Cost-Sensitive Top-down/Bottom-up Inference for Multiscale Activity Recognition

Mohamed R. Amer[1], Dan Xie[2], Mingtian Zhao[2],
Sinisa Todorovic[1], and Song-Chun Zhu[2]

[1]Oregon State University, Corvallis, Oregon
{amerm,sinisa}@onid.orst.edu
[2]University of California, Los Angeles, California
{xiedan,mtzhao,sczhu}@ucla.edu

Abstract. This paper addresses a new problem, that of multiscale activity recognition. Our goal is to detect and localize a wide range of activities, including individual actions and group activities, which may simultaneously co-occur in high-resolution video. The video resolution allows for digital zoom-in (or zoom-out) for examining fine details (or coarser scales), as needed for recognition. The key challenge is how to avoid running a multitude of detectors at all spatiotemporal scales, and yet arrive at a holistically consistent video interpretation. To this end, we use a three-layered AND-OR graph to jointly model group activities, individual actions, and participating objects. The AND-OR graph allows a principled formulation of efficient, cost-sensitive inference via an explore-exploit strategy. Our inference optimally schedules the following computational processes: 1) direct application of activity detectors – called α process; 2) bottom-up inference based on detecting activity parts – called β process; and 3) top-down inference based on detecting activity context – called γ process. The scheduling iteratively maximizes the log-posteriors of the resulting parse graphs. For evaluation, we have compiled and benchmarked a new dataset of high-resolution videos of group and individual activities co-occurring in a courtyard of the UCLA campus.

1 Introduction

This paper addresses a new problem. Our goal is to detect and localize all instances of a queried human activity present in high-resolution video. The novelty of this problem is two-fold: (i) the queries can be about a wide range of activities, including actions of individuals, their interactions with objects and other people, or collective activities of a group of people; and (ii) all these various types of activities may simultaneously co-occur in a relatively large scene captured by high-resolution video. The video resolution allows for digital zoom-in (or zoom-out) for examining fine details (or coarser scales), as needed for recognition. We call this problem multiscale activity recognition.

With the recent rapid increase in the spatial resolution of digital cameras, and growing capabilities of capturing long video footage, the problem of multiscale activity recognition becomes increasingly important for many applications, including video surveillance and monitoring. While recent work typically focuses on short videos of

a particular activity type, there is an increasing demand for developing principled approaches to interpreting long videos of spatially large, complex scenes with many people engaged in various, co-occurring, individual and group activities. The key challenge of this new problem is complexity of inference. It is infeasible to apply sliding windows for detecting all activity instances at all spatiotemporal scales of the video volume.

To address the above challenge, we account for the compositional nature of human activities, and model them explicitly with the AND-OR graph [1–3]. The AND-OR graph is suitable for our purposes, because it is capable of compactly representing many activities, each recursively defined in terms of spatial layouts of human-human or human-object interactions. Modeling the temporal structure of activities is left for the future work. The recursion ends with primitive body parts and objects. Also, its hierarchical structure allows for a principled formulation of cost-sensitive inference. Our formulation rests on two computational mechanisms. First, following the work of [4], we express inference in terms of the α, β, and γ processes. The three processes are specific to each node in the AND-OR graph, where

1. α(node): detecting the activity directly from video features extracted from the video part associated with the node;
2. β(node): bottom-up binding of parts of the activity represented by the node;
3. γ(node): prediction of the activity represented by the node from the context provided by a parent node.

Second, we specify an explore-exploit (E^2) strategy for cost-sensitive inference. The E^2 strategy optimally schedules the sequential computation of α, β, and γ, such that the log-posteriors of the resulting parse graphs are maximized. In this way, the E^2 strategy digitally zooms-in or zooms-out at every iteration, conditioned on previous moves, and thus resolves ambiguities in all hypothesized parse graphs.

To initiate research on this important problem, we have collected and annotated a new dataset of high-resolution videos of various, co-occurring activities taking place in a courtyard of the UCLA campus [5]. Fig. 1 shows an example, cropped out frame from our UCLA Courtyard dataset. As can be seen, the cropped-out part shows a vast space wherein students are standing in a line to buy food, walking together in a campus tour led by a guide, or sitting and reading on the staircase. In other parts of the same video (not shown), people may be riding bicycles or scooters, buying soda from a vending machine, or jogging together. The video has a high resolution to allow activity recognition at different spatial and temporal scales. For example, it may be necessary to exploit the high resolution for digital zoom-in, and thus disambiguate particular objects defining the queried activity (e.g., buying a soda or a snack from the vending machine).

Prior Work – Multiscale activity recognition has received scant attention in the literature. Recent work typically studies prominently featured, single-actor, punctual or repetitive actions [6]. Activities with richer spatiotemporal structure have been addressed using graphical models, including Deformable Action Templates [7], Sum Product Networks [8], and AND-OR graphs [2, 3]. However, this work considers only one specific scale of human activities. Our work is related to recent methods for recognizing group and individual activities using context [9–11], and identifying objects in videos based on activity recognition [12]. There are two major differences. First, that work

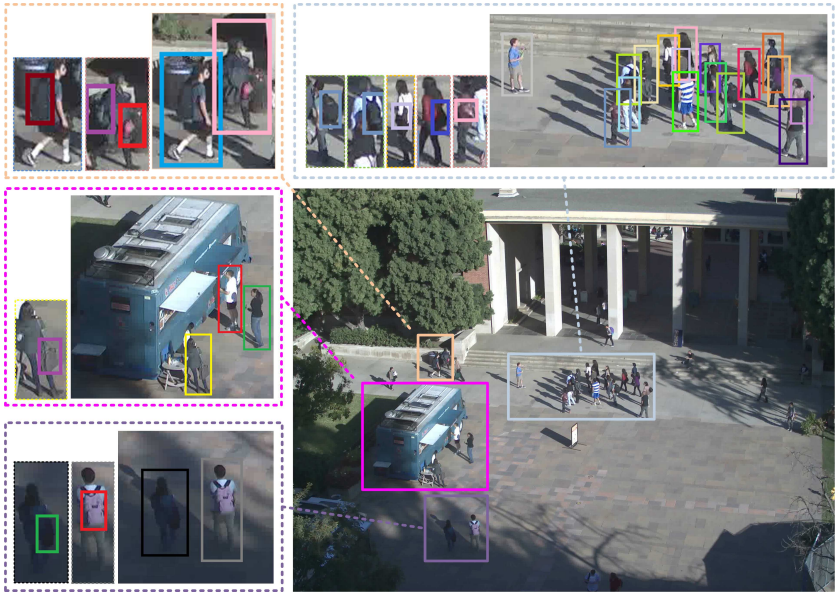


Fig. 1. An example from our UCLA Courtyard dataset, showing multiple co-occurring group activities, primitive actions, and objects. Overlaid over the original frame, the purple marks the group walking together, the magenta marks the group standing in a line for food, the beige marks the group going to class, and the light blue marks the UCLA Courtyard tour. Within the dashed boxes, we show that each of these group activities consists of individual actions of group participants, where some of them interact with objects, e.g., carry backpacks.

considers only two semantic levels – namely, either context and activities, or activities and objects. We jointly consider three semantic levels: objects, individual actions, and group activities. Second, prior work typically focuses on simple videos showing a single activity (or object) in the entire video. Our high-resolution videos, instead, show a spatially large scene with multiple co-occurring activities of many people interacting with many objects over a relatively long time interval. We advance recent work on localizing single-actor, punctual, and repetitive activities [13] by parsing significantly more challenging videos with co-occurring activities at different scales.

Our work builds upon an empirical study of the α, β, and γ process for face detection in still images, presented in [4]. That work considered only one object class (i.e., faces), whereas we seek to recognize a multitude of activity and object classes. Our extensions include: (i) a new formulation of the expected gains of α, β, and γ, and specifying the E^2 strategy for cost-sensitive inference of the AND-OR graph.

In the sequel, Sec. 2 defines the AND-OR graph. Sec. 3 presents our inference. Sec. 4 specifies low-level detectors used in inference, and the computation of α, β, and γ. Sec. 5 formulates the E^2 strategy. Sec. 6 specifies our learning. Sec. 7 presents our experimental evaluation.

2 AND-OR Graph

This section presents the AND-OR graph following the notation and formalism presented in [4]. The AND-OR graph, illustrated in Fig. 2, organizes domain knowledge in a hierarchical manner at three levels. Group activities, $a \in \mathcal{A}$, (e.g., Standing-in-a-line) are defined as a spatial relationship of a set of primitive actions (e.g., a group of people Standing, in a certain Pose, Orientation, and at certain Displacement). They are represented by nodes at the highest level of the graph. Primitive actions, $r \in \mathcal{R}$, (e.g., Riding-a-bike) are defined as punctual or repetitive motions of a single person, who may interact with an object (e.g., Bike or Phone). They are represented as children nodes of the group-activity nodes. Objects, $o \in \mathcal{O}$, include body parts and tools or instruments that people interact with while conducting a primitive action. Object nodes are placed at the lowest level of the AND-OR graph, and represent children of the primitive-action nodes. Modeling efficiency is achieved by sharing children nodes among multiple parents, where AND nodes encode particular configurations of parts, and OR nodes account for alternative configurations.

More formally, the AND-OR graph is $\mathcal{G} = (\mathcal{V}_{NT}, \mathcal{V}_T, \mathcal{E}, \mathcal{P})$, where $\mathcal{V}_{NT}$ is a union set of non-terminal AND and OR nodes. An AND node is denoted as $\wedge$, and an OR node is denoted as $\vee$. Let $l = 1, ..., L$ denote a level in $\mathcal{G}$, where $l-1$ is the level closer to the root than level l. Then, a parent of $\wedge^l$ is denoted as $\wedge^{l-}$. Similarly, ith child of $\wedge^l$ is denoted as $\wedge_i^{l+}$. We also use $X_{\wedge^l}$ to denote a descriptor vector of the video part associated with node $\wedge^l$, including the information about location, scale and orientation relative to the video part associated with the parent node $\wedge^{l-1}$. $\mathcal{V}_T = \{t_{\wedge_i} : \forall \wedge_i \in \mathcal{V}_{NT}\}$ is a set of terminal nodes connected to the corresponding non-terminal nodes, where each $t_{\wedge_i}$ represents a detector applied to the video part associated with $\wedge_i$. $\mathcal{E}$ is a set of edges of $\mathcal{G}$. A parse graph, pg, is a valid instance of the grammar $\mathcal{G}$. $\mathcal{P}$ is the probability over the space of all parse graphs. The edge set of a parse graph is a union of switching edges $\mathcal{E}_{\text{switch}}(\text{pg})$, decomposition edges $\mathcal{E}_{\text{dec}}(\text{pg})$, and relation edges $\mathcal{E}_{\text{rel}}(\text{pg})$, $\mathcal{E}(\text{pg}) = \mathcal{E}_{\text{switch}}(\text{pg}) \cup \mathcal{E}_{\text{dec}}(\text{pg}) \cup \mathcal{E}_{\text{rel}}(\text{pg})$, as explained below.

The prior probability of a parse graph is defined as $p(\text{pg}) = \frac{1}{Z}\exp(-E(\text{pg}))$, where the partition function is $Z = \sum_{\text{pg}} \exp(-E(\text{pg}))$, and the total energy is

$$E(\text{pg}) = -\sum_l \Big[\sum_{(\vee^l, \wedge^l) \in \mathcal{E}_{\text{switch}}(\text{pg})} \log p(\wedge^l | \vee^l) + \sum_{(\wedge^l, \wedge^{l-}) \in \mathcal{E}_{\text{dec}}(\text{pg})} \log p(X_{\wedge^l} | X_{\wedge^{l-}}) + \sum_{(\wedge_i^{l+}, \wedge_j^{l+}) \in \mathcal{E}_{\text{rel}}(\text{pg})} \log p(X_{\wedge_i^{l+}}, X_{\wedge_j^{l+}}) \Big]. \tag{1}$$

In (1), the first term denotes the probability that OR node $\vee^l$ selects AND node $\wedge^l$, the second term defines parent-child statistical dependencies, and the third term defines pairwise dependencies between pairs of children of $\wedge^l$.

Given an input video frame, I, with domain defined on lattice Λ, the likelihood of a parse graph is defined as $p(I|\text{pg}) = \prod_{t \in \mathcal{V}_T(\text{pg})} p(I_{\Lambda_t}|t)$, where $\Lambda_t \in \Lambda$ is video domain occupied by the terminal node t.

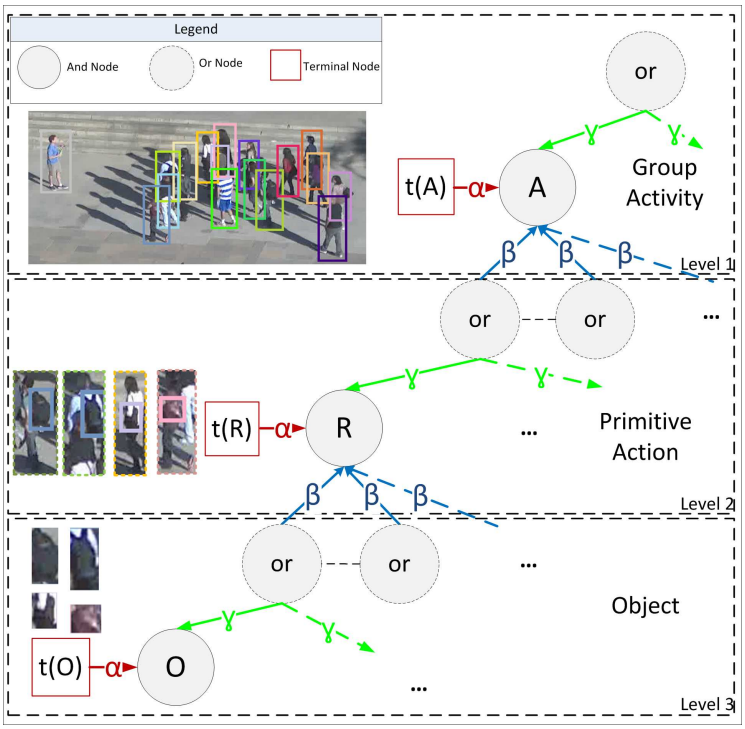


Fig. 2. The AND-OR graph of group activities $\mathcal{A}$, primitive actions $\mathcal{R}$, and objects $\mathcal{O}$. t is the terminal node representing a detector of the corresponding activity or object. Detector responses $t(\cdot)$ constitute the α process. The top-down γ process is aimed at predicting and localizing the corresponding primitive action (or object), based on context provided by the detected group activity (or primitive action). The bottom-up β process is aimed at inferring the corresponding primitive action (or group activity), based on detections of participating objects (or primitive actions).

3 Inference

Given a video, we conduct inference frame by frame. Temporal characteristics of activities are implicitly accounted for via descriptor vectors, which collect visual cues from space-time windows centered around spatial domains, $\Lambda_t \in \Lambda$, occupied by every terminal node t. Similar to the derivation in [4], the video frame, I_Λ, contains an unknown number, K, of instances of the queried activities at different spatial scales. Each inferred instance is represented by a parse graph in the world representation, $W = (K, \{\text{pg}_k : k = 1, 2, \ldots, K\})$. Under the Bayesian framework, we infer W by maximizing its posterior probability, $W^* = \arg\max_{W \in \Omega} p(W)p(I_\Lambda|W)$, where Ω is the space of solutions.

The prior of W is defined as $p(W) = p(K) \prod_{k=1}^{K} p(\text{pg}_k)$, where $p(K) \propto \exp(-\lambda_0 K)$ is the prior of the number of parse graphs, and $p(\text{pg}_k)$ is defined by (1). To compute the likelihood $p(I_\Lambda|W)$, we define foreground lattice $\Lambda_{\text{fg}} = \cup_k \Lambda_{\text{pg}_k}$, and background

lattice $\Lambda_{\text{bg}} = \Lambda \setminus \Lambda_{\text{fg}}$, and use a generic background pdf, $q(I)$, as

$$p(I_\Lambda|W) = p(I_{\Lambda_{\text{fg}}}|W)q(I_{\Lambda_{\text{bg}}})\frac{q(I_{\Lambda_{\text{fg}}})}{q(I_{\Lambda_{\text{fg}}})} = q(I_\Lambda)\prod_{k=1}^{K}\frac{p(I_{\Lambda_{\text{pg}_k}}|\text{pg}_k)}{q(I_{\Lambda_{\text{pg}_k}})} \tag{2}$$

where $p(I_{\Lambda_{\text{pg}_k}}|\text{pg}_k)$ means that domain Λ_{pg_k} is explained away by the parse graph pg_k, and $q(I_{\Lambda_{\text{pg}_k}})$ explains domain Λ_{pg_k} as background.

In inference, we sequentially infer the parse graphs, one at a time, and augment W. The inference of a parse graph is formulated as

$$\text{pg}^* = \arg\max_{\text{pg}\in\Omega(\text{pg})}\Big[\log p(\text{pg}) + \log\frac{p(I_{\Lambda_{\text{pg}}}|\text{pg})}{q(I_{\Lambda_{\text{pg}}})}\Big], \tag{3}$$

where $p(\text{pg})$ is defined by (1). The likelihood ratio in (3) can be factorized over terminal nodes, $t \in \mathcal{V}_T(\text{pg})$, representing detector responses over the corresponding video parts. Specifically, we can write $\log\frac{p(I_{\Lambda_{\text{pg}}}|\text{pg})}{q(I_{\Lambda_{\text{pg}}})} = \sum_{t\in\mathcal{V}_T(\text{pg})}\log\frac{p(p(I_{\Lambda_t}|t)}{q(I_{\Lambda_t})} = \sum_{t\in\mathcal{V}_T(\text{pg})}\psi(t)$, where $\psi(t)$ denotes the confidence of detector t applied at video part I_{Λ_t}. From (1) and (3), we have:

$$\begin{aligned}\text{pg}^* = \arg\max_{\text{pg}\in\Omega(\text{pg})}\sum_l\Big\{ &\underbrace{\log p(\wedge^l|\vee^l)}_{\text{AND-OR graph structure}} + \underbrace{\psi(t_{\wedge^l})}_{\alpha^l} + \underbrace{\Big[\underbrace{\psi(t_{\wedge^{l-}})}_{\alpha^{l-}} + \underbrace{\log p(X_{\wedge^l}|X_{\wedge^{l-}})}_{\gamma^{l-}}\Big]}_{\text{zoom-out}} \\ &\underbrace{+ p(N^l)\sum_{i=1}^{N^l}\Big[\underbrace{\log p(X_{\wedge_i^{l+}}|X_{\wedge^l})}_{\gamma_i^{l+}} + \underbrace{\psi(t_{\wedge_i^{l+}})}_{\alpha_i^{l+}} + \sum_{i\neq j}\underbrace{\log p(X_{\wedge_i^{l+}}, X_{\wedge_j^{l+}})}_{\beta_{ij}^{l+}}\Big]\Big\}}_{\text{zoom-in}}\end{aligned} \tag{4}$$

Equation (4) specifies the α^l, β^l, and γ^l processes at level l of the AND-OR graph. Confidences of the activity detectors constitute α^l process. The top-down γ^l process is aimed at predicting and localizing the corresponding primitive action (or object), based on the context of the group activity (or primitive action). For example, to zoom-out for examining the context of a primitive action, it is necessary to detect the action's contextual group activity, α^{l-}, and to estimate the likelihood of the corresponding parent-child configuration γ^{l-}. The bottom-up β^l process is aimed at inferring the corresponding group activity (or primitive action), based on its children primitive actions (or objects), and their configuration. For example, to zoom-in for examining individual actions within a group activity, it is first necessary to detect the primitive actions α_i^{l+}, $i = 1, ..., N^l$, then, estimate the likelihood of the corresponding parent-child configuration γ_i^{l+}, and finally estimate the likelihood of their configuration β_{ij}^{l+}, $i, j = 1, ..., N^l$.

4 Computing α, β, γ

For each level l of the AND-OR graph, we define a set of α^l detectors aimed at detecting corresponding activities. As the α's are independent across the three levels of our AND-OR graph, we specify three different types of detectors. All detectors have access to the

Deformable-Parts-Model (DPM) person detector [14], and a multiclass SVM classifier aimed at detecting a person's facing direction. The person detector is initially applied to each frame using the scanning procedure recommended in [14]. A person's facing direction is classified by an 8-class classifier, learned by LibSVM on HOGs (the 5-fold cross-validation precision of orientation is 69%).

For detecting objects, we train the DPM on bounding boxes of object instances annotated in training videos, and apply this detector in a vicinity of every people detection. For each object detection, we use the above SVM to identity the object's orientation.

For detecting primitive actions, we apply the motion-appearance based detector of [15] in a vicinity of every people detection. From a given window enclosing a person detection, we first extract motion-based STIP features [16], and describe them with HOG descriptors. Then, we extract KLT tracks of Harris corners, and quantize the motion vectors along the track to obtain a descriptor called the Sequence Code Map. The descriptors of STIPs and KLT tracks are probabilistically fused into a relative location probability table (RLPT), which captures the spatial and temporal relationships between the features. Such a hybrid descriptor is then classified by a multiclass SVM to detect the primitive actions of interest.

For detecting group activities, we compute the STV (Space-Time Volume) descriptors of [17] in a vicinity of every people detection, called an anchor. STV counts people, and their poses, locations, and velocities, in different space-time bins surrounding the anchor. Each STV is oriented along the anchor's facing direction. STVs calculated per frame are concatenated to capture the temporal evolution of the activities. Since the sequence of STVs captures a spatial variation over time, the relative motion and displacement of each person in a group is also encoded. Tracking STVs across consecutive frames is performed in 2.5D scene coordinates. This makes detecting group activities robust to perspective and view-point changes. The tracks of STVs are then classified by a multiclass SVM to detect the group activities of interest.

The β process binds pairs of children nodes $(\wedge_i^{l+}, \wedge_j^{l+})$ of parent $\wedge^l$. This is evaluated using the Gaussian distribution $p(X_{\wedge_i^{l+}}, X_{\wedge_j^{l+}}) = N(X_{\wedge_i^{l+}} - X_{\wedge_j^{l+}}; \mu_{\beta^l}, \Sigma_{\beta^l})$.

The γ process predicts ith child $\wedge_i^{l+}$ conditioned on the context of parent $\wedge^l$. This is evaluated using the Gaussian distribution $p(X_{\wedge_i^{l+}} | X_{\wedge^l}) = N(X_{\wedge_i^{l+}} - X_{\wedge^l}; \mu_{\gamma^l}, \Sigma_{\gamma^l})$.

5 The E^2 Strategy for Cost-sensitive Inference

The E^2 strategy optimally schedules a sequential computation of α, β, and γ processes, such that the posterior distributions of K parse graphs in W are iteratively maximized. We make the assumption that every process carries the same computational cost.

More formally, given a query, q, the E^2 strategy sequentially selects an optimal move at a given state, which results in another state. The set of states, $\mathbb{S}_q$, that can be visited are defined by all AND nodes which form the transitive closure of node $\wedge_q$ representing q in the AND-OR graph. Thus, a state $s \in \mathbb{S}_q$ represents an AND node in the transitive closure of $\wedge_q$. A move, $m \in \mathbb{M}_s$, at state s, is defined by the edges in the AND-OR graph that directly link $\wedge_s$ to its parents and children nodes in $\mathbb{S}_q$. For example, a move to ith child node of $\wedge_s$ means running the detector defined by the

terminal node $t_{\wedge_i}$, i.e., zooming-in and computing the α process of the child. Similarly, a move to lth parent node of $\wedge_s$ means zooming-out and running the detector $t_{\wedge_l}$.

We make the assumption that we have access to a *simulator*, which deterministically identifies next state s' (i.e., next AND node) after taking move m at state s. This simulator computes the log-posterior of K parse graphs in W, given by (4), from all α, β, and γ processes available until a given iteration. Since the simulator will always account for available detector responses in (4), the E^2 strategy should not repeat the moves which have already been taken. Since the moves are Markovian, we keep a record of detectors that have already been used $\mathbb{M}_{\text{used}}$.

A relatively small number of moves $|\mathbb{M}_s|$ at each state $s \in \mathbb{S}_q$ allows for a robust estimation of expected utilities of taking the moves, denoted as $\mathbb{Q}_q = [\mathbb{Q}(s, m; q)]$. $\mathbb{Q}_q$ is then used for guiding the scheduling of optimal moves in inference. One of the strengths of Q-learning is that it is able to compute $\mathbb{Q}_q$ without requiring a model of the environment. We specify a reward $\mathbb{R}_t(s, m; q)$ for taking move $m \in \mathbb{M}_s$ in state $s \in \mathbb{S}_q$, which results in the next state $s' \in \mathbb{S}_q$, and evaluate this reward for a given set of training parse graphs, $\{\text{pg}_t : t = 1, ..., T\}$. The reward is defined using the sigmoid function: $\mathbb{R}_t(s, m; q) = \left(1+\exp^{-\left(\log p(\text{pg}_t|\mathbb{M}_{\text{used}}) - \log p(\text{pg}_t|\mathbb{M}'_{\text{used}})\right)}\right)^{-1}$, where $\log p(\text{pg}_t|\mathbb{M}_{\text{used}})$ denotes the log-posterior distribution of tth training parse graph, given all detector responses in $\mathbb{M}_{\text{used}}$. Then, the Q-learning is run T times over all parse graphs $\{\text{pg}_t\}$, and $\mathbb{Q}_q$ is updated as, for $t = 1, ..., T$:

$$\mathbb{Q}(s, m; q) \leftarrow \mathbb{Q}(s, m; q) + \eta_s \left(\mathbb{R}_t(s, m; q) + \rho \max_{m'} \mathbb{Q}(s', m'; q) - \mathbb{Q}(s, m; q)\right), \quad (5)$$

where η_s is the learning rate, and ρ is the discounting factor. We estimate η_s as the inverse of the number of times state s has been visited, and set $\rho = 1$.

The E^2 strategy is summarized in Alg. 1. The initial state $s^{(0)} \in \mathbb{S}_q$ is assumed to be the query node in the AND-OR graph. The first move $m^{(0)} \in \mathbb{M}_s$ is defined as running the detector of the query. For selecting optimal moves in the following iterations, $\tau = 1, 2, ..., \mathcal{B}$, the E^2 strategy flips a biased coin, and, if the outcome is "heads", takes the best expected move $m^{(\tau+1)} = \arg\max_m \mathbb{Q}(s^{(\tau)}, m; q)$, otherwise takes any allowed move in state $s^{(\tau)}$. In both cases, the move is selected from the allowed set of previously unselected moves $\mathbb{M}_{s^{(\tau)}} \setminus \mathbb{M}_{\text{used}}$. We specify the probability of "heads" to be $\epsilon = 0.75$, and thus enable a mechanism for avoiding local optima. For the selected move $m^{(\tau+1)}$, our simulator evaluates the log-posterior of the parse graphs, $\{\text{pg}_k^{*(\tau+1)} : k = 1, ..., K\}$, over all available α, β, and γ processes, given by (4). If these K log-posteriors are above a certain threshold, δ, estimated in training, the algorithm can terminate before the allowed number of iterations $\mathcal{B}$. We do not study here the right values of δ and $\mathcal{B}$.

In our empirical evaluations, we have observed that the E^2 strategy produces a reasonable scheduling of α, β and γ. Fig. 3a, shows our evaluation of the E^2 strategy for the query Walking, under different time budgets, on the UCLA Courtyard dataset. Fig. 3b shows our sensitivity to ϵ values averaged over 10 different types of queries about group activities, primitive actions, and objects, for the allowed budget of 100 iteration steps, on the UCLA Courtyard dataset.

Algorithm 1: E^2 Strategy

Input: Query q; budget $\mathcal{B}$; Bernoulli "success" probability ϵ;
expected utilities $\mathbb{Q}_q = [\mathbb{Q}(s, m; q)]$; threshold δ

Output: All instances of q, inferred by the parse graphs, $\{\text{pg}_k^{*(\mathcal{B})} : k = 1, ..., K\}$

Initialize: $\tau = 0$; state $s^{(0)}$; move $m^{(0)}$; $\mathbb{M}_{\text{used}} = \emptyset$;
Compute $\{\text{pg}_k^{*(0)} : k = 1, ..., K\}$ given by (4) ;
while $(\tau < \mathcal{B})$ **or** $(\forall k,\ \log p(\text{pg}_k^{*(\tau)}|\mathbb{M}_{\text{used}}) \leq \delta)$ **do**
  Toss a biased coin with $p(\text{"heads"}) = \epsilon$;
  if *("heads")* **then**
    Select the best expected move $m^{(\tau+1)} = \arg\max_{m \in \mathbb{M}_{s^{(\tau)}} \setminus \mathbb{M}_{\text{used}}} \mathbb{Q}(s^{(\tau)}, m; q)$;
  else
    Select randomly a move $m^{(\tau+1)} \in \mathbb{M}_{s^{(\tau)}} \setminus \mathbb{M}_{\text{used}}$;
  end
  $\mathbb{M}_{\text{used}} = \mathbb{M}_{\text{used}} \cup \{m^{(\tau+1)}\}$;
  Evaluate $\{\text{pg}_k^{*(\tau+1)} : k = 1, ..., K\}$ for $\mathbb{M}_{\text{used}}$, given by (4);
  $\tau = \tau + 1$;
end

6 Learning the Model Parameters

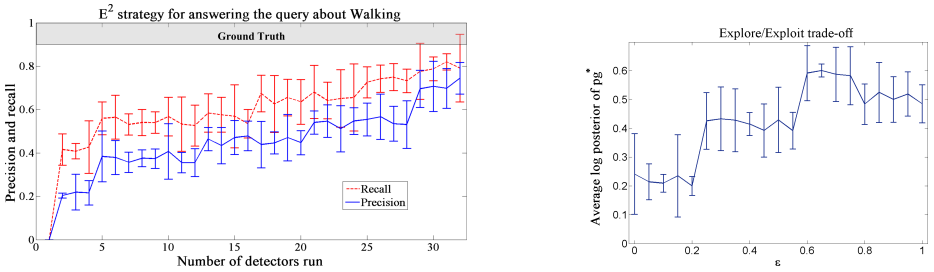
This section explains how to learn parameters of the pdf's appearing in (4).

We learn the distribution of the AND-OR graph structure, $p(\wedge^l|\vee^l)$, as the frequency of occurrence of pairs $(\wedge^l, \vee^l)$ in training parse graphs. The prior over the number of children nodes $p(N^l)$ is assumed exponential. Its ML parameter is learned on the numbers of corresponding children nodes of $\wedge^l$ in training parse graphs.

Learning α: For learning α^l, at a particular level l of the AND-OR graph, we use annotated sets of positive and negative training examples, $\{T^+_{\alpha^l}, T^-_{\alpha^l}\}$. $T^+_{\alpha^l}$ consists of labeled bounding boxes around corresponding group activities ($l = 1$), or primitive actions ($l = 2$), or objects ($l = 3$). Parameters of a classifier used for α^l detector (e.g., DPM of [14]) is learned on $\{T^+_{\alpha^l}, T^-_{\alpha^l}\}$ in a standard way for that classifier (e.g., using the cutting-plane algorithm for learning the structural latent SVM).

Learning γ: For learning γ^l of a primitive action (or object), we use training set T_{γ^l}. T_{γ^l} consists of pairs of descriptor vectors, $\{(X_{\wedge^l}, X_{\wedge^{l-}_i})\}$, extracted from bounding boxes annotated around instances of the primitive action (or object), and its contextual group activity (or primitive action) occurring in training videos. The descriptors capture the relative location, orientation, and scale of the corresponding pairs of training instances. T_{γ^l} is used for the ML learning of the mean and covariance, $(\mu_{\gamma^l}, \Sigma_{\gamma^l})$, of the Gaussian distribution $p(X_{\wedge^l}|X_{\wedge^{l-}})$.

Learning β: For learning β^l, we use two training sets: T'_{β^l}, and $T''_{\beta^{l+}}$. For a group activity (or primitive action), T'_{β^l} consists of pairs of descriptor vectors, $\{(X_{\wedge^l}, X_{\wedge^{l+}_i}) : i = 1, ..., N^l\}$, extracted from bounding boxes annotated around instances of the group activity (or primitive action), and its constituent primitive actions (or objects) occurring



(a) (b)

Fig. 3. Evaluation on the UCLA Courtyard dataset: (a) Precision and recall under different time budgets for the query Walking, averaged over all parse graphs. Our precision and recall increase as the number of detectors used reaches the maximum number 33. (b) Average log-posterior of ground-truth parse graphs of 10 different queries about group activities, primitive actions, and objects, for the budget of 100 iterations. The best results are achieved for $\epsilon \in [0.6 - 0.8]$.

in training videos. For a particular group activity (or primitive action), $T''_{\beta^{l+}}$ consists of all pairs of descriptor vectors, $\{(X_{\wedge_i^{l+}}, X_{\wedge_j^{l+}}) : i, j = 1, ..., N^l\}$, extracted from bounding boxes annotated around pairs of children of primitive actions (or objects) comprising the group activity (or primitive action). The descriptors capture the relative location, orientation, and scale of the corresponding pairs of training instances. T'_{β^l}, and $T''_{\beta^{l+}}$ are used for the ML learning of the means and covariances, $(\mu'_{\beta^l}, \Sigma'_{\beta^l})$ and $(\mu''_{\beta^l}, \Sigma''_{\beta^l})$, of the Gaussian distributions $p(X_{\wedge_i^{l+}} | X_{\wedge^l})$ and $p(X_{\wedge_i^{l+}}, X_{\wedge_j^{l+}})$.

7 Results

Existing benchmark datasets are not suitable for our evaluation. Major issues include: (1) unnatural, acted activities in constrained scenes; (2) limited spatial and temporal coverage; (3) limited resolution; (4) poor diversity of activity classes (particularly for multi-object events); (5) lack of concurrent events; and (6) lack of detailed annotations. For example, the VIRAT Ground dataset shows only single-actor activities (e.g., entering a building, parking a vehicle). The resolution of these videos (1280×720 or 1920×1080) is not sufficient to allow for digital zoom-in. Other surveillance datasets such as, VIRAT Aerial and CLIF, are not appropriate for our problem, since they are recorded from a high altitude where people are not visible. Other datasets (e.g, KTH, Weizmann, Youtube, Trecvid, PETS04, Olympic, CAVIAR, IXMAS, Hollywood, UCF, UT-Interaction or UIUC) are also not adequate, since they are primarily aimed at evaluating video classification. To address the needs of our evaluation, we have collected and annotated a new dataset, as explained below.

UCLA Courtyard Dataset [5]: The videos show two distinct scenes from a bird-eye viewpoint of a courtyard at the UCLA campus. The videos are suitable for our evaluation, since they show human activities at different semantic levels, and have a sufficiently high resolution to allow inference of fine details. The dataset consists of a 106-minute, 30 fps, 2560×1920-resolution video footage. We provide annotations in

terms of bounding boxes around group activities, primitive actions, and objects in each frame. A bounding box is annotated with the orientation and pose, where we use 4 orientation classes for groups, 8 orientations for people, and 7 poses for people. Each frame is also annotated with the ground plane, so as to allow finding a depth of each individual or group. The following group activities are annotated: 1. Walking-together, 2. Standing-in-line, 3. Discussing-in-group, 4. Sitting-together, 5. Waiting-in-group, and 6. Guided-tour. The following primitive actions are annotated: 1. Riding-skateboard, 2. Riding-bike, 3. Riding-scooter, 4. Driving-car, 5. Walking, 6. Talking, 7. Waiting, 8. Reading, 9. Eating, and 10. Sitting. Finally, the following objects are annotated: 1. Food, 2. Book, 3. Car, 4. Scooter, 5. Bike, 6. Food Bus, 7. Vending Machine, 8. Food Menu, 9. Bench, 10. Stairs, 11. Table, 12. Chair, 13. Bottle, 14. Phone, 15. Handbag, 16. Skateboard, and 17. Backpack. For each group activity or primitive action, the dataset contains 20 instances, and for each object the dataset contains 50 instances. We split the dataset 50-50% for training and testing.

We also use the Collective Activity Dataset [17] that consists of 75 short videos of crossing, waiting, queuing, walking, talking, running, and dancing. This dataset tests our performance on a collective behavior of individuals under realistic conditions, including background clutter, and transient occlusions. For training and testing, we use the standard split of $2/3$ and $1/3$ of the videos from each class. The dataset provides labels of every 10th frame, in terms of bounding boxes around people performing the activity, their pose, and activity class.

The Collective Activity Dataset mostly shows a single group activity per video. We increase its complexity by synthesizing a composite dataset. The composite videos represent a concatenation of multiple original videos randomly placed on a 2×2 grid, as shown in Fig. 4. The composite videos show four co-occurring group activities. We formed 20 such composite sequences of multiple co-occurring group activities, and used 50% for training and 50% for testing.

We evaluate our performance for varying time budgets: $\mathcal{B} = \{1, 15, \infty\}$. $\mathcal{B} = 1$ means that we are allowed to run only the detector directly appropriate for the query (e.g., the detector of Riding-bike). This is our baseline. $\mathcal{B} = \infty$ means that we run the E^2 strategy as long as all detectors and their integration via the α, β, and γ processes are not executed. Finally, $1 < \mathcal{B} < \infty$ means that the E^2 strategy is run for $\mathcal{B}$ iterations.

We evaluate: i) Classification accuracy and ii) Recall and precision of activity detection. For detection evaluation, we compute a ratio, ρ, of the intersection and union of detected and ground-truth time intervals of activity occurrences. True positive (TP) is declared if the activity is correctly recognized, and $\rho > 0.5$, otherwise we declare false positive (FP). Note that this also evaluates localization of the start and end frames of activity occurrences.

Table 1 shows our precision, false positive rates, and running times, under varying time budgets, on the UCLA Courtyard dataset. As the budget increases, we observe better performance. The E^2 strategy gives slightly worse results in a significantly less amount of time, than the full inference with unlimited budget. Thus, the E^2 strategy improves the accuracy-complexity trade-off.

Table 2 compares our classification accuracy and running times with those of the state of the art [9, 11, 17] on the Collective Activity Dataset. For this comparison, we

	Query about group activities						
E^2 strategy	Standing-in-line	Guided-tour	Discussing	Sitting	Walking	Waiting	Time
$\mathcal{B} = 1$, Precision	62.2%	63.7%	68.1%	65.3%	69.4%	61.2%	5s
$\mathcal{B} = 1$, FP	7.2%	2.3%	9.8%	12.6%	8.1%	10.4%	5s
$\mathcal{B} = 15$, Precision	65.4%	66.1%	69.0%	68.7%	70.3%	66.5%	75s
$\mathcal{B} = 15$ FP	10.1%	4.7%	11.1%	11.1%	8.7%	10.9%	75s
$\mathcal{B} =$, Precision	68.0%	70.2%	75.1%	71.4%	78.6%	72.6%	230s
$\mathcal{B} = \infty$, FP	13.6%	10.3%	17.1%	13.7%	10.1%	12.2%	230s

	Query about primitive actions										
E^2 strategy	Walk	Wait	Talk	Drive Car	Ride S-board	Ride Scooter	Ride Bike	Read	Eat	Sit	Time
$\mathcal{B} = 1$, Precision	63.3%	61.2%	58.4%	65.8%	63.5%	60.1%	56.8%	55.3%	60.9%	54.3%	10s
$\mathcal{B} = 1$, FP	12.1%	16.2%	11.4%	3.4%	10.2%	11.6%	6.2%	8.2%	2.2%	5.3%	10s
$\mathcal{B} = 15$, Precision	67.6%	63.4%	62.3%	67.2%	67.1%	65.9%	59.3%	61.2%	66.3%	59.2%	150s
$\mathcal{B} = 15$, FP	14.2%	17.1%	15.1%	7.1%	13.8%	13.2%	9.3%	10.3%	4.3%	7.1%	150s
$\mathcal{B} = \infty$, Precision	69.1%	67.7%	69.6%	70.2%	71.3%	68.4%	61.4%	67.3%	71.3%	64.2%	330s
$\mathcal{B} = \infty$, FP	18.7%	20.2%	17.9%	9.7%	17.1%	16.3%	12.3%	12.1%	7.7%	9.0%	330s

Table 1. Average precision, and false positive rates on the UCLA Courtyard Dataset for primitive actions and group activities. The larger the time budget, the better precision.

allow infinite budget in inference, and do not account for objects, since this information is not available to the competing approaches. As can be seen, we our performance is superior in reasonable running times. Figures 4 and 5 illustrate our qualitative results.

8 Conclusion

We have formulated and addressed a new problem, that of multiscale activity recognition, where the main challenge is to make inference cost-sensitive and scalable. Our approach models group activities, individual actions, and participating objects with the AND-OR graph, and exploits its hierarchical structure to formulate a new inference algorithm. The inference is iterative, where the direct application of activity detectors, bottom-up and top-down computational processes are optimally scheduled using an explore-exploit (E^2) strategy. For evaluation, we have compiled a new dataset of 106-minute, 30 fps, 2560×1920-resolution video footage. The dataset alleviates the shortcomings of existing benchmarks, since its videos show unstaged human activities of different semantic scales co-occurring in a vast scene, and have a sufficiently high resolution to allow for digital zoom-in (or zoom-out) for examining fine details (or coarser scales), as needed for recognition. The E^2 strategy improves the accuracy-complexity trade-off of full inference of the AND-OR graph. We have also reported competitive results on the benchmark Collective activities dataset.

Acknowledgement

This research has been sponsored in part by grants DARPA MSEE FA 8650-11-1-7149 and ONR MURI N00014-10-1-0933.

Class	**Our**	[11]	[18]	[9]	[17]
Walk	74.7%	38.8%	72.2%	68%	57.9%
Cross	77.2%	76.4%	69.9%	65%	55.4%
Queue	95.4%	78.7%	96.8%	96%	63.3%
Wait	78.3%	76.7%	74.1%	68%	64.6%
Talk	98.4%	85.7%	99.8%	99%	83.6%
Run	89.4%	N/A	87.6%	N/A	N/A
Dance	72.3%	N/A	70.2%	N/A	N/A
Avg	83.6%	70.9%	81.5%	79.1%	65.9%
Time	165s	N/A	55s	N/A	N/A

Table 2. Average classification accuracy, and running times on the Collective Activity Dataset [17]. We use $\mathcal{B} = \infty$.

Class	**Our**	**Our** FP-Rate	[18]	[18] FP-Rate
Walk	65.3%	8.2%	58.1%	12.2%
Cross	69.6%	8.7%	61.5%	15.5%
Queue	76.2%	5.2%	65.5%	8.7%
Wait	68.3%	7.7%	59.2%	8.2%
Talk	82.1%	6.2%	67.5%	7.1%
Run	80.4%	8.8%	72.1%	10.2%
Dance	63.1%	10.2%	55.3%	12.9%
Avg	72.1%	6.7%	62.7%	10.6%

Table 3. Average precision, and false positive rates on the Composite Collective Activity dataset. We use $\mathcal{B} = \infty$.

Fig. 4. Our results on detecting group activities of the Composite Collective Activity dataset, for $\mathcal{B} = \infty$. The figure shows a single frame (not 4 frames) from the Composite dataset. A total of 7 co-occurring activity instances are detected. The detections are color coded. Top left: we detect the co-occurring Walking and Waiting. Top right: we detect the co-occurring Queuing, Talking, and Waiting. Bottom row: we detect Crossing (left), and Talking (right).

References

1. Zhu, S.C., Mumford, D.: A stochastic grammar of images. Found. Trends. Comput. Graph. Vis. **2** (2006) 259–362
2. Gupta, A., Srinivasan, P., Shi, J., Davis, L.: Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In: CVPR. (2009)
3. Si, Z., Pei, M., Yao, B., Zhu, S.C.: Unsupervised learning of event AND-OR grammar and semantics from video. In: ICCV. (2011)
4. Wu, T., Zhu, S.C.: A numerical study of the bottom-up and top-down inference processes in and-or graphs. IJCV **93** (2011) 226–252
5. : UCLA Courtyard Dataset. `http://vcla.stat.ucla.edu/Projects/Multiscale_Activity_Recognition/` (2012)
6. Wu, X., Xu, D., Duan, L., Luo, J.: Action recognition using context and appearance distribution features. In: CVPR. (2011)
7. Yao, B., Zhu, S.C.: Learning deformable action templates from cluttered videos. In: ICCV. (2009)
8. Amer, M., Todorovic, S.: Sum-product networks for modeling activities with stochastic structure. In: CVPR. (2012)
9. Lan, T., Wang, Y., Yang, W., Mori, G.: Beyond actions: Discriminative models for contextual group activities. In: NIPS. (2010)
10. Ryoo, M.S., Aggarwal, J.K.: Stochastic representation and recognition of high-level group activities. IJCV **93** (2011) 183–200

Fig. 5. Our results on an example video from the UCLA Courtyard dataset, under unlimited time budget. Detections are color coded, where the codes are given below each frame. Top left: results of the α's of group activities using the input poses and person detections. Top right: results of the α's of 10 objects. Bottom left: results of the α's of primitive actions. Bottom right: results for group activities and primitive actions of all parse graphs. (Best viewed zoomed-in, in color.)

11. Choi, W., Shahid, K., Savarese, S.: Learning context for collective activity recognition. In: CVPR. (2011)
12. Sivic, J., Zisserman, A.: Efficient visual search for objects in videos. Proceedings of the IEEE **96** (2008) 548–566
13. Yao, A., Gall, J., Gool, L.J.V.: A hough transform-based voting framework for action recognition. In: CVPR. (2010)
14. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1627–1645
15. Matikainen, P., Hebert, M., Sukthankar, R.: Representing pairwise spatial and temporal relations for action recognition. In: ECCV. (2010)
16. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008)
17. Choi, W., Shahid, K., Savarese, S.: What are they doing? : Collective activity classification using spatio-temporal relationship among people. In: ICCV. (2009)
18. Amer, M., Todorovic, S.: A Chains model for localizing group activities in videos. In: ICCV. (2011)