Context-Aware Activity Modeling using Hierarchical Conditional Random Fields

Yingying Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury

Abstract—In this paper, rather than modeling activities in videos individually, we jointly model and recognize related activities in a scene using both motion and context features. This is motivated from the observations that activities related in space and time rarely occur independently and can serve as the context for each other. We propose a two-layer conditional random field model, that represents the action segments and activities in a hierarchical manner. The model allows the integration of both motion and various context features at different levels and automatically learns the statistics that capture the patterns of the features. With weakly labeled training data, the learning problem is formulated as a max-margin problem and is solved by an iterative algorithm. Rather than generating activity labels for individual activities, our model simultaneously predicts an optimum structural label for the related activities in the scene. We show promising results on the UCLA Office Dataset and VIRAT Ground Dataset that demonstrate the benefit of hierarchical modeling of related activities using both motion and context features.

Index Terms—Activity localization and recognition, Context-aware activity model, Hierarchical Conditional Random Field.

1 Introduction

It has been demonstrated in [28] that context is significant in human visual systems. As there is no formal definition of context in computer vision, we consider all the detected objects and motion regions as providing contextual information about each other. Activities in natural scenes rarely happen independently. The spatial layout of activities and their sequential patterns provide useful cues for their understanding. Consider the activities that happen in the same spatio-temporal region in Fig. 1: the existence of the nearby car gives information about what the person (bounded by red circle) is doing, and the relative position of the person of interest and the car says that activities (b) and (c) are very different from activity (a). Moreover, just focusing on the person, it may be hard to tell what the person is doing in (b) and (c) - “opening vehicle trunk” or “closing vehicle trunk”. If we knew that these activities occurred around the same vehicle along time, it would be immediately clear that in (b) the person is opening the vehicle trunk and in (c) the person is closing the vehicle trunk. This example shows the importance of spatial and temporal relationships for activity recognition.

1.1 Overview of the Framework

Many existing works on activity recognition assume that, the temporal locations of the activities are known [1], [27].

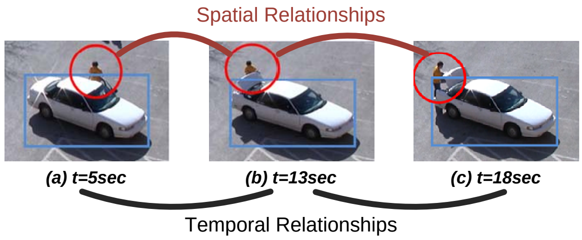


Fig. 1. An example that demonstrates the importance of context in activity recognition. Motion region surrounding the person of interest is located by red circle, interacting vehicle is located by blue bounding box.

In practice, activity-based analysis of videos should involve reasoning about motion regions, objects involved in these motion regions, and spatio-temporal relationships between the motion regions. We focus on the problem of detecting activities of interest in *continuous* videos without prior information about the locations of the activities. The main challenge is to develop a representation of the continuous video that respects the spatio-temporal relationships of the activities. To achieve this goal, we build upon existing well-known feature descriptors and spatio-temporal context representations that, when combined together, provide a powerful framework to model activities in continuous videos.

An activity can be considered as a union of action segments or actions that are neighbors to each other closely in space and time. We provide an integrated framework that conducts multiple stages of video analysis, starting with motion localization. The detected motion regions are divided into action segments, which are considered as the elements of activities, using a motion segmentation algorithm based on the nonlinear dynamic model (NDM) in [5]. The goal then is to generate smoothed activity labels, which

- **This work was partially supported under ONR grant N00014-12-1-1026 and NSF grant IIS-1316934.*

- *Y. Zhu is with the Department of Electrical and Computer Engineering, University of California, Riverside. E-mail: yzhu010@ucr.edu.*
- *N. M. Nayak is with the Department of Computer Science and Engineering, University of California, Riverside. E-mail: nandita.nayak@email.ucr.edu.*
- *A. K. Roy-Chowdhury is with the Department of Electrical and Computer Engineering, University of California, Riverside. E-mail: amitrc@ee.ucr.edu.*

are optimum in a global sense, for the action segments; and thus obtaining semantically meaningful activity regions and corresponding activity labels.

Towards this goal, we perform an initial labeling to group adjacent action segments into semantically meaningful activities using a baseline activity detector. Any existing activity detection method, such as sliding window bag-of-words (BOW) with a support vector machine (SVM) [25] can be used in this step. We call the labeled groups of action segments as the candidate activities. Candidate activities that are related to each other in space and time are grouped together into activity sets. For each set, the underlying activities are jointly modeled and recognized with the proposed two-layer Conditional Random Field model, which models the hierarchical relationship between the action segments and activities. We refer to this proposed two-layer Hierarchical-CRF as Hierarchical-CRF in short for simplicity of expression. First, the action layer is modeled as a linear-chain CRF model with the activity labels with the action segments as the random variables. Latent activity variables, which represent the detected activities, are then introduced in the hidden activity layer. Doing so, action-activity consistency and intra-activity potentials, as the higher-order smoothness potentials, can be introduced into the model to smooth the preliminary activity labels in the action layer. Finally, the activity layer variables, whose underlying activities are within the neighborhoods of each other in space and time, are connected to utilize the spatial and temporal relationships between activities. The resulting model is the action-based two-layer Hierarchical-CRF model.

Potentials in and between the action and activity layers are developed to represent the motion and context patterns of individual variables and groups of them in both action and activity levels, as well as action-activity consistency patterns between variables in the two layers. The action-activity potentials upon sets of action nodes and their corresponding activity nodes are introduced between action and activity layers. Such potentials, as smoothness potentials, are used to enforce label consistency of action segments within activity regions while allowing for label inconsistency for certain circumstances. This allows the rectification of the preliminary activity labels of action segments during the inference of the Hierarchical-CRF model according to the motion and context patterns in and between actions and activities.

Fig. 2 shows the framework of our approach. Given a video, we detect the motion regions using background subtraction. Then, the segmentation algorithm aims to divide a continuous motion region into action segments, whose motion pattern is consistent and is different from its adjacent segments. These action segments, as the nodes in the action layer, are modeled as a linear-chain CRF and the proposed Hierarchical-CRF model is built accordingly as described above.

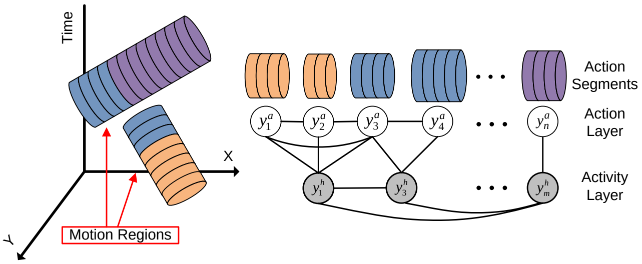


Fig. 2. The left graph shows the video representation of an activity set with n motion segments and m candidate activities. The right graph shows the graphical representation of our Hierarchical-CRF model. The white nodes are the action variables and the gray nodes in the graph are the hidden activity variables. Note that observations associated with the model variables are not shown for clear representation.

The model parameters are learned automatically from weakly-labeled training data with the location and labels of activities of interest. Image-level features are detected and organized to form the context for activities. Common sense domain knowledge about the activities of interest is used to guide the formulation of these context features within activities from the weakly-labeled training data. We utilize a structural model in a max-margin framework, iteratively inferring the hidden activity variables and learning the parameters of different layers. For the testing, the action segments, which are merged together and assigned with activity labels by the preliminary activity detection method, are relabeled through inference on the learned Hierarchical-CRF model.

1.2 Main Contributions

The main contribution of this work is three-fold.

(i) We combine low-level motion segmentation with high-level activity model under one framework. With the detected individual action segments as the elements of activities, we design a Hierarchical-CRF model that jointly models the related activities in the scene.

(ii) We propose a weakly supervised approach that utilizes context within and between actions and activities that provide helpful cues for activity recognition. The proposed model integrates motion and various context features within and between actions and activities into a unified model. The proposed model can localize and label activities in continuous videos simultaneously, in the presence of multiple actors in the scene interacting with each other or acting independently.

(iii) With a task-oriented discriminative approach, the model learning problem is formulated as a max-margin problem and is solved by an Expectation Maximization approach.

2 Related Work

Many existing works exploring context focus on interactions among features, objects and actions [1], [3], [14], [34], [39], environmental conditions such as spatial locations of certain activities in the scene [23], and temporal relationships between activities [24], [35]. Spatio-temporal constraints across activities in a wide-area scene are rarely considered.

Motion segmentation and action recognition are done simultaneously in [26]. The proposed algorithm models the temporal order of actions while ignoring the spatial relationships between actions. The work in [35] models a complex activity by a variable-duration hidden Markov model on equal-length temporal segments. It decomposes a complex activity into sequential actions, which are the context of each other. However, it considers only the temporal relationships, while ignoring the spatial relationships between actions. AND-OR graph [2], [13], [32] is a powerful tool for activity representation. It has been used for multi-scale analysis of human activities in [2], α, β, γ procedures were defined for a bottom-up cost sensitive inference of low-level action detection. However, the learning and inference processes of AND-OR graphs become more complex as the graph grows large and more and more activities are learned. In [20], [21], a structural model is proposed to learn both feature-level and action-level interactions of group members. This method labels each image with a group activity label. How to smooth the labeling results along time is a problem and is not addressed in the paper. Also, these methods aim to recognize group activities and are not suitable in our scenario where activities cannot be considered as the parts of larger activities.

In [4], complex activities are represented as spatiotemporal graphs representing multi-scale video segments and their hierarchical relationships. Existing higher-order models [16], [17], [19], [42] propose the use of higher order potentials that encourage the smoothness of variables within cliques of the graph. Higher-order graphical models have been frequently used in image segmentation, object recognition, etc. However, few works exist in the field of activity recognition. We propose a novel method that explicitly models the action and activity level motion and context patterns with a Hierarchical-CRF model and use them in the inference stage for recognition.

The problem of simultaneous tracking and activity recognition was addressed in [6], [15]. In these works, tracking and action/activity recognition are expected to benefit each other through an iterative process that maximizes a decomposable potential function which consists of tracking potentials and action/activity potentials. However, only collective activities are considered in [6], [15], in which the individual persons of interest have a common goal in terms of activity. This work address the general problem of activity recognition, when individual persons in the scene may conduct heterogeneous activities.

The inference method on a structural model proposed in [20], [21] searches through the graphical structure, in order to find the one that maximizes the potential function. Though this inference method is computationally less intensive than exhaustive search, it is still time consuming. As an alternative, greedy search has been used for inference in object recognition [8].

This paper has major differences with our previous work in [45]. In [45], we proposed a structural SVM to explicitly model the durations, motion, intra-activity context and the spatio-temporal relationships between the activities. In this work, we develop a hierarchical model which represents the related activities in a hidden activity layer, which interacts with a lower-level action layer. Representing activities as hidden activity variables simplifies the inference problem, by associating each hidden activity with a small set of neighboring action segments, and enables efficient iterative learning and inference algorithms. Furthermore, the modeling of more aspects of the activities of interest adds additional feature functions that measure both action and activity variables. Since more information about the activities to be recognized is modeled, the recognition accuracy is improved as demonstrated by the experiments.

3 Model Formulation for Context-Aware Activity Representation

In this section, we describe how the higher-order conditional random field (CRF) modeling of activities that integrates activity durations, motion features and various context features within and across activities is built upon automatically detected action segments to jointly model related activities in space and time.

3.1 Video Preprocessing

Assuming there are $M+1$ classes of activities at the scene, including a background class with label 0 and M classes of interest with labels $1,...,M$ (the background class can be omitted if all the activity classes in the scene are known). Our goal is to locate and label each activity of interest in videos. Given a continuous video, background substraction [46] is used to locate the moving objects. Moving persons are identified, and local trajectories of moving persons are generated (any existing tracking methods like [33] can be used). Spatio-temporal Interest Point (STIP) features [22] are generated only for these motion regions. Thus, STIPs generated by noise, such as slight tree shaking, camera jitter and motion of shadows, are avoided. Each motion region is segmented into action segments using the motion segmentation based on the method in [5] with STIP histograms as the model observation. The detailed motion segmentation algorithm is described in Section 5.3.1.

3.2 Hierarchical-CRF Models for Activities

The problem of activity recognition in continuous videos requires two main tasks: to detect motion regions and to label these detected motion regions. The detection and labeling problems can be solved simultaneously as proposed in [26] or separately as proposed in [44], [45]. For the latter, candidate action or activity regions are usually detected before the labeling task. The problem of activity recognition is then converted to a problem of labeling, that is, to assign each candidate region with an optimum activity label.

CRF is a discriminative model often used usually used for labeling problems of image and image objects. Essentially, CRF can be considered as a special version of Markov Random Field (MRF) where the variable potentials are conditioned on the observed data. Let $\mathbf{x}$ be the model observations and $\mathbf{y}$ be the label variables. The posterior distribution $p(\mathbf{y}|\mathbf{x},\omega)$ of the label variables over the CRF is a *Gibbs* distribution and is usually represented as

$$p(\mathbf{y}|\mathbf{x},\omega) = \frac{1}{Z(\mathbf{x},\omega)} \prod_{c\in C} exp({\omega_c}^T \varphi_c(\mathbf{x},\mathbf{y}_c)), \tag{1}$$

where ω_c is a model weight vector, which needs to be learned from training data. $Z(\mathbf{x},\omega)$ is a normalizing constant called the partition function. $\varphi_c(\mathbf{x},\mathbf{y}_c)$ is a feature vector derived from the observation $\mathbf{x}$ and the label vector, $\mathbf{y}_c$, in the clique c.

The potential function of the CRF model given the observations $\mathbf{x}$ and model weight vector ω is defined as

$$\psi(\mathbf{y}|\mathbf{x},\omega) = \sum_c {\omega_c}^T \varphi_c(\mathbf{x},\mathbf{y}_c). \tag{2}$$

For the development of the Hierarchical-CRF model, the action layer is first modeled as a linear-chain CRF. Activity layer variables which are associated with detected activities are then introduced for the smoothing of the action-layer variables. Finally, activity-layer variables are connected to represent the spatial and temporal relationships between activities. The evolution of the proposed two-layer Hierarchical-CRF model from the one-layer CRF model is shown in Fig. 3. Details on the development of these models will be described in the following sub-sections. The various feature vectors used for the calculation of the potentials are described in Section 3.3.

3.2.1 *Action-based Linear-chain CRF*

We first describe the linear-chain CRF model in Fig. 3(a). We first define the following items: *intra-action potential* $\psi_\nu(y_i^a|\mathbf{x},\omega)$, which measures the compatibility of the observed feature of i and its label y_i^a; *inter-action potential* $\psi_\varepsilon(y_i^a,y_j^a|\mathbf{x},\omega)$, which measures the consistency between two connected action segments i and j. Let $\mathscr{V}^a$ be the set of vertices, each representing an action segment as the element in the action layer and $\mathscr{E}^a$ denotes the set of connected action pairs. The potential function of the action-layer linear-chain CRF is

$$\begin{aligned}\psi(\mathbf{y}^a|\mathbf{x},\omega) &= \sum_{i\in\mathscr{V}^a} \psi_\nu(y_i^a|\mathbf{x},\omega) + \sum_{ij\in\mathscr{E}^a} \psi_\varepsilon(y_i^a,y_j^a|\mathbf{x},\omega) \qquad (3)\\ &= \sum_{i\in\mathscr{V}^a} {\omega_{\nu,y_i^a}^a}^T \varphi_\nu(\mathbf{x}_i^a,y_i^a) + \sum_{ij\in\mathscr{E}^a} {\omega_{\varepsilon,y_i^a,y_j^a}^a}^T \varphi_\varepsilon(\mathbf{x}_i^a,\mathbf{x}_j^a,y_i^a,y_j^a),\end{aligned}$$

where $\varphi_\nu(\mathbf{x}_i^a,y_i^a)$ is the *intra-action feature vector* that describes action segment i. $\omega_{\nu,y_i^a}^a$ is the weight vector of the intra-action features for class y_i^a. $\varphi_\varepsilon(\mathbf{x}_i^a,\mathbf{x}_j^a,y_i^a,y_j^a)$ is the *inter-action feature*, which is derived from the labels y_i^a, y_j^a and intra-action feature vectors $\mathbf{x}_i^a$ and $\mathbf{x}_j^a$. $\omega_{\varepsilon,y_i^a,y_j^a}^a$ is the weight vector of the inter-action features for class pair y_i^a,y_j^a.

3.2.2 *Incorporating Higher Order Potentials*

According to experimental observations, action segments in a candidate activity region, which are generated by activity detection methods [44], tend to have the same activity labels. However, consistent labeling is not guaranteed due to inaccurate detections. Let an action clique c^a denote the union of action segments in a candidate activity c. The linear-chain CRF can be converted to a higher-order CRF by adding a latent activity variable y_c^h, representing the label of c, for each action clique c^a. All action variables associated with the same activity variable are connected. Then, the associated higher-order potential $\psi_c(\mathbf{y}_c^a|\mathbf{x},\omega)$ is introduced to encourage action segments in the clique c^a to take the same label, while still allowing some of them to have different labels without additional penalty. The resulting CRF model is shown in 3 (b). The potential function ψ for the higher-order CRF model is represented as

$$\begin{aligned}\psi(\mathbf{y}^a,y_c^h|\mathbf{x},\omega) = &\sum_{i\in\mathscr{V}^a} {\omega_{\nu,y_i^a}^a}^T \varphi_\nu(\mathbf{x}_i^a,y_i^a) \qquad (4)\\ &+ \sum_{ij\in\{\mathscr{E}'^a\}} {\omega_{\varepsilon,y_i^a,y_j^a}^a}^T \varphi_\varepsilon(\mathbf{x}_i^a,\mathbf{x}_j^a,y_i^a,y_j^a) + \sum_{c\in C^{ah}} \psi_c(\mathbf{y}_c^a|\mathbf{x},\omega),\end{aligned}$$

where $\mathscr{E}'^a$ denotes the set of connected action pairs in the new model. C^{ah} is the set of action-activity cliques and each action-activity clique c in C^{ah} corresponds to an action clique c^a in the action layer and its associated activity c in the activity layer. Let $\mathscr{L} = 0,1,\cdots,M$ be the activity label set in the action layer, from which the action variables may take values. The activity variable y_c^h takes values from an extended label set $\mathscr{L}_h = \mathscr{L} \cup l_f$, where $\mathscr{L}$ is the set of variable values in the action layer. When an activity variable takes value l_f, it allows its child variables to take different labels in $\mathscr{L}$, without additional penalty upon label inconsistency.

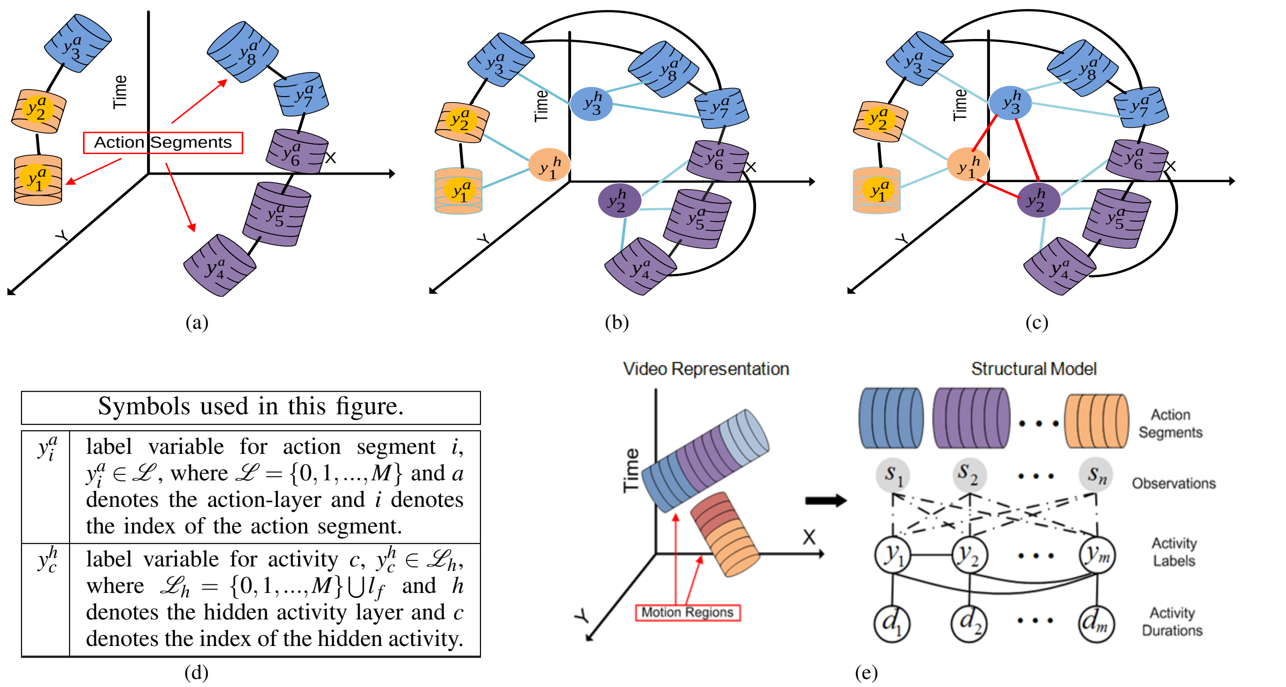
We define $\varphi_{c,l}(\mathbf{y}_c^a,y_c^h)$ as the *action-activity consistency feature* of activity c, and $\omega_{c,l,y_c^h}^{ah}$ to be the weight vector of the action-activity consistency feature for class y_c^h. Define $\varphi_{c,f}(\mathbf{x}_c^a,y_c^h)$ as the *intra-activity feature* for activity c, and $\omega_{c,f,y_c^h}^{ah}$ to be the weight vector of intra-activity feature for class y_c^h. The corresponding action-activity higher-order potential can be defined as

$$\begin{aligned}\psi(\mathbf{y}_c^a|\mathbf{x},\omega) &= \max_{y_c^h} {\omega_{c,y_c^h}^{ah}}^T \varphi_c(\mathbf{x}_c^a,\mathbf{x}_c^h,\mathbf{y}_c^a,y_c^h) \qquad (5)\\ &= \max_{y_c^h}[{\omega_{c,l,\mathbf{y}_c^a,y_c^h}^{ah}}^T \varphi_{c,l}(\mathbf{y}_c^a,y_c^h) + {\omega_{c,f,y_c^h}^{ah}}^T \varphi_{c,f}(\mathbf{x}_c^a,y_c^h)],\end{aligned}$$

where ${\omega_{c,l,y_c^h}^{ah}}^T \varphi_{c,l}(\mathbf{y}_c^a,y_c^h)$ measures the labeling consistency within the activity c. Intuitively, the higher-order potentials are constructed such that a latent variable tends to take a label from $\mathscr{L}$ if majority of its child nodes take the same value, and take the label l_f if its child nodes take diversified values. ${\omega_{c,f,y_c^h}^{ah}}^T \varphi_{c,f}(\mathbf{x}_c^a,y_c^h)$ is the *intra-activity potential* that measures the compatibility between the activity label of clique c and its activity features.

3.2.3 *Incorporating Inter-Activity Potentials*

As stated before, it would be helpful to model the spatial and temporal relationships between activities. For this reason, we connect activity nodes in the higher-order CRF model. The resulting CRF is shown in Fig. 3(c). We define $\varphi_{sc}(\mathbf{x}_s^h,\mathbf{x}_d^h,y_s^h,y_d^h)$ as the *inter-activity spatial feature* that



(a) (b) (c)

Symbols used in this figure.	
y^a_i	label variable for action segment i, $y^a_i \in \mathscr{L}$, where $\mathscr{L} = \{0,1,...,M\}$ and a denotes the action-layer and i denotes the index of the action segment.
y^h_c	label variable for activity c, $y^h_c \in \mathscr{L}_h$, where $\mathscr{L}_h = \{0,1,...,M\} \bigcup l_f$ and h denotes the hidden activity layer and c denotes the index of the hidden activity.

(d) (e)

Fig. 3. Illustration of CRF models for activity recognition. (a): Action-based Linear-Chain CRF; (b): Action-based higher-order CRF model (with latent activity variables); (c): Action-based two-layer Hierarchical-CRF. Note that all the observations for the random variables are omitted for compactness; (d): symbols in sub-figures (a, b, c); (e): graph representation of the model in [45] for comparison. One action segment denotes a random variable in the action layer, whose value is the activity label for the action segment. A colored circle denotes a random variable in the activity layer, whose value is the label for its connected clique. As shown in (a), in the action layer, action segments that belong to the same trajectory are modeled as a linear-chain CRF. Then, hidden activity-level variables with action-activity edges (in light blue) are added for each action clique to form higher-order CRF as shown in (b). An activity and its associated action nodes have a same color. Finally, pair-wise activity edges (in red) are added to form the proposed two-layer Hierarchical-CRF mdoel.

encodes the spatial relationship between activities s and d, and $\omega^h_{sc,y^h_s,y^h_d}$ to be the weight vector of inter-activity spatial feature for class pair (y^h_s, y^h_d). Define $\varphi_{tc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)$ as the *inter-activity temporal feature* that encodes the temporal relationship between activities s and d, and $\omega^h_{tc,y^h_s,y^h_d}$ to be the weight vector of inter-activity temporal feature for class pair (y^h_s, y^h_d).

The pairwise activity potential between clique s and d is defined as

$$\psi(\mathbf{y}^h|\mathbf{x},\omega) = \sum_{sd\in\mathscr{E}^h} [{\omega^h_{sc,y^h_s,y^h_d}}^T \varphi_{sc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d) + {\omega^h_{tc,y^h_s,y^h_d}}^T \varphi_{tc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)], \tag{6}$$

where ${\omega^h_{sc,y^h_s,y^h_d}}^T \varphi_{sc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)$ is the pairwise spatial potential between activities s and d that measures the compatibility between the candidate labels of s and d and their spatial relationship. ${\omega^h_{tc,y^h_s,y^h_d}}^T \varphi_{tc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)$ is the pairwise temporal potential between activities s and d that measures the compatibility between the candidate labels of s and d and their temporal relationship.

3.3 Feature Descriptors

We now define the concepts we use for the feature development. An activity is a 3D region consisting of one or multiple consecutive action segments. An agent is the underlying moving person(s) or a trajectory. Motion region at frame n is the region surrounding the moving objects of interest in the n^{th} frame of the activity. Activity region is the smallest rectangle region that encapsulates the motion regions over all frames of the activity. In general, same type of features for different class or class pair can be different. There are mainly three kinds of features in our model: action-layer features, action-activity features and activity-layer features, which can be further divided into five types of features. We now describe how to encode motion and context information into feature descriptors.

Intra-action Feature: $\varphi_v(\mathbf{x}^a_i, y^a_i)$ encodes the motion information of the action segment i that is extracted from low-level motion features such as STIP features. Since in the action layer, we obtain action segments by utilizing their discriminative motion patterns, we use only motion features for the development of action-layer features. STIP histograms are generated for each action segment using bag-of-word method [25]. We train a kernel multi-SVM upon action segments to generate the normalized confidence scores, $s_{i,j}$, of classifying the action segment i as activity class j, where $j \in \{0,1,...,M\}$, such that $\sum_{j=0}^{M} s_{i,j} = 1$. In general, any kind of classifier and low-level motion features can be used here. Given an action segment i,

$\varphi_v(\mathbf{x}_i^a, y_i^a) = [s_{i,0} \cdots s_{i,M}]^T$ is developed as the intra-action feature descriptor of action segment i.

Inter-action Feature: $\varphi_\varepsilon(\mathbf{x}_i^a, \mathbf{x}_j^a, y_i^a, y_j^a)$ encodes the probabilities of coexistence of action segments i and j according to their features and activity labels. $\varphi_\varepsilon(\mathbf{x}_i^a, \mathbf{x}_j^a, y_i^a, y_j^a) = \mathbf{I}(y_i^a)\mathbf{I}(y_j^a)$, where $\mathbf{I}(y_k^a)$ is the Dirac measure that equals 1 if the true label of segment k is y_k^a and equals to 0 other wise, for $k = i, j$.

Action-Activity Consistency Feature: $\varphi_{c,l}(\mathbf{y}_c^a, y_c^h)$ encodes the labeling information within clique c as

$$\varphi_{c,l}(\mathbf{y}_c^a, y_c^h) = \begin{cases} 1 & y_c^h = l_f \\ \frac{\sum_{i \in c} I(y_i^a = y_c^h)}{N_c} & y_c^h \in \mathscr{L} \end{cases}.$$

where $I(\cdot)$ is the Dirac measure and N_c is the number of action segments in clique c.

Intra-activity Feature: $\varphi_{c,f}(\mathbf{x}_c^a, x_c^h, \mathbf{y}_c^a, y_c^h)$ encodes the intra-activity motion and context information of activity c. To capture the motion pattern of an activity, we use the intra-action features of action segments which belong to the activity. Given an activity, $[\max_{i \in \aleph} s_{i,0}, \ldots, \max_{i \in \aleph} s_{i,M}]$ is developed as the intra-activity motion feature descriptor, where $\aleph$ is a list of action segments in activity c.

Intra-activity context feature captures the context information about the agents and relationships between the agents, as well as the the interacting objects (e.g. the object classes, interactions between agents and their surroundings). We define a set, G, of attributes that describes such context for activities of interest, using common-sense knowledge about the activities of interest (how to identify such attributes automatically is another research topic that we do not address in this paper). For a given activity, whether the defined attributes are true or not are determined from image-level detection results. The resulting feature descriptor is a normalized feature histogram. The attributes used and the development of intra-activity context features are different for different tasks (please refer to Section 5.3.1 for the details).

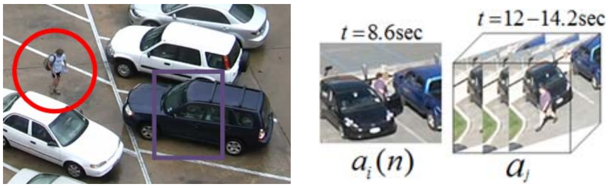
Finally, the weighted motion and context features are used as the input to a multi-SVM and the output confidence scores are used to develop the intra-activity feature as $\varphi_{c,f}(\mathbf{x}_c^a, y_c^h) = [s_{c,0}, \ldots, s_{c,M}]^T$.

Inter-activity Spatial and Temporal Features $\varphi_{sc}(\mathbf{x}_s^h, \mathbf{x}_d^h, y_s^h, y_d^h)$ and $\varphi_{tc}(\mathbf{x}_s^h, \mathbf{x}_d^h, y_s^h, y_d^h)$ capture the spatial and temporal relationships between activities s and d. Define the scaled distance between activities s and d at the n^{th} frame of s as

$$r_s(s(n), d) = \frac{D(O_s(n), O_d)}{R_s(n) + R_d}, \tag{7}$$

where $O_s(n)$ and $R_s(n)$ denote the center and radius of the motion region of activity s at its n^{th} frame and O_d and R_d denote the center and radius of the activity region of activity d. $D(\cdot)$ denotes the Euclidean distance. Then, the spatial relationship of s and d at the n^{th} frame is modeled by $sc_{sd}(n) = bin(r_s(s(n), d))$ as in Fig. 4 (a). The normalized histogram $sc_{s,d} = \frac{1}{N_f} \sum_{n=1}^{N_f} sc_{sd}(n)$ is the inter-activity spatial feature of activity s and d.

Let TC be defined by the following temporal relationships: n^{th} frame of s is before d, n^{th} frame of s is during d and n^{th} frame of s is after d. $tc_{sd}(n)$ is the temporal relationship of s and d at the n^{th} frame of s as shown in Fig. 4 (b). The normalized histogram $tc = \frac{1}{N_f} \sum_{n=1}^{N_f} tc_{sd}(n)$ is the inter-activity temporal context feature of activity s with respect to activity d.



(a) (b)

Fig. 4. (a) The image shows one example of inter-activity spatial relationship. The red circle indicates the motion region of s at this frame while the purple rectangle indicates the activity region of d. Assume SC is defined by quantizing and grouping $r_s(n)$ into three bins: $r_s(n) \leq 0.5$ (s and d is at the same spatial position at the n^{th} frame of s), $0.5 < r_s(n) < 1.5$ (s is near d at the n^{th} frame of s) and $r_s(n) \geq 1.5$ (s is far away from d at the n^{th} frame of s). In the image, $r_s(n) > 1.5$, so, $sc_{sd}(n) = [0 \quad 0 \quad 1]$. (b) The image shows one example of inter-activity temporal relationship. The n^{th} frame of s occurs before d. So, $t_{sd}(n) = [1 \quad 0 \quad 0]$.

4 Model Learning and Inference

The parameters of the overall potential function $\psi(\mathbf{y}|\mathbf{x}, \omega)$ for the two-layer hierarchical CRF include ω_v^a, ω_ε^a, $\omega_{c,l}^{ah}$, $\omega_{c,f}^{ah}$, ω_{sc}^h and ω_{tc}^h. We define the weight vector as the concatenation of these parameters:

$$\omega = [\omega_v^a, \omega_\varepsilon^a, \omega_{c,l}^{ah}, \omega_{c,f}^{ah}, \omega_{sc}^h, \omega_{tc}^h]. \tag{8}$$

Thus, the potential function, $\psi(\mathbf{y}|\mathbf{x}, \omega)$, can be converted into a linear function with a single parameter ω as

$$\psi(\mathbf{y}^a) = \max_{\mathbf{y}^h} \omega^T \Gamma(\mathbf{x}, \mathbf{y}^a, \mathbf{y}^h), \tag{9}$$

where $\Gamma(\mathbf{x}, \mathbf{y}^a, \mathbf{y}^h)$, called the joint feature of activity set $\mathbf{x}$, can be easily obtained by concatenating various feature vectors in (4),(5) and (6).

4.1 Learning Model Parameters

Suppose we have P activity sets for learning. Let the training set be $(X, Y^a, Y^h) = (\mathbf{x}^1, \mathbf{y}^{1,a}, \mathbf{y}^{1,h}), \ldots, (\mathbf{x}^P, \mathbf{y}^{P,a}, \mathbf{y}^{P,h})$, where $\mathbf{x}^i$ denotes the i^{th} activity set as well as the observed features of the set. $\mathbf{y}^{i,a}$ is the label vector in the action layer and $\mathbf{y}^{i,h}$ is the label vector in the hidden activity layer. While there are various ways of learning the model parameters, we choose a task-oriented discriminative approach. We would like to train the model in such a way that it increases the average precision scores on a training data and thus tend to produce the correct activity labels for each action segment.

A natural way to learn the model parameter ω is to adopt the latent structural SVM. The loss $\Delta(\mathbf{x}^i, \hat{\mathbf{y}}^{i,a})$ of labeling

$\mathbf{x}^i$ with $\widehat{\mathbf{y}}^{i,a}$ in the action layer equals the number of action segments that associate with incorrect activity labels (an action segment is mislabeled if over half of the segment is mislabeled). From the construction of the higher-order potentials in section 3.2.2, it is observed that, in order to achieve the best labeling of the action segments, the optimum latent activity label of an action clique must be the dominant ground truth label l_c of its child nodes in the action layer; or the free label l_f if no dominant label exists for the action clique. Thus the loss $\Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,h})$ of labeling the activity layer of $\mathbf{x}^i$ with $\widehat{\mathbf{y}}^{i,h}$ is

$$\Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,h}) = \sum_{c \in \mathscr{V}^h} I(y_c^{i,h} \neq \{l_c^i, l_f\}), \tag{10}$$

where $I(\cdot)$ is the indicator function which equals 1 if the inside equation is satisfied and 0 otherwise. (10) counts the number of activity labels in $\widehat{\mathbf{y}}^{i,h}$ that are neither a free label nor the dominant label of its child nodes. Finally, the loss function of assigning $\mathbf{x}^i$ with $(\widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h})$ is defined as the summation of the two, that is

$$\Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}) = \Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}) + \Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,h}). \tag{11}$$

Next, we define a convex function $F(\omega)$ and a concave function $J(\omega)$ as

$$F(\omega) = \frac{1}{2}\omega^T \omega \tag{12}$$
$$+ C \sum_{i=1}^{P} \max_{(\widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h})} \left[\omega^T \Gamma\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right) + \Delta\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right) \right],$$
$$\text{and} \quad J(\omega) = -C \sum_{i=1}^{P} \max_{\mathbf{y}^{i,h}} \omega^T \Gamma\left(\mathbf{x}^i, \mathbf{y}^{i,a}, \mathbf{y}^{i,h}\right).$$

The model learning problem is given as:

$$\omega^* = \arg\min_{\omega} [F(\omega) + J(\omega)] \tag{13}$$

Although the objective function to be minimized in (13) is not convex, it is a combination of a convex function and a concave function [29]. Such kind of problems can be solved using the Concave-Convex Procedure (CCCP) [40], [41]. We describe an algorithm similar to the CCCP in [40] that iteratively infers the latent variables $\mathbf{y}^{i,h}$ for $i = 1, \ldots, P$ and optimizes the weight vector ω. The inference and optimization procedures continue until convergence or a predefined maximum number of iterations is reached.

The limitation of all learning algorithms that involve gradient optimization is that they are susceptible to local extrema and saddle points [18]. Thus, the performance of the proposed latent structural model is sensitive to initialization. There have been many works dealing with the problem of learning the parameters of hierarchical models [10], [36]. We use a coarse to fine scheme that separately initializes the model parameters using piecewise training, and then refines the model parameters jointly in a globally optimum manner. Specifically, the separately learned model parameters are used as the initialization values for the proposed learning algorithm. Given the weakly labeled training data with activity labels for each action segment, the dominant label l_c for each action clique can be determined. We initialize the latent activity variable of c with the dominant label l_c of its action clique c^a, and with l_f if there is no dominant label for c^a.

In the "E step", we infer latent variables using the previously learned weight vector ω_t (or the initially assigned weight vector for the first iteration) leading to

$$\mathbf{y}_{t+1}^{i,h*} = \arg\max_{\mathbf{y}^{i,h}} \omega_t{}^T \Gamma\left(\mathbf{x}^i, \mathbf{y}^{i,a}, \mathbf{y}^{i,h}\right). \tag{14}$$

Then, in the "M step", with the inferred latent variable $\mathbf{y}_{t+1}^{i,h*}$, we solve a fully visible structural SVM (SSVM). Let us define the risk function at iteration $t+1$, $\Lambda(\omega)$, as

$$\Lambda_{t+1}(\omega) = C \sum_{i=1}^{P} \max_{(\widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h})} \Big\{ \Delta\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right) \tag{15}$$
$$+ \omega^T \left[\Gamma\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right) - \Gamma\left(\mathbf{x}^i, \mathbf{y}^{i,a}, \mathbf{y}_{t+1}^{i,h*}\right) \right] \Big\}.$$

Thus, the optimization problem in (13) is converted to a fully visible SSVM as

$$\omega_{t+1}^* = \arg\min_{\omega} \left\{ \frac{1}{2}\omega^T \omega + \Lambda_{t+1}(\omega) \right\}. \tag{16}$$

The problem in (16) can be converted to an unconstrained convex optimization problem [44] and solved by the modified bundle method in [38]. The algorithm iteratively searches for the increasingly tight quadratic upper and lower cutting planes of the objective function until the gap between the two bounds reaches a predefined threshold. The algorithm is effective because of its very high convergence rate [37]. The visible SSVM learning algorithm specified for our problem is summarized in Algorithm 1.

Algorithm 1 Learning the model parameter in (16) through bundle method

```
Input:   $S = ((a_T(1), y_T(1)), \ldots, (a_T(P), y_T(P))), \omega_t^*, \mathbf{y}_{t+1}^{i,h*}, C, \varepsilon$
Output:  Optimum model parameter $\omega_{t+1}^*$
initialize $\omega_{t+1}^0$ with $\omega_t^*$, $\mathscr{G}_{t+1}$(cutting plane set) $\leftarrow \emptyset$.
for $k = 0$ to $\infty$ do
    for $i = 1, \ldots, P$ do
        find the most violated label vector for each training instance,
        if any, using $\omega_{t+1}^k$ (the value of $\omega_{t+1}$ at the $k^{th}$ iteration);
    end for
    find the cutting plane $g_{\omega_{t+1}^k}$ of $\Lambda(\omega)$ at $\omega_{t+1}^k$:
    $g_{\omega_{t+1}^k} = \omega^T \partial_\omega \Lambda_{t+1}(\omega_{t+1}^k) + b_{\omega_{t+1}^k}$,
    where  $b_{\omega_{t+1}^k} = \Lambda_{t+1}(\omega_{t+1}^k) - {\omega_{t+1}^k}^T \partial_\omega \Lambda(\omega_{t+1}^k)$.
    $\mathscr{G}_{t+1} \leftarrow \mathscr{G}_{t+1} \cup g_{\omega_{t+1}^k}(\omega)$;
    update $\omega_{t+1}$:   $\omega_{t+1}^{k+1} = \arg\min_\omega F_{\omega_{t+1}^k}(\omega)$,
    where   $F_{\omega_{t+1}^k}(\omega) = \frac{1}{2}\omega^T \omega + max(0, max_{j=1,\ldots,k} g_{\omega_{t+1}^j}(\omega))$.
    $gap_{k+1} = \min_{k' \leq k} F_{\omega_{t+1}^{k'+1}}(\omega_{t+1}^{k'+1}) - F_{\omega_{t+1}^k}(\omega_{t+1}^{k+1})$;
    if $gap_{k+1} \leq \varepsilon$, then return $\omega_{t+1}^* = \omega_{t+1}^{k+1}$;
end for
```

4.2 Inference

Suppose the model parameter vector ω is given. We now describe how to identify the optimum label vector $\mathbf{y}^a$ for a test instance $\mathbf{x}$ that maximizes (9). The inference problem is generally NP hard for multi-class problems, thus MAP inference algorithms, such as loopy belief propagation [29], are slow to converge. We propose an approximation method that alternatively optimizes the hidden variable $\mathbf{y}^h$ and the label vector $\mathbf{y}^a$. Such an algorithm is guaranteed to increase the objective at every iteration [29]. Let us define the activity layer potential function as

$$\psi^h(\mathbf{y}^h) = \sum_{c \in C^a} \psi(\mathbf{y}_c^a|\mathbf{x},\omega) + \psi(\mathbf{y}^h|\mathbf{x},\omega). \tag{17}$$

For each iteration, with current predicted label vector $\mathbf{y}^a$ fixed, the inference sub-problem is to find the $\mathbf{y}^h$ that maximizes $\psi^h(\mathbf{y}^h)$. An efficient greedy search method is used to find the optimum $\mathbf{y}^h$ as described in Algorithm 2. In order to simplify the inference, we force the edge weights between non-adjacent actions to be zeros. With the inferred hidden variable $\mathbf{y}^h$, the model is reduced to a one-layer discriminative CRF. The inference sub-problem of finding the optimum $\mathbf{y}^a$ can now be solved by computing the exact mixed integer solution. We initialize the process by holding the hidden variable fixed using the values obtained from automatic activity detection. The process continues until convergence or a predefined maximum number of iterations is reached.

Algorithm 2 Greedy Search Algorithm for the sub-problem of finding optimum hidden variable $\mathbf{y}^h$

```
Input:    Testing instance with action layer labels $\mathbf{y}^a$
Output:   Hidden variable labels $\mathbf{y}^h$
1) initialize $(\mathcal{V}^h, \mathbf{y}^h) \leftarrow \{\emptyset, \emptyset\}$ and $\psi^h = 0$.
2) repeat
      $\Delta\psi^h(y_c^h)_{c \notin \mathcal{V}^h} = \psi(\mathbf{y}^h \cup y_c^h) - \psi(\mathbf{y}^h)$;
      $y_c^{h^{opt}} = \arg\max_{c \notin \mathcal{V}^h} \Delta\psi^h(y_c^h)$;
      $(\mathcal{V}^h, \mathbf{y}^h) \leftarrow (\mathcal{V}^h, \mathbf{y}^h) \cup (c, y_c^{h^{opt}})$;
3) end if all activities are labeled.
```

4.2.1 Analysis of Computational Complexity

We now discuss the computational complexity of inference for a particular activity set consists of n action segments and m activities. Assuming there are M activity classes in the problem. For the graphical model in [45], the time complexity of the inference as discussed in the paper is $O(d_{max}n^2M)$, where d_{max} is the maximum number of action segments one activity may have. The inference on both the higher-order CRF and hierarchical-CRF is carried out layer-by-layer, and so the overall time complexity is linear in the number of layers used. Specifically, we use two-layer CRFs with an action layer and an activity layer. For the higher-order CRF model, inference on the activity layer takes $O(mM)$ computation to obtain the activity labels for each candidate activity. With the inferred activity labels, inference on the action layer takes $O(nM^2)$, since the model is reduced to a chain-CRF. For the hierarchical-CRF, the increase of computational complexity over the higher-order CRF lies in the inference on the activity layer, because the activities are connected with each other in this model. Using the proposed greedy search algorithm, the time complexity for inference on the activity layer is $O(m^2M)$. Thus, the overall complexity of inference is $O[T \cdot ((mM) + O(nM^2))]$ for higher-order CRF and $O[T \cdot ((m^2M) + O(nM^2))]$ for hierarchical-CRF, where T is the number of iterations. Furthermore, the number of action segments n is usually several times of the number of activities, that is $n = \alpha m$, where α is a small positive value larger that one. d_{max} and T are small positive value larger than one. Assuming n, m and M are in the same order, which is a reasonable assumption for our case, the asymptotic computational complexity of the model in [45] and the compared higher-order CRF and hierarchical-CRF models is of the same order.

5 Experimental Results

The goal of our framework is to locate and recognize activities of interest in continuous videos using both motion and context information about the activities; therefore, datasets with segmented video clips or independent activities like Weizmann [11], KTH [31], UT-Interaction Dataset [30] and Collective Activity Dataset [7] do not fit our evaluation goal. To assess the effectiveness of our framework in activity modeling and recognition, we perform experiments on two challenging datasets containing long duration videos: the UCLA office Dataset [32] and VIRAT Ground Dataset [9].

5.1 Motion Segmentation and Activity Localization

We first develop an automatic motion segmentation algorithm by detecting boundaries where the statistics of motion features change dramatically, and thus obtain the action segments. Let two NDMs be denoted as M_1 and M_2, and d_s be the dimension of the hidden states. The distance between the models can be measured by the normalized geodesic distance $dist(M_1, M_2) = \frac{4}{d_s\pi^2}\sum_{i=1}^{d_s}{\theta_i}^2$, where θ_i is the principal subspace angle (please refer to [5] for details on the distance computation).

A sliding window of size T_s, where T_s is the number of temporal bins in the window, is applied to each detected motion region along time. A NDM $M(t)$ is built for the time window centered at the t^{th} temporal bin. Since an action can be modeled as one dynamic model, the model distances between subsequences from the same action should be small, compared to those of subsequences from a different action. Suppose an activity starts from temporal bin k; the average model distance between temporal bin $j > k$ and k is defined as the weighted average distance between model j and neighboring models of k as

$$DE_k(j) = \sum_{i=0}^{T_d - 1} \gamma_i \cdot dist(M(k+i), M(j)), \tag{18}$$

where T_d is the number of neighboring bins used, and γ_i is the smoothing weight for model $k+i$ that decreases along time. When the average model distance grows above a predefined threshold d_{th}, an action boundary is detected. Action segments along tracks are thus obtained.

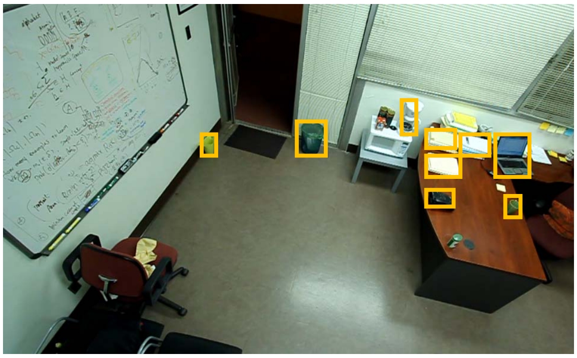
A multi-class SVM is trained upon the intra-activity features (as described in Section 3.3) of activities of different classes. After obtaining the action segments, we use the sliding window method with the trained multi-class SVM to group adjacent action segments into candidate activities. To speed up, we only work on candidate activities with confidence scores larger than a predefined threshold, indicating they are likely to be of activity classes of interest.

5.2 UCLA Dataset

The UCLA Office Dataset [32] consists of indoor and outdoor videos of single activities and person-person interactions. Here, we perform experiments on the videos of office scene containing about 35 minutes of activities in an office room that captured with a single fixed camera. We identify 10 frequent activities as the activities of interest:1 - enter room, 2 - exit room, 3 - sit down, 4 - stand up, 5 - work on laptop, 6 - work on paper, 7 - throw trash, 8 - pour drink, 9 - pick phone, 10 - place phone down. Each activity occurs 9 to 26 times in the dataset. Since the dataset contains only single person activities, it is natural to model activities in one sequence together. The dataset is divided into 8 sets, each set contains 2 sequences of activities and each sequence contains 2 to 19 activities of interest, as well as varying number of background activities. We use leave-one-set-out cross validation for the evaluation: use 7 sets for training and 1 set for testing.

5.2.1 Preprocessing

Intra-activity context feature is based on interactions between the agent and the surroundings. In the office dataset, there are 7 classes of objects that are frequently involved in the activities of interest: laptop, garbage can, papers, phone, coffee maker and cup. Fig. 5 shows the detected objects of interest in the office room. Since the UCLA Dataset consists of single person activities, the intra-activity attributes considered include agent-object interactions and their relative locations. We identify $(N_G = 10)$ subsets of attributes for the development of intra-activity context features in the experiment as shown in Fig. 6. For a given activity, the above attributes are determined from image-level detection results. The locations of objects are automatically tracked. Similar to [32], if enough skin color is detected within the areas of laptop, paper and phone,the corresponding attributes are considered as true. Fig. 7 shows examples of detected agent-object interactions.

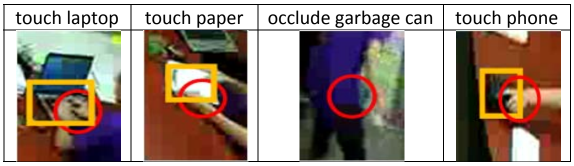


(a)

Fig. 5. Detected objects of interest in the UCLA office scene.

Attribute Subset	Associated Attributes
G_1 G_2 G_3	the agent is touching / not touching laptop[1], paper[2], phone[3].
G_4 G_5	the agent is occluding / not occluding the garbage can[4], coffee maker[5].
G_6 G_7 G_8	the agent is near / far away from the garbage can[6], coffee maker[7], door[8].
G_9	the agent disappears / not disappears at the door.
G_{10}	the agent appears / not appears at the door.

Fig. 6. Subsets of context attributes used for the development of intra-activity context features for UCLA Dataset (the superscripts indicates the correspondence between the subsets and the objects).



(a)

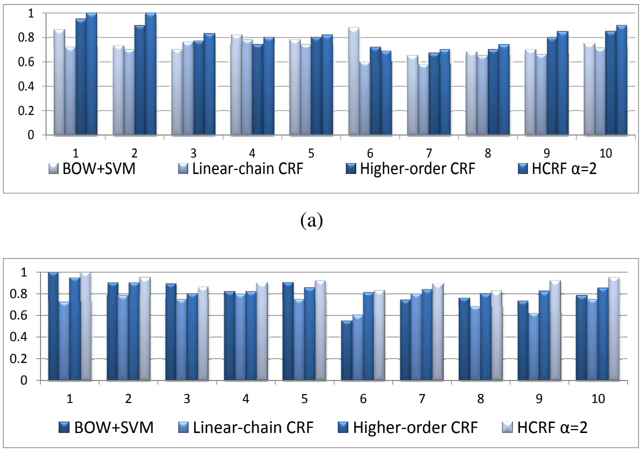
Fig. 7. Examples of agent-object interactions detected from image.

Whether the agent is near or far away from an object is determined by the distance between the two based on normal distributions of the distances of the two scenarios. Probabilities indicating how likely the agent is near or far away from an object are thus obtained. For frame n of an activity, we obtain $g_i(n) = I(G_i(n))$, where $I(\cdot)$ is the indicator function. $g_i(n)$ is then normalized so that its elements sum to 1.

Related candidate activities are connected. Whether two activities are related can be naturally determined by their temporal distances. One way to decide if the relationships between two candidate activities should be modeled is to see if they are in the α-neighborhood of each other in time. Two activities are said to be in the α-neighborhood of each other if there are less than α other activities occurring between the two.

5.2.2 Experimental Results

Although UCLA Dataset has been used in [32], the recognition accuracy for the office dataset has not been provided in the paper. We compare the performance of the popular BOW+SVM classifier and our model. The experiment results in precision and recall as shown in Fig. 8. In order to show the affects of incorporating different kinds of motion and context features, we also show results of using the action-based linear-chain CRF approach and the action-based higher-order CRF approach (Fig. 3 (a) and 3 (b)). It can be seen that the use of intra-activity context increases the recognition accuracy of activities with obvious context patterns. For example, "enter room" is characterized by the context that the agent appears at the door. The increased recognition accuracy of "enter room" by using intra-activity context features indicates that our model successfully captures this characteristics. From the performance of higher-order CRF approach and Hierarchical-CRF approach, we can see that for activities with strong spatio-temporal patterns, such as "pick phone" and "place phone down", modeling the inter-activity spatio-temporal relationships increases the recognition accuracy significantly. Next, we change the value of α to see how it influences the recognition accuracy of the Hierarchical-CRF approach. Fig. 9 compares the overall accuracy of different methods and the Hierarchical-CRF approach with different α values. From the results, we can see that Hierarchical-CRF approach with $\alpha = 2$ outperforms other models. This is expected. When α is too small, the spatio-temporal relationships of related activities are not fully utilized, while Hierarchical-CRF with fully connected activity layer models the spatio-temporal relationships of unrelated activities. For instance, in the UCLA office Dataset, one typical temporal pattern of activities is a person sits down to work on the laptop, then, the same person stands up to do other things, and then sits down to work on the laptop. All these activities are conducted sequentially. Thus, Hierarchical-CRF model with fully connected activity layer captures the false temporal pattern of "stand up" followed by "work on the laptop". The optimum value of α can be obtained using cross validation on the training data.



(a)

(b)

Fig. 8. Precision (a) and recall (b) for the ten activities in UCLA Office Dataset. The activities are defined in Section 5.2. HCRF is the short of Hierarchical-CRF.

Method	Overall	Average per-class
BOW+SVM	82.2	80.7
Linear-chain CRF	73.6	72.5
Higher-order CRF	83.9	84.3
HCRF ($\alpha = 1$)	87.9	87.1
HCRF ($\alpha = 2$)	89.9	90.8
HCRF (fully connected)	73.5	74.2

Fig. 9. Overall and average per-class accuracy for different methods on UCLA Office Dataset. The BOW+SVM method is tested on video clips, while other results are in the framework of our proposed action-based CRF models upon automatically detected action segments. HCRF is the short of Hierarchical-CRF.

5.3 VIRAT Ground Dataset

The VIRAT Ground Dataset is a state-of-the-art activity dataset with many challenging characteristics, such as wide variation in the activities and clutter in the scene. The dataset consists of surveillance videos of realistic scenes with different scales and resolution, each lasting 2 to 15 minutes and containing upto 30 events. The activities defined in Release 1 include 1 - person loading an object to a vehicle; 2 - person unloading an object from a vehicle; 3 - person opening a vehicle trunk; 4 - person closing a vehicle trunk; 5 - person getting into a vehicle; 6 - person getting out of a vehicle. We work on the all the scenes in Release 1 except scene 0002 and use half of the data for training and the rest for testing. Five more activities are defined in VIRAT Release 2 as: 7 - person gesturing; 8 - person carrying an object; 9 - person running; 10 - person entering a facility; 11 - person exiting a facility. We work on the all the scenes in Release 2 except scene 0002 and 0102, and use two-third of the data for training and the rest for testing.

5.3.1 *Preprocessing*

Motion regions that do not involve people are excluded from the experiments since we are only interested in person activities and person-vehicle interactions. For the development of STIP histograms, nearest neighbor soft-weighting scheme [25] is used.

Since we work on the VIRAT Dataset with individual person activities and person-object interactions, we use the following $N_G = 7$ subsets of attributes for the development of intra-activity context features in the experiments as shown in Fig. 10.

Persons and vehicles are detected based on the part-based object detection method in [9]. Opening/closing entrance/exit doors of facilities, boxes and bags are detected using method in [6] with binary linear-SVM as the classifier. Using these high-level image features, we follow the description in Section 5.2.1 to develop the feature descriptors for each activity set. The first three sets of attributes in Fig. 10 are used for the experiments on Release 1, and all are used for the experiments on Release 2. Fig. 11 shows examples of $g_i(n)$ defined as in Section 5.2.1 for different activities in VIRAT. Since, in VIRAT, activities are naturally related to each other, the activity layer nodes are fully connected to utilize the spatio-temporal relationships of activities occurring in the same local space-time volume.

5.4 Recognition Results on VIRAT Release 1

Fig. 12 compares the precision and recall for the six activities defined in VIRAT Release 1 using BOW+SVM method and our approach with different kinds of features. The results show, as expected, the recognition accuracy

Subset	Associated Attributes
G_1	moving object is a person; moving object is a vehicle trunk; moving object is of other kind.
G_2	the agent is at the body of the interacting vehicle; the agent is at the rear/head of the interacting vehicle; the agent is far away from the vehicles.
G_3	the agent disappears at the body of the interacting vehicle; the agent appears at the body of the interacting vehicle; none of the two.
G_4	the agent disappears at the entrance of a facility; the agent appears at the exit of a facility; none of the two.
G_5	velocity of the agent (in pixel) is larger than a predefined threshold; velocity of object of interest is smaller than a predefine threshold.
G_6	the activity occurs at parking areas; the activity occurs at other areas.
G_7	an object (e.g. bag/box) is detected on the agent; no object is detected on the agent.

Fig. 10. Subsets of context attributes used for the development of intra-activity context features.

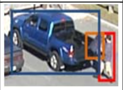
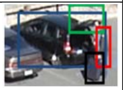
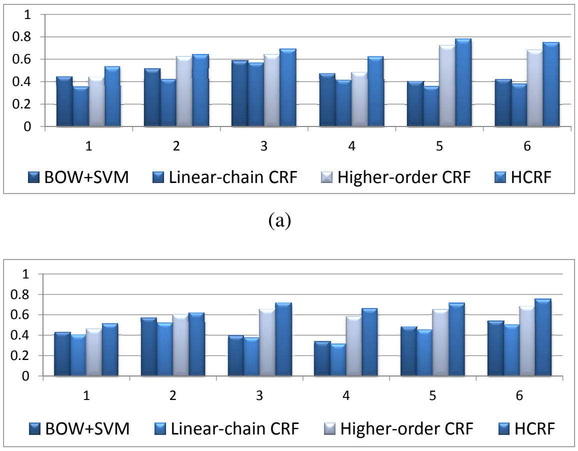
Activity	person loading	person unloading	opening trunk	closing trunk
Example Image				
$g_1(n)$	$[\frac{1}{2} \quad 0 \quad \frac{1}{2}]$	$[1 \quad 0 \quad 0]$	$[\frac{1}{2} \quad \frac{1}{2} \quad 0]$	$[\frac{1}{2} \quad \frac{1}{2} \quad 0]$
$g_2(n)$	$[0 \quad 1 \quad 0]$	$[0 \quad 1 \quad 0]$	$[0 \quad 1 \quad 0]$	$[0 \quad 1 \quad 0]$
$g_5(n)$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$
$g_6(n)$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$
$g_7(n)$	$[1 \quad 0]$	$[0 \quad 1]$	$[0 \quad 1]$	$[1 \quad 0]$
Activity	getting into vehicle	getting out of vehicle	gesturing	carrying object
Example Image				
$g_1(n)$	$[1 \quad 0 \quad 0]$	$[1 \quad 0 \quad 0]$	$[1 \quad 0 \quad 0]$	$[\frac{1}{2} \quad 0 \quad \frac{1}{2}]$
$g_2(n)$	$[1 \quad 0 \quad 0]$	$[1 \quad 0 \quad 0]$	$[0 \quad 0 \quad 1]$	$[0 \quad 0 \quad 1]$
$g_5(n)$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$
$g_6(n)$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$
$g_7(n)$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$	$[1 \quad 0]$

Fig. 11. Examples of detected intra-activity context features. The example images are shown with detected high-level image features. Object in red bounding box is a moving person; object in blue bounding box is a static vehicle; object in orange bounding box is a moving object of other kind; object in black bounding box is a bag/box on the agent.

increases by encoding the various context features. For instance, the higher-order CRF approach encodes intra-activity context patterns of activities of interest. Thus, activities with strong intra-activity context pattern, such as "person getting into vehicle", are better recognized by the higher-order CRF approach than by the linea-chain CRF approach, which does not model intra-activity context of activities. The Hierarchical-CRF approach further encodes inter-activity context patterns of activities. Thus, activities with strong spatio-temporal relationships with each other are better recognized by the Hierarchical-CRF approach. For instance, the higher-order CRF approach often confuses "open a vehicle trunk" and "close a vehicle trunk" with each other. However, if the two activities happen closely in time in the same place, the first activity in time is probably "open a vehicle trunk". This kind of contextual information within and across activity classes are captured by the Hierarchical-CRF approach and used to improve the recognition performance. Fig. 13 shows examples that demonstrate the significance of context in activity recognition.



(a)

(b)

Fig. 12. Precision (a) and recall (b) for the six activities defined in VIRAT Release 1.

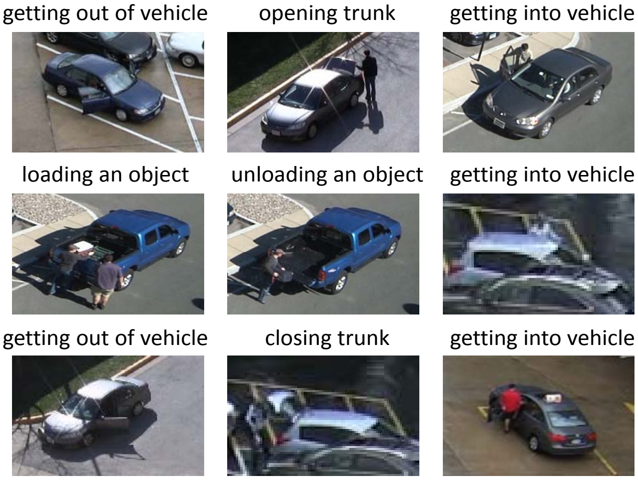


Fig. 13. Example activities (defined in VIRAT Release 1) correctly recognized by action-based linear-chain CRF (top), incorrectly by linear-chain CRF but corrected using higher-order CRF with intra-activity context (middle), and incorrectly recognized by higher-order CRF, but rectified using action-based hierarchical CRF with inter-activity context (bottom).

We also show the results on VIRAT Release 1 for different methods using overall and average accuracy in Fig. 14. We have compared our results with the popular BOW+SVM approach, the more recently proposed String-of-Feature-Graphs approach [12], [43] and structural model in [45].

Method	average accuracy
BOW+SVM [25]	45.8
SFG [44]	57.6
Structural Model [45]	62.9
Linear-chain CRF	42.6
Higher-order CRF	60.4
Hierarchical-CRF	66.2

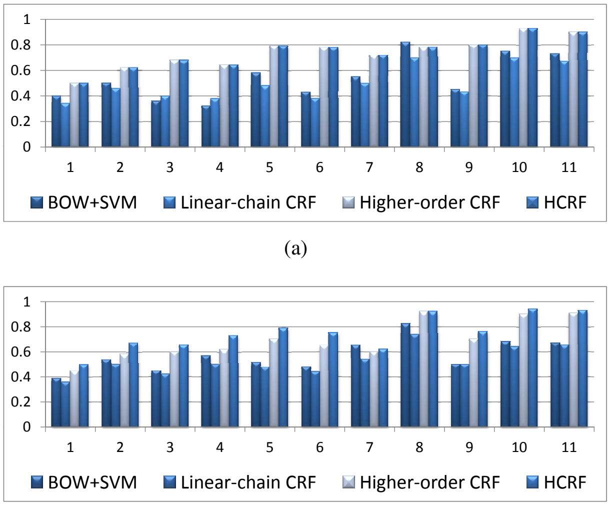
Fig. 14. Average accuracy for the six activities defined in VIRAT Release 1. Note that SVM+BOW works on video clips; while other methods work on continuous videos. Note that BOW+SVM works on video clip while others work on continuous video.

The Hierarchical-CRF approach outperforms the other methods. The results are expected since the intra-activity and inter-activity context within and between action and activities gives the model additional information about the activities of interest beyond the motion information encoded in low-level features. SFG approach models the spatial and temporal relationships between the low-level features and thus takes into account the local structure of the scene; However, it does not consider the relationships between various activities and thus our method outperforms the SFGs. Structural model in [45] models the intra and inter context within and between activities, however, it does not model the action layer and the interactions between action and activities.

5.5 Recognition Results on VIRAT Release 2

VIRAT Release 2 defines additional activities of interest. We work on VIRAT Release 2 to further evaluate the effectiveness of the proposed approach. We follow the method defined above to get the recognition results on this dataset. Fig. 15 compares the precision and recall for the eleven activities defined in VIRAT Release 2 for BOW+SVM method, the structural model in [45], and our method. We see that by modeling the relationships between activities, those with strong context patterns, such as "person closing a vehicle trunk"(4) and "person running"(9), achieve larger performance gain compared to activities with weak context patterns such as "person gesturing"(7). Fig. 16 shows example results on activities in Release 2.

Fig. 17 compares the recognition accuracy using recall for different methods. We can see that the performance of our Hierarchical-CRF approach is comparable to the recently proposed method in [1]. In [1], a SPN on BOW is learned to explore the context among motion features. However, [1] works on video clips, each containing an activity of interest with additional 10 seconds occurring randomly before or after the target activity instance, while we work on continuous video.



(a)

(b)

Fig. 15. Precision (a) and recall (b) for the eleven activities defined in VIRAT Release 2.

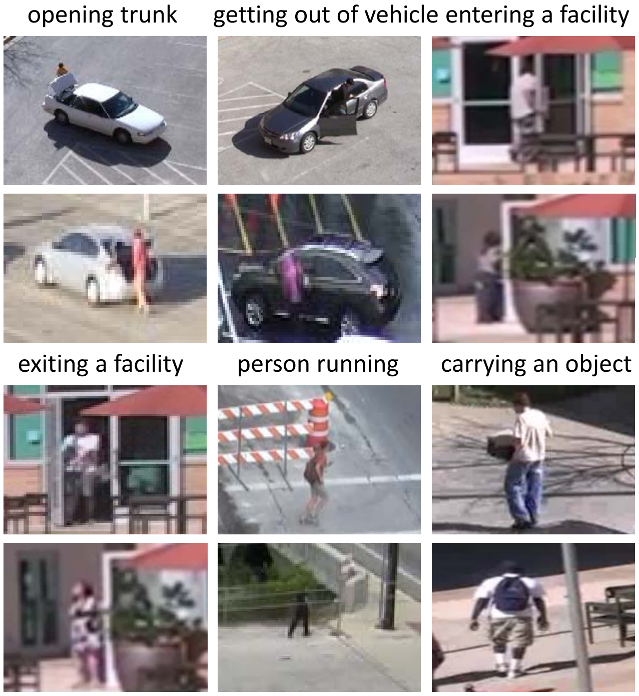


Fig. 16. Examples of recognition results (from VIRAT Release 2). For each two rows, examples in the bottom row show the effect of context features in correctly recognizing activities that were incorrectly recognized by the linear-chain CRF approach, while other examples of the same activities correctly recognized by the linear-chain CRF are shown in the top row.

6 Conclusion

In this paper, we design a framework for modeling and detection of activities in continuous videos. The proposed framework jointly models a variable number of activities in continuous videos, with action segments as the basic motion elements. The model explicitly learns the activity durations and motion patterns for each activity class as well as the context patterns within and across action and activities of different classes from training activity sets. It has been demonstrated that joint modeling of activities by encapsulating object interactions and spatial and temporal

Method	average accuracy
BOW+SVM [25]	55.4
SPN [1]	70
Structural Model [45]	73.5
Linear-chain CRF	52.5
Higher-order CRF	69.4
Hierarchical-CRF	75.1

Fig. 17. Average accuracy (in recall) for different methods.

relationships of activity classes can significantly improve the recognition accuracy.

It is worth noticing that more complex activities can be modeled by adding additional layers to the hierarchical model. However, the additional layers increase the learning and inference complexity by increasing the tree width. Balance between the representation power of the hierarchical model and the computational complexity of the model should be achieved.

References

[1] M. R. Amer and S. Todorovic. Sum-product networks for modeling activities with stochastic structure. In *CVPR*, 2012. 1, 2, 12, 13

[2] M. R. Amer, D. Xie, M. Zhao, S. Todorovic, and S.-C. Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *ECCV*, 2012. 3

[3] Y. B. and F. L. Modeling mutual context of object and human pose in human object interaction activities. In *CVPR*, 2010. 2

[4] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011. 3

[5] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, 2009. 1, 3, 8

[6] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012. 3

[7] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, 2011. 8

[8] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. In *International Journal of Computer Vision*, 2011. 3

[9] S. O. et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 8

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminative trained part based models. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010. 7

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as spatio-temporal shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Dec. 2007. 8

[12] U. Guar, Y. Zhu, B. Song, and A. K. Roy-Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural videos. In *ICCV*, 2011. 12

[13] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009. 3

[14] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009. 2

[15] S. Khamis, V. I. Morariu, and L. S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *ECCV*, 2012. 3

[16] P. Kohli, L. Ladicky, P. H.S., and Torr. Robust higher order potentials for enforcing label consistency. In *IJCV*, 2010. 3

[17] N. Komodakis. Learning to cluster using high order graphical models with latent variables. In *ICCV*, 2011. 3

[18] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009. 7

[19] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where & how many? combining object detectors and crfs. In *ECCV*, 2010. 3

[20] T. Lan, Y. Wang, S. N. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012. 3

[21] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, 2010. 3

[22] I. Laptev. On spatio-temporal interest points. In *International Journal of Computer Vision*, 2005. 3

[23] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 2

[24] V. I. Morariu and L. S. Davis. Multi-agent event recognition in structured scenarios. In *CVPR*, 2011. 2

[25] Y.-G. J. G.-W. Ngo and J. Yang. Towards optimal bag of features for object categorization and semantic video retrieval. *ACM-CIVR*, 2007. 2, 5, 10, 12, 13

[26] M. Nguyen, Z. Lan, and F. DellaTorre. Joint segmentation and classification of human actions in video. In *CVPR*, 2011. 3

[27] J. C. Niebles, H. Wang, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 1

[28] A. Oliva and A. Torralba. The role of context in object recognition. In *Trends in Cognitive Science*, 2007. 1

[29] K. p. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. 7, 8

[30] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010. 8

[31] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004. 8

[32] Z. Si, M. Pei, B. Yao, and S. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *ICCV*, 2011. 3, 8, 9

[33] B. Song, T. Jeng, E. Staudt, and A. Roy-Chowdury. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*, 2010. 3

[34] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009. 2

[35] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012. 2, 3

[36] B. Taskar, V. Chatalbashev, and D. Koller. Learning associative markov networks. In *ICML*, 2004. 7

[37] C. H. Teo, Q. Le, A. Smola, and S. V. N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In *SIGKDD*, pages 727–736, 2007. 7

[38] C. H. Teo, S. Vusgwanathan, A. Smola, and Q. V. Le. Bundle methods for regularized risk minimization. In *Journal of Machine Learning Research*, 2010. 7

[39] J. Wang, Z. Chen, and Y. Wu. Action recognition with multiscale spatio-temporal contexts. In *CVPR*, 2011. 2

[40] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *International Conference on Machine Learning*, 2009. 7

[41] A. L. Yuille and A. Rangarajan. The concave-convex procedure (CCCP). In *Neural Computation*, volume 15, April 2003. 7

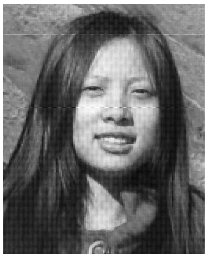
[42] E. Zheleva, L. Getoor, and S. Sarawagi. Higher-order graphical models for classification in social and affiliation networks. In *NIPS*, 2010. 3

[43] Y. Zhu, N. Nayak, U. Gaur, B. Song, and A. Roy-Chowdhury. Modeling multi-object interactions using "string of feature graphs". In *Computer Vision and Image Understanding*, 2013. 12

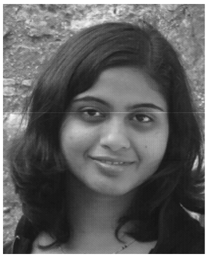
[44] Y. Zhu, N. Nayak, and A. Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. In *IEEE Journal of Selected Topics in Signal Processing*, 2013. 3, 4, 7, 12

[45] Y. Zhu, N. M. Nayak, and A. K. Roy-chowdhury. Context-aware modeling and recognition of activities in video. In *CVPR*, 2013. 3, 5, 8, 12, 13

[46] Z.Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *ICPR*, 2004. 3



Yingying Zhu Yingying Zhu received her Master's degree in Engineering in 2007 and 2010 from Shanghai Jiao Tong University and Washington State University, respectively. She is currently pursuing the Ph.D. degree, and is expected to receive the Ph.D. soon, within Intelligent Systems in the Department of Electrical and Computer Engineering at the University of California, Riverside. Her main research interests include pattern recognition and machine learning, computer vision, image/video processing and communication.



Nandita Nayak Nandita M. Nayak received her Bachelor's degree in Electronic and Communications Engineering from M. S. Ramaiah Institute of Technology, Bangalore, India in 2006 and her Master's degree in Computational Science from Indian Institute of Science, Bangalore, India. She received her Ph.D. degree from the Department of Computer Science and Engineering in University of California, Riverside. Her main research interests include image processing and analysis, computer vision and artificial intelligence.

Amit K. Roy-Chowdhury Amit K. Roy-Chowdhury received the Bachelors degree in Electrical Engineering from Jadavpur University, Calcutta, India, the Masters degree in systems science and automation from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree in Electrical Engineering from the University of Maryland, College Park. He is a Professor of Electrical and Computer Engineering at the University of California, Riverside. His research interests include image processing and analysis, computer vision, pattern recognition, and statistical signal processing. His current research projects include vision networks, distributed visual analysis, wide-area scene understanding, visual recognition and search, video-based biometrics (gait), and biological video analysis. He has authored the monograph Camera Networks: The Acquisition and Analysis of Videos over Wide Areas, and published about 150 papers and book chapters. He has been on the organizing and program committees of multiple conferences and serves on the editorial boards of a number of journals.