of atomic actions are defined as: *standing* and *walking*. Due to a lack of standard experimental protocol on this data set, we adopt two experimental scenarios. First, we divide the whole set into four subsets without overlap of videos and perform four-fold training and testing. Second, we divide the set into separate training and testing sets as suggested by [21]. Since the first scenario provides more data to be analysed, we run the main analysis with the first scenario and use the second for comparison against [21]. In the experiments, we use the tracklets provided on the website of the authors of [5], [7].

The second data set is composed of 32 video clips with six collective activities: *gathering*, *talking*, *dismissal*, *walking together*, *chasing*, *queueing*. For this data set, we define nine interaction labels: *approaching* , *walking-in-opposite-direction*, *facing-each-other*, *standing-in-a-row*, *walking-side-by-side*, *walking-one-after-the-other* (WR), *running-side-by-side* (RS), *runn ing-one-after-the-other* (RR), and *no-interaction*. The atomic actions are labelled as *walking*, *standing still*, and *running*. We define eight poses similarly to the first data set. We divide the whole set into three subsets and run three-fold training and testing. For this data set, we obtain the tracklets using [31] and create back projected 3D trajectories using the simplified camera model [15].

Results and Analysis. We analyze the behavior of the proposed model by disabling the connectivity between various variables of the graphical structure (see Table 1 and Fig. 11 for details). We study the classification accuracy of collective activities C and interaction activities I. As seen in the Table 1, the best classification results are obtained by our full model. Since the data set is unbalanced, we present both overall accuracy and mean-per-class accuracy, denoted as Ovral and Mean in Tables 1 and 2, respectively. We observe that our full model also obtains better or similar accuracy in atomic pose and action classification. Our full model achieves $45.3/39.2$ percent in pose classification (overall/mean) and $89.4/87.9$ percent in action classification using data set [7] while the baseline individual pose and action classifier achieves $42.6/38.8$ and $89.8/89.1$ percent, respectively.

Next, we analyse the model by varying the parameter values that define the temporal supports of collective and interaction activities ($\triangle t_C$ and $\triangle t_I$). We run different experiments by fixing one of the temporal supports to a reference value and change the other. As any of the temporal supports becomes larger, the collective and interaction activity variables are connected with a larger number of interactions and atomic activity variables, respectively, which provides richer coupling between variables across labels of the hierarchy and, in turn, enables more robust classification results (Table 2). Notice that, however, by increasing connectivity, the graphical structure becomes more complex and thus inference becomes less manageable.

Since previous works adopt different ways of calculating the accuracy of the collective activity classification, a direct

TABLE 2
Comparison of Classification Results Using Different Lengths of Temporal Support $\triangle t_C$ and $\triangle t_I$ for Collective and Interaction Activities, Respectively

	Dataset [7]				New Dataset			
Method	Ovral (C)	Mean (C)	Ovral (I)	Mean (I)	Ovral (C)	Mean (C)	Ovral (I)	Mean (I)
$\triangle t_C = 30, \triangle t_I = 25$	79.1	79.9	56.1	50.8	80.8	77.0	**54.3**	**46.3**
$\triangle t_C = 20, \triangle t_I = 25$	79.0	79.6	**56.2**	**50.8**	**83.0**	**79.2**	53.3	43.7
$\triangle t_C = 10, \triangle t_I = 25$	77.4	78.2	56.1	50.7	81.5	77.6	52.9	41.8
$\triangle t_C = 30, \triangle t_I = 15$	76.1	76.7	52.8	40.7	80.7	71.8	48.6	34.8
$\triangle t_C = 30, \triangle t_I = 5$	**79.4**	**80.2**	45.5	36.6	77.0	67.3	37.7	25.7

Notice that in general larger support provides more stable results.

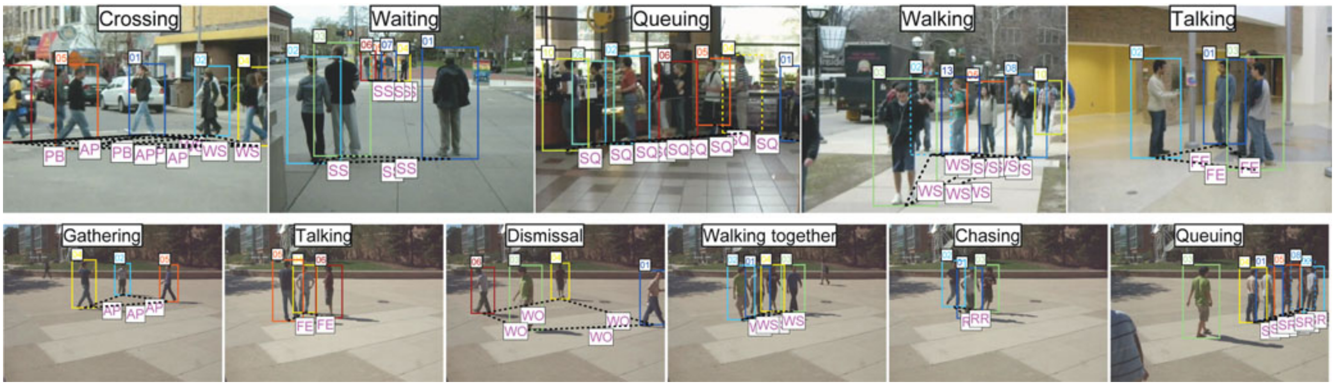


Fig. 12. Anecdotal results on different types of collective activities. In each image, we show the collective activity estimated by our method. Interactions between people are denoted by the dotted line that connects each pair of people. To make the visualization more clear, we only show interactions that are not labelled as NA. Anecdotal results on the data set [7] and the newly proposed data set are shown on the top and bottom rows, respectively. Our method automatically discovers the interactions occurring within each collective activity; eg., *walking-side-by-side* occurs with *crossing* or *walking*, whereas *standing-side-by-side* occurs with *waiting*. See text for the definition of other acronyms.

TABLE 3
Quantitative Tracking Results and Comparison with Baseline Methods (See Text for Definitions)

	Match (baseline)	*Linear* (partial model)	*Quadratic* (full model)	*Linear GT*	*Quad. GT*	*Tracklet*
Dataset [7]	1109/28.73%	974/37.40%	894/42.54%	870/44.09%	736/52.70%	1556/0%
New Dataset	110/81.79%	107/82.28%	104/82.78%	97/83.94%	95/84.27%	604/0%

Each cell of the table shows the number of match errors and match error correction rate (MECR) $\frac{\#\ error\ in\ tracklet\ -\ \#\ error\ in\ result}{\#\ error\ in\ tracklet}$ *of each method, respectively. Since we focus on correctly associating each tracklet with another, we evaluate the method by counting the number of errors made during association (rather than detection-based accuracy measurements such as recall, FPPI, etc) and MECR. An asssociation error is defined for each possible match of a tracklet (thus at most two per tracklets, previous and next match). This measure can effectively capture the amount of fragmentization and identity switches in association. In the case of a false alarm tracklet, any association with this track is considered to be an error.*

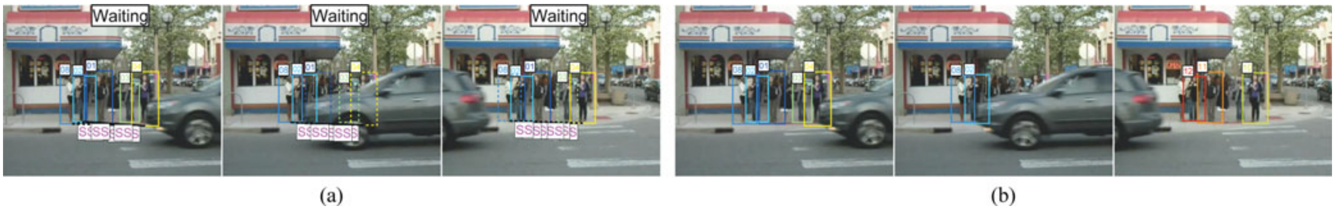


Fig. 13. The discovered interaction *standing-side-by-side* helps to keep the identity of tracked individuals after an occlusion. Notice the complexity of the association problem in this example. Due to the proximity of the targets and similarity in color, the *Match* method (b) fails to keep the identity of targets. However, our method (a) finds the correct match despite the challenges. The input tracklets are shown as a solid box and associated paths are shown in dotted box.

comparison of the results may not be appropriate. Choi et al. [7] and Choi et al. [8] adopt a leave-one-video-out training/testing scheme and evaluate per-person collective activity classification. Lan et al. [21] train the model on three fourths of the data set, test on the remaining fourth and evaluate per-scene collective activity classification. To compare against the baseline collective activity classifiers discussed in Section 4 and introduced in [7], [8], we assign the per-scene collective activity labels that we obtain with four-fold experiments to each individual. We obtain an accuracy of 74.4 percent which is superior than 65.9 and 70.9 percent obtained by STL and RSTV classifier respectively (Section 4). These results were also reported in [7] and [8]. In addition, we run the experiments on the same training/testing split of the data set suggested by [21] and achieve competitive accuracy (80.4 overall and 75.7 percent mean-per-class compared to 79.1 overall and 77.5 percent mean-per-class, respectively, reported in [21]). Anecdotal results are shown in Fig. 12.

Table 3 summarizes the tracklet association accuracy of our method. In this experiment, we test three different algorithms for tracklet matching: pure match, linear model, and full quadratic model. *Match* represents the max-flow method without interaction potential (only appearance, motion and detection scores are used). *Linear* model represents our model where the quadratic relationship is ignored and only the linear part of the interaction potentials is considered (e.g., those interactions that are involved in selecting only one path). The *Quadratic* model represents our full Branch-and-Bound method for target association. The estimated activity labels are assigned to each variable for the two methods. We also show the accuracy of association when ground truth (GT) activity labels are provided, in the fourth and fifth columns of the table. The last column shows the number of association errors in the initial input tracklets. In these experiments, we adopt the same four fold training/testing and three fold training/testing for the data set [7] and newly proposed data set, respectively. Note that, in the data set [7], there exist 1,821 tracklets with 1,556 match errors in total. In the new data set, which includes much less crowded sequences than [7], there exist 474 tracklets with 604 errors in total. As the Table 3 shows, we achieve significant improvement over baseline method

(*Match*) using the data set [7] as it is more challenging and involves a large number of people (more information from interactions). On the other hand, we observe a smaller improvement in matching targets in the second data set, since it involves few people (typically $2 \sim 3$) and is less challenging (note that the baseline (*Match*) already achieves 81 percent correct match). Experimental results obtained with ground truth activity labels (*Linear GT* and *Quad. GT*) suggest that better activity recognition would yield more accurate tracklet association. Anecdotal results are shown in Fig. 13. When the interactions are less structured (e.g., passing by or leaving), we observe that mistakes in recognizing interactions often produce more errors in the tracklet association.

The training procedure takes 24 hours to learn all the model parameters and the testing algorithm takes typically several minutes to process a video excluding the feature extraction and tracklet generation process.

10 Conclusion

In this paper, we have presented a new framework to coherently identify target associations and classify collective activities as well as the novel concept, *crowd context*, that encodes the essential contextual information for collective activity recognition. We have demonstrated that collective activities provide critical contextual cues for making target association more robust and stable; in turn, the estimated trajectories as well as atomic activity labels allow the construction of more accurate interaction and collective activity models. As a future work, we aim to introduce a *no activity* class to separate interesting activities from not interesting activities in real world videos. Also, introducing latent interaction variables in the training procedure would make the system more scalable to a large number of collective activities and avoid biases in assigning interaction labels.

Acknowledgments

The authors acknowledge the support of the ONR grant N000141110389, DARPA UPSIDE award HR0011-13-2-0016 and Toyota. They appreciate Khuram Shahid for his contribution to this project.

References

[1] M.R. Amer, D. Xie, M. Zhao, S. Todorovic, and S.-C. Zhu, "Cost-Sensitive Top-Down/Bottom-Up Inference for Multiscale Activity Recognition," *Proc. 12th European Conf. Computer Vision (ECCV)*, pp. 187-200, 2012.

[2] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509-522, Apr. 2002.

[3] L. Breiman, and A. Cutler, "Random Forest," [online], marzec 2004.

[4] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001.

[5] W. Choi and S. Savarese, "Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera," *Proc. 11th European Conf. Computer Vision (ECCV)*, Sept. 2010.

[6] W. Choi and S. Savarese, "A Unified Framework for Multi-Target Tracking and Collective Activity Recognition," *Proc. 12th European Conf. Computer Vision (ECCV)*, 2012.

[7] W. Choi, K. Shahid, and S. Savarese, "What Are They Doing?: Collective Activity Classification Using Spatio-Temporal Relationship Among People," *Proc. Workshop Visual Surveillance (VSWS)*, 2009.

[8] W. Choi, K. Shahid, and S. Savarese, "Learning Context for Collective Activity Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[9] F. Cupillard, F. Brémond, and M. Thonnat, "Group Behavior Recognition with Multiple Cameras," *Proc. IEEE Sixth Workshop Applications of Computer Vision (WACV)*, pp. 177-183, 2002.

[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[11] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features," *Proc. IEEE Second Joint Int'l Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2005.

[12] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "A Mobile Vision System for Robust Multi-Person Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[13] P. Felzenszwalb and D. Huttenlocher, "Efficient Belief Propagation for Early Vision," *Int'l J. Computer Vision*, vol. 70, no. 1, pp. 41-54, 2006.

[14] A. Gupta, P. Srinivasan, J. Shi, and L.S. Davis, "Understanding Videos, Constructing Plots Learning a Visually Grounded Storyline Model From Annotated Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2012-2019, 2009.

[15] D. Hoiem, A.A. Efros, and M. Herbert, "Putting Objects in Perspective," *Int'l J. Computer Vision*, vol. 80, no. 1, pp. 3-15, 2008.

[16] S. Intille and A. Bobick, "Recognizing Planned, Multiperson Action," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 414-445, 2001.

[17] T. Joachims, T. Finley, and C.-N. Yu, "Cutting-Plane Training of Structural SVMs," *Machine Learning*, vol. 77, 2009.

[18] S.M. Khan and M. Shah, "Detecting Group Activities Using Rigidity of Formation," *Proc. ACM 13th Ann. Int'l Conf. Multimedia*, pp. 403-406, 2005.

[19] Z. Khan, T. Balch, and F. Dellaert, "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805-1819, 2005.

[20] T. Lan, L. Sigal, and G. Mori, "Social Roles in Hierarchical Models for Human Activity Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1354-1361, 2012.

[21] T. Lan, Y. Wang, W. Yang, and G. Mori, "Beyond Actions: Discriminative Models for Contextual Group Activities," *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2010.

[22] A.H. Land and A.G. Doig, "An Automatic Method of Solving Discrete Programming Problems," *Econometrica*, vol. 28, 1960.

[23] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn, "Everybody Needs Somebody: Modeling Social and Grouping Behavior on a Linear Programming Multiple People Tracker," *Proc. IEEE Int'l Conf. Computer Vision Workshop Modeling, Simulation and Visual Analysis of Large Crowds*, 2011.

[24] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, A Tutorial on Energy-Based Learning, MIT Press, 2006.

[25] R. Li, R. Chellappa, and S.K. Zhou, "Learning Multi-Modal Densities on Discriminative Temporal Interaction Manifold for Group Activity Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[26] J. Liu, J. Luo, and M. Shah, "Recognizing Realistic Actions from Videos 'in the Wild'," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[27] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event Detection and Analysis from Video Streams," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 873-889, Aug. 2001.

[28] B. Ni, S. Yan, and A. Kassim, "Recognizing Human Group Activities with Localized Causalities," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[29] J.C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," *Int'l J. Computer Vision*, vol. 79, pp. 299-318, 2008.

[30] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, 2009.

[31] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[32] A. O. Ramin Mehran and M. Shah, “Abnormal Crowd Behavior Detection Using Social Force Model,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[33] M. Rodriguez, S. Ali, and T. Kanade, “Tracking in Unstructured Crowded Scenes,” *Proc. IEEE Int’l Conf. Computer Vision (ICCV)*, 2009.

[34] M. S. Ryoo and J. K. Aggarwal, “Spatio-Temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities,” *Proc. IEEE Int’l Conf. Computer Vision (ICCV)*, 2009.

[35] M.S. Ryoo and J.K. Aggarwal, “Stochastic Representation and Recognition of High-Level Group Activities,” *Int’l J. Computer Vision*, vol. 93, pp. 183-200, 2010.

[36] S. Savarese, A. DelPozo, J. Niebles, and L. Fei Fei, “Spatial-Temporal Correlatons for Unsupervised Action Classification,” *Proc. IEEE Workshop Motion and Video Computing (WMVC)*, 2008.

[37] P. Scovanner and M. Tappen, “Learning Pedestrian Dynamics from the Real World,” *Proc. IEEE Int’l Conf. Computer Vision (ICCV)*, 2009.

[38] V.K. Singh, B. Wu, and R. Nevatia, “Pedestrian Tracking by Associating Tracklets Using Detection Residuals,” *Proc. IEEE Workshop Motion and Video Computing (IMVC)*, 2008.

[39] E. Swears and A. Hoogs, “Learning and Recognizing Complex Multi-Agent Activities with Applications to American Football Plays,” *Proc. IEEE Workshop Applications of Computer Vision (WACV)*, 2011.

[40] J. Weston, and C. Watkins, *Multi-Class Support Vector Machines*, Technical Report CSD-TR-98-04, University of London, 1998.

[41] B. Wu and R. Nevatia, “Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Based Part Detectors,” *Int’l J. Computer Vision*, vol. 75, pp. 247-266, 2007.

[42] K. Yamaguchi, A.C. Berg, T. Berg, and L. Ortiz, “Who Are You With and Where Are You Going?” *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[43] J.Y. Yen, “Finding the K Shortest Loopless Paths in a Network,” *Management Science*, vol. 17, pp. 712-716, July 1971.

[44] L. Zhang, Y. Li, and R. Nevatia, “Global Data Association for Multi-Object Tracking Using Network Flows,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[45] B. Zhou, X. Wang, and X. Tang. “Understanding Collective Crowd Behaviors: Learning a Mixture Model of Dynamic Pedestrian Agents, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2871-2878, 2012.

Wongun Choi received the MS and PhD degrees in electrical and computer engineering from the University of Michigan, Ann Arbor, in 2011 and 2013, respectively. He is currently a research scientist at NEC Laboratories. His research interests include object tracking, object detection, scene understanding and activity recognition. He co-organized the First IEEE Workshop on Understanding Human Activities: Context and Interaction in conjunction with the ICCV 2013.

Silvio Savarese received the PhD degree in electrical engineering from the California Institute of Technology in 2005 and was a Beckman Institute fellow at the University of Illinois at Urbana-Champaign from 2005 to 2008. He is currently an assistant professor of computer science at Stanford University. He joined Stanford in 2013 after being assistant and then associate professor (with tenure) of electrical and computer engineering at the University of Michigan, Ann Arbor, from 2008 to 2013. His research interests include computer vision, object recognition and scene understanding, shape representation and reconstruction, human activity recognition and visual psychophysics. He received several awards including the James R. Croes Medal in 2013, a TRW Automotive Endowed Research Award in 2012, a US National Science Foundation (NSF) Career Award in 2011 and Google Research Award in 2010. In 2002 he was awarded the Walker von Brimer Award for outstanding research initiative.