Learning Latent Spatio-Temporal Compositional Model for Human Action Recognition

Xiaodan Liang  
Sun Yat-Sen University, China  
xdliang328@gmail.com

Liang Lin  
Sun Yat-Sen University, China  
linliang@ieee.org

Liangliang Cao  
IBM Research, U.S.  
liangliang.cao@us.ibm.com

ABSTRACT

Action recognition is an important problem in multimedia understanding. This paper addresses this problem by building an expressive compositional action model. We model one action instance in the video with an ensemble of spatio-temporal compositions: a number of discrete temporal anchor frames, each of which is further decomposed to a layout of deformable parts. In this way, our model can identify a Spatio-Temporal And-Or Graph (STAOG) to represent the latent structure of actions, e.g., triple jumping, swinging and high jumping. The STAOG model comprises four layers: (i) a batch of leaf-nodes in bottom for detecting various action parts within video patches; (ii) the or-nodes over bottom, i.e., switch variables to activate their children leaf-nodes for structural variability; (iii) the and-nodes within an anchor frame for verifying spatial composition; and (iv) the root-node at top for aggregating scores over temporal anchor frames. Moreover, the contextual interactions are defined between leaf-nodes in both spatial and temporal domains. For model training, we develop a novel weakly supervised learning algorithm which iteratively determines the structural configuration (e.g., the production of leaf-nodes associated with the or-nodes) along with the optimization of multi-layer parameters. By fully exploiting spatio-temporal compositions and interactions, our approach handles well large intra-class action variance (e.g., different views, individual appearances, spatio-temporal structures). The experimental results on the challenging databases demonstrate superior performance of our approach over other competing methods.

Categories and Subject Descriptors

I.5 [Computing Methodologies]: Image Processing and Computer Vision  
I.4 [Computing Methodologies]: Pattern Recognition  
I.3 [Computing Methodologies]: Image Processing and Computer Vision

General Terms

Algorithms, Experimentation, Performance

Keywords

Video Understanding, Action Recognition, Structural Learning, And-Or Graph

1. INTRODUCTION

With the popularity of personal video cameras and multi-view video capturing devices, we are entering an era with rich amount of multimedia documents surrounding us. To interact with these videos, there have been increasing demands of understanding human activities in these videos. Although many research studies have been carried out to understand and retrieve large scale video contents, these works focus on high level semantics instead of describing human activities. There still exists a need to recognize fine-grained information for human activities, e.g., body poses and temporal motions.

This paper targets on the challenge of understanding spatial and temporal variances of video phenomenons. More specifically, we are considering the following difficulties:

- A human body is composed of multiple parts, and different parts are associated with different motions.
- Both the short term and long term motions may be fused by different background movement or camera motion, which brings the difficulties of accurately modeling the temporal characteristics for actions.
- In real world videos, human actions often happen with uncertainties: some happen along with occlusions, some are caused by view/pose variance, or due to diverse actor appearances and motions.

Due to these difficulties, it is crucial to reduce the spatio-temporal ambiguity when modeling the human actions. Most of the previous studies were built on simplified action models while overlooking the detailed spatio-temporal structure information. Of these works, a large amount of studies were based on spatio-temporal interest points [16, 42, 33]. Some researchers proposed to enrich the action model with the appearance information or context information [40, 34]. Some other researchers learned temporal structures for action recognition [28, 35]. However, none of these works provides an effective model which can unify spatial and temporal information to infer the structure of human motion.

We aims to develop an effective configurable model, namely the Spatio-Temporal And-Or Graph (STAOG) for action recognition, which addresses the problems mentioned above. Our idea is partially motivated by the image grammar model [47], which hierarchically decomposes an image pattern with mixed and-nodes and or-nodes, as well as modeling rich structural variations of parts.

The challenges of generalizing And-Or graphs for action recognition are two-folds. First, the traditional And-Or graphs are lim-
A novel weakly supervised learning algorithm for model training, inspired by the non-convex optimization techniques [43, 38]. This algorithm trains the model in a dynamic manner: the model structure (e.g., the configuration of leaf-nodes and or-nodes) is iteratively generated and reconfigured on the training data, with optimizing the multi-layer parameters. The other structure attributes (e.g., the activation of leaf-nodes, and temporal deformation of anchor frames) are modeled with the latent variables and optimized simultaneously.

Figure 1: An example of the Spatio-Temporal And-Or Graph model, where the and-nodes represent compositions in either time or space, the or-nodes indicate structural alternatives, and the leaf-nodes (at the bottom) correspond to local part detectors. The links between leaf-nodes represent spatio-temporal contextual interactions.
including a fixed number of part detectors as well as the predefined composition.

One unique characteristic of human action recognition problem lies in the temporal structure. A lot of works were proposed to build temporal structure models [9, 24, 35, 4] based on discriminative and interesting motion segments of the video. Raptis et al. [25] extracted clusters of trajectories and proposed a graphical model to incorporate constraints for individual and group events. Albanese et al. [1] represented temporal relations of activities using the probabilistic Petri Nets and integrated high-level reasoning approaches. Different from these approaches, we do not treat the whole temporal frames as units. Instead, we model temporal structure based on action parts with explicit relations and presents a solution to find both spatial and temporal configurations for dynamic activities.

Recently, the And-Or graph models [47] have been discussed for several vision tasks such as object recognition [20] and shape modeling [19]. These works mainly focused on images instead of videos and do not take the temporal dynamic structure into account. The very recent work by Amer et al. [3] proposed to recognize dynamic activities with the spatio-temporal And-Or graph model, but they over-simplified the model training by manually fixing the model structure (i.e. the layout of graph nodes).

It is worth mentioning that this paper learns the spatio-temporal graph without using any extra annotations or scripts. Research works which utilize rich annotation for event parsing and interpretation are beyond the scope of this work. In contrast, Marszalek et al. [23] explored the action contexts of natural dynamic scenes with movie scripts. Gupta et al. [11] proposed to learn a visually rich action grammar model for daily activities based on a predefined set of unary and binary relations. Extra annotations are also required for these studies.

### 3. SPATIO-TEMPORAL AND-OR GRAPH

The STAOG model is defined as $G = (V, E)$, where $V$ represents the four types of nodes and $E$ the graph edges as Fig. 1. The root node in top verifies the temporal composition, which aggregates scores over anchor frames. Each and-node represents a temporal anchor frame for verifying spatial composition. The or-nodes are derived from each and-node, which are “switch” variables for specifying the activation of their children leaf-nodes. The number of leaf-nodes for each or-node is dynamically learned with a predefined limit number $m$. For simplicity, we use $t = 1, \ldots, T$ to index all and-nodes in the whole STAOG model, $i = 1, \ldots, Z$ for or-nodes and $j = 1, \ldots, n$ for leaf-nodes. We also index the child or-node of and-node $A_i$ as $i = \text{ch}(i)$, and index the child leaf-node of or-node $U_j$ as $j = \text{ch}(i)$. The spatio-temporal graph edges (i.e., interactions) are defined between the leaf-nodes associated with different or-nodes. In this section, we describe two factors in detail: the spatio-temporal compositions, and the contextual interactions in both spatial and temporal domains.

#### 3.1 Spatio-Temporal Compositions

We employ Laptev’s 3-D corner detector [16] to detect interest points in video sequences, and each interest point is described by HoG (histogram of gradient) and HoF (histogram of optical flow) [17]. Furthermore, we generate a dictionary of spatio-temporal interest points’ descriptors, clustered by the k-means method in training stage. Given a video sequence $X$, we first equally divide it into $T$ temporal segments. The center frame in each video segment is chosen as an initial anchor frame. Each anchor frame is further decomposed into a number of action parts. In our method, we define the action parts based on the video patch representation, i.e. 3-D volumes spanning $\rho$ consecutive frames. Thus, for each anchor frame $I_t$, we observe a sequence of frames centered at $I_t$, and the sequence denoted as $L_t$ is treated as the input for anchor frame processing.

**Leaf-node:** Each leaf-node $L_j$ represents a local classifier for detecting action parts within video patches and includes two terms: an appearance feature $\phi^a$ and the spatial displacement feature $\phi^d$. Within an anchor frame $L_t$, the features of action parts are described by the BoW histogram based on the generated dictionary. Assume the action part detected by $L_j$ is localized at position $p_j = (p_{jx}, p_{jy})$, then $\phi(L_t, p_j)$ is denoted as the appearance feature. During detection, the locations of action parts are allowed to be perturbed to tackle the spatial deformation. We incorporate the spatial displacement $\phi^d(q_j, p_j) = (d_{jx}, d_{jy})$ for each action part, which can be computed by maximizing the response of $L_j$ during inference; $q_j$, representing the initialized position of $L_j$, is set according to the center-point of the frame. Thus the spatial displacement is defined as $\phi^d(q_j, p_j) = (d_{jx}, d_{jy})$, where $(d_{jx}, d_{jy})$ is the displacement.

The response of $L_j$ is defined as,

$$\omega^d_j \cdot \phi(L_t, p_j),$$

where $\omega^d_j$ is the parameter for the appearance feature and $\omega^d_j$ corresponds to the spatial deformation parameter.

**Or-node:** Each or-node $U_j$ is proposed to specify an appropriate candidate from its children leaf-nodes. For each leaf-node $L_j$ of $U_j$’s children, the indicator variable $v_j \in \{0, 1\}$ represents whether it is activated or not and each or-node only selects one leaf-node. Briefly, we utilize the indicator vector $v_j$ for the or-node $U_j$ and each element of $v_j$ is an indicator variable $v_j$ of the leaf-node $L_j$. Intuitively, the significant intra-class variance caused by views, background clutters or actors can be captured by different spatial configurations that are determined with the or-nodes.

The response of the or-node $U_j$ is defined as,

$$R^u_j(L_t, v_j) = \sum_{j \in \text{ch}(i)} R^a_j(L_t, p_j) \cdot v_j,$$

where $R^a_j(L_t, p_j)$ is the appearance feature of the anchor frame.

![Figure 2: Illustration of spatial compositions. (a) The black boxes denote the initial positions of action parts. (b) The parts are exhibited which are associated with a set of leaf-nodes. Each $(d_x, d_y)$ indicates the location displacement determined by the model. (c) The activated leaf-nodes are highlighted by red and spatial contextual interactions are defined between the pairwise spatial adjacent leaf nodes.](image-url)
And-node: Each and-node $A_t$ verifies the holistic appearance of action for the anchor frame $I_t$, and spatial composition of the or-nodes in its children. We define the configuration vector $V_t$ for all leaf-nodes within the anchor frame, which includes all indicator vectors $v_i$ corresponding to its children or-nodes $U_i$. The response of the and-node $A_t$ is defined as,

$$R^a_t(A_t, V_t) = \omega^a_t \cdot \phi^a(A_t) + \sum_{i \in \text{ch}(t)} R^a_i(A_i, v_i),$$

(3)

where $\phi^a(A_t)$ is the BoW histogram globally extracted from the 3-D volume $A_t$ centered at $I_t$. The second term aggregates the response scores from all or-nodes of $A_t$’s children.

Root-node: The root-node is a global potential function that verifies the temporal compatibility of model, including three terms: the global BoW histogram of the video clip, aggregated scores of its $T$ children and-nodes, and temporal displacements of anchor frames. We introduce the latent variable $\Delta_t$ to indicate the temporal ordering constraints which are crucial for discriminating human activities.

In particular, the temporal displacement penalty $\xi_t$ punishes the position of the and-node $A_t$ (i.e. one anchor frame) shifting far away from the initial anchor point $\tau_t$ in time. Once $\Delta_t$ is optimized, the position of each anchor frame can be determined by $\tau_t + \Delta_t$ accordingly. We define $\xi_t$ by,

$$\xi_t = -\omega^\tau_t \cdot \Delta_t,$$

(4)

where $\omega^\tau_t$ is the corresponding parameter. The response of the root-node can be then defined as,

$$R^r_t(X, V, \Delta) = \omega^r \cdot \phi^r(X) + \sum_{t=1}^T R^r_t(A_t, V_t) + \xi_t,$$

(5)

where $\phi^r(X)$ is the BoW histogram feature extracted from the whole video sequence $X$, $V = (V_1, \ldots, V_T)$ and $\Delta = (\Delta_1, \ldots, \Delta_T)$ are latent variables in the model for specifying the spatial and temporal configurations.

3.2 Contextual Interactions

Spatial interactions. We impose spatial contextual interactions, i.e. spatial edges, between pairwise spatially adjacent leaf-nodes in each anchor frame, as Fig. 2(c) illustrates. Note that we only link the edges between a pair of leaf-nodes that are respectively associated with two different or-nodes.

For one edge connecting two leaf-nodes $(L_j, L_j')$, we define it with a set of informative relations, i.e. a 8-bin binary feature $\phi^\varphi(L_j, L_j')$: above, below, left, right, near, far, clockwise and anti-clockwise between two adjacent leaf-nodes. The relations are visualized in Fig. 3. Suppose one edge connects two leaf-nodes $(L_j, L_j')$ which detect action parts at positions $p_j$ and $p_{j'}$ respectively. The centered red rectangle represents the location $p_j$, and the other red rectangles represent the adjacent parts. In the right chart of Fig. 3, the dashed line represents the initial layout of the two leaf-nodes, and the black solid line the adjusted actual layout during inference. Then we define the relations as,

- near or far: If $p_{j'}$ is fallen into the outer dashed ellipse, it is near to $p_j$, i.e. the bin of near is activated (i.e. set as 1); otherwise it is far to $p_j$.

- above, below, left or right: The corresponding bin is set as 1 only if the center of $p_{j'}$ is inside the corresponding dashed rectangles.

- clockwise or anti-clockwise: one of the two relations is activated (i.e. set as 1) according to the angle between the dashed line and the black solid line.

The relations intuitively encode the spatial contexts of two action parts detected via the two leaf-nodes with respect to two different or-nodes. The response of the pairwise potentials can be parameterized as,

$$\Gamma^\varphi_{jj'} = \beta^\varphi_{jj'} \cdot \phi^\varphi(L_j, L_j'),$$

(6)

where $\beta^\varphi_{jj'}$ is the corresponding 8-bin parameter vector.

Temporal interactions. We also impose the edges in temporal domain in our model to represent the temporal interactions of action parts. The edges connect temporally adjacent leaf-nodes, illustrated in Fig. 4. The edges are connected between any pair of leaf-nodes $(L_j, L_{j'})$ that belong to the same part within the two adjacent anchor frames respectively. A set of temporal relations are collected to concatenate a 4-bin binary feature vector $\varphi^\rho(L_j, L_{j'})$.

Specifically, we adopt four predicates: intersect, after, meets, interrupt, inspired by Allen’s temporal predicates [2] [26]. These predicates describe relations between two time intervals. The action part detected by one leaf-node $L_j$ for a specific anchor frame is described by the feature $\phi^\rho$ extracted from a 3-D volume with time span $\rho$. Note that we ignore some predicates of ordering such as before and equals, as the order of temporally adjacent anchor frames is supposed to be fixed. Assume that leaf-node $L_j$ is associated with the anchor frame localized at $\tau_t + \Delta_t$, (initial position plus displacement). The starting and ending time $(f^{start}_{L_j}, f^{end}_{L_j})$ of $L_j$ can be calculated as,

$$f^{start}_{L_j} = \tau_t + \Delta_t - \frac{\rho}{2},$$

$$f^{end}_{L_j} = \tau_t + \Delta_t + \frac{\rho}{2},$$

(7)

Then we can define the four temporal predicates for the two temporal adjacent leaf-nodes $(L_j, L_{j'})$ as,

$$\text{intersect}(L_j, L_{j'}) \iff f^{start}_{L_{j'}} < f^{end}_{L_j},$$

$$\text{after}(L_j, L_{j'}) \iff f^{end}_{L_j} < f^{start}_{L_{j'}} < f^{end}_{L_j} + \rho,$$

$$\text{meets}(L_j, L_{j'}) \iff f^{start}_{L_{j'}} = f^{end}_{L_j} + \rho,$$

$$\text{interrupt}(L_j, L_{j'}) \iff f^{end}_{L_j} + \rho < f^{start}_{L_{j'}}.$$

(8)
Thus, we define the response of one temporal edge linking two leaf-nodes accordingly,

\[ \Gamma^{*}_{j,j'} = \beta^{*}_{j,j'} \cdot \varphi^{*}(L_{j}, L_{j'}), \]  

(9)

where \( \beta^{*}_{j,j'} \) is the corresponding 4-bin parameter. If the pairwise leaf-nodes \( (L_{j}, L_{j'}) \) satisfies the specific predicate, the corresponding bin is set to 1, otherwise 0.

Therefore, the overall response of the STAOG model is:

\[ R^{\theta}(X, V, \Delta) = R^{c}(X, V, \Delta) + \sum_{j} \left( \sum_{j' \in \gamma^{*(j)}} \Gamma^{a}_{j,j'} \cdot v_{j} \cdot v_{j'} \right) + \sum_{j',j'' \in \gamma^{*(j)}} \Gamma^{j^{'}}_{j,j''} \cdot v_{j} \cdot v_{j''}, \]  

(10)

where \( (V, \Delta) \) are the hidden variables in the STAOG model. The second term defines the spatio-temporal contextual interactions between the leaf-nodes. \( \gamma^{*(j)} \) is denoted as the set of leaf-node \( L_{j} \)'s neighbors which are spatially adjacent to the leaf-node \( L_{j} \), and \( \gamma^{j^{'}}(j) \) is introduced for the leaf-nodes which are temporally adjacent to the leaf-node \( L_{j} \). Intuitively, spatial interactions between action parts guarantee the spatial coherence, as well as temporal interactions embedding the temporal contextual relations. Briefly, we refer \( L = (V, \Delta) \) as the latent variables in the following. The Eq.10 can be briefly written as:

\[ R^{\theta}(X, L) = \psi \cdot \Phi(X, L), \]  

(11)

where \( \psi \) includes the complete parameters of the STAOG model, and \( \Phi(X, L) \) denotes the overall feature vector.

4. INFERENCE

The inference task is to detect \( T \) optimal temporal anchor frames for one video instance as well as the spatial composition of action parts within each anchor frame. In our approach, we perform a cascaded search that integrates three steps: spatial testing, temporal testing and global verification to maximize the global potential \( R^{\theta}(X, L) \) defined in Eq.10.

**Step 1. Spatial Testing via the and-nodes.**

The subgraph of the STAOG model, rooted at one and-node, can be viewed as the spatial composition classifier for localizing action parts in one frame. We first use all existing leaf-nodes to search for candidate actions parts. Assume the leaf-node \( L_{j} \) associated with the frame \( I_{j} \) detects the action part at the position \( p_{j}^{*} \) by maximizing the response in Eq.1. Each or-node is allowed to activate only one leaf-node, then a possible configuration consisting of action parts is decided by the indicator variables of the or-node, (i.e. \( v_{i} \) for or-node \( U_{i} \)) indicating which leaf-node is activated. In this way, a set of possible configuration hypotheses \( \{ V_{l} \} \) are generated for further testing, which ensemble the hypotheses proposed by the or-nodes for the frame \( I_{t} \). In practice, we limit the maximum number of hypotheses by setting a threshold on \( R^{c}_{l}(L_{i}, V_{l}) \) in Eq.3.

**Algorithm 1 Inference Algorithm**

**Input:**
A learned STAOG model \( G \), the action parts detected by all leaf-nodes by maximizing the response in Eq.1.

**Initialization:**
The set of possible hypotheses \( l_{t} = \{ \} \) for all \( t = 1, \ldots, T \) anchor frames.

**Iteration:**

for all \( t = 1 \ldots T \) do

For each and-node \( A_{t} \), a set of temporal displacement steps \( \Sigma \) is predefined for sliding the possible anchor frames.

for all \( \Delta_{t}, \in \Sigma \) do

(a) initialize the set of pair terms \( Q = \{ Q_{1}, \ldots, Q_{K} \} \) for all or-nodes of \( A_{t} \)'s children.

(b) generate a set of pair terms \( Q_{i} \) for each or-node \( U_{i} \).

• for all \( U_{i}, i \in ch(t) \) do

  for all \( L_{j}, J \in ch(i) \) do

  \( Q_{i} = Q_{i} \cup (i, j) \).

  end for

• end for

(c) obtain possible hypotheses \( V_{i} = (v_{1}, \ldots, v_{K}) \) by assembling the indicator variables of \( K \) or-nodes according to the set \( Q \).

(d) The set of possible hypotheses for each specific displacement \( \Delta_{t} \) is constructed as \( l_{t} = l_{t} \cup \{ V_{l} \} \).

end for

Assemble these hypothesis \( \{ l_{t} \} \) for all \( T \) anchor frames orderly to generate the set of hypotheses sequence \( l \). Each possible configuration \( \{ V, \Delta \} \) belongs to the set \( l \). The global response of STAOG model can be calculated by Eq.12.

**Output:**
The latent variables \( V, \Delta \) and the final score \( S_{\psi}(X) \).

**Step 2. Temporal Testing via the root-node.**

We apply the spatial testing with the and-nodes to localize a number of candidate anchor frames over several frames. The scores over candidate anchor frames (proposed by the and-nodes) are aggregated via the root-node for a possible temporal composition. For efficiency, we utilize a fixed number of discrete steps \( \Sigma \) for searching each anchor frame. Several possible hypotheses are then produced with different anchor frame determinations by sliding the discrete steps \( \Sigma \). In addition, we re-weight the hypotheses at the root-node in Eq.5 by considering the temporal displacements of anchor frames as well as the the global features over the video clip. Intuitively, the hypotheses are represented by the specified latent variables \( \{ V, \Delta \} \).

**Step 3. Global Verification.**

Given all the hypotheses from the root-node, we apply the global potential function defined in Eq.10 to validate the optimal detection. The objective of the global verification is to cope with the noisy local detections on leaf-nodes. It combines the score of the
where \( R \) frame can be optimized by maximizing
the temporal anchor frames and spatial configurations for each
is determined by latent variables \( L \) or removed to reconfigure the model structure. The model structure
in a dynamic manner: the leaf-nodes can be automatically created
this model by a novel latent learning method extended from the
interactions (edges).

Let \( D = ((X_1, y_1), (X_2, y_2), \ldots, (X_N, y_N)) \) be a set of la-
abeled training samples, with \( y_k \in \{1, -1\} \). The feature vector for each \( X, y \) is defined as,
\[
\Phi(X, y, L) = \begin{cases} 
\Phi(X, L) & \text{if } y = +1, \\
0 & \text{if } y = -1.
\end{cases}
\] (13)
The temporal anchor frames and spatial configurations for each frame can be optimized by maximizing \( R^k(X, L) \) in the inference procedure. We refer to \( L = (V, \Delta) \) as the latent variables, then we redefine Eq.12 as,
\[
S_{\psi}(X) = \max_{\psi \in \mathcal{L}}(\psi \cdot \Phi(X, y, L)).
\] (14)
The optimization of this function can be solved by the latent struc-
tural SVM framework,
\[
\min_{\psi} \frac{1}{2} ||\psi||^2 + C \sum_{k=1}^{N} \max_{y \in \mathcal{L}}(\psi \cdot \Phi(X_k, y, L) + h(y_k, y)) - \max_{L \in \mathcal{L}}(\psi \cdot \Phi(X_k, y_k, L))
\] (15)
where \( C \) is a penalty parameter set as 0.003 empirically and \( h(y_k, y) \)
is the cost function, where \( h(y_k, y) = 0 \) if \( y_k = y \), otherwise 1. The optimization problem described above is not convex in general.
Following the CCCP framework, we convert the function in Eq.15 into a convex and concave form as,
\[
\min_{\psi} \frac{1}{2} ||\psi||^2 + C \sum_{k=1}^{N} \max_{y \in \mathcal{L}}(\psi \cdot \Phi(X_k, y, L) + h(y_k, y)) - [C \sum_{k=1}^{N} \max_{L \in \mathcal{L}}(\psi \cdot \Phi(X_k, y_k, L))] 
= \min_{\psi} [f(\psi) - g(\psi)],
\] (16)
where \( f(\psi) \) represents the first two terms, and \( g(\psi) \) the last term. This leads to an iterative learning algorithm that alternates estimating model parameters and the hidden variables \( L \). However, we still need to dynamically determine the graph configuration, i.e., the produc-
tion of leaf-nodes associated with or-nodes. An additional step for dynamically reconfiguring structure is added between the two original steps. The procedure is presented as follows.

(1) The model parameter \( \psi_i \) obtained in the previous iteration is fixed. We find a hyperplane \( q_i \) to upper bound the concave part \( g(\psi) \) in Eq. 16. Specifically, \( q_i \) is the derivative of \( g(\psi) \). Thus, we have \(-g(\psi) \leq -g(\psi_i) + (\psi - \psi_i) \cdot q_i, \forall \psi \). The optimal latent variable \( L_k^\ast \) is calculated by \( L_k^\ast = \argmax_{\psi \in \mathcal{L}}(\psi \cdot \Phi(X_k, y_k, L)) \) for each positive example. Then the hyperplane is constructed as \( q_i = -C \sum_{k=1}^{N} \Phi(X_k, y_k, L_k^\ast) \).

(II) In the second step, the STAOG model is adjusted by struc-
tural reconfiguration with applying the current model on training examples. The reconfiguration is performed for each part, i.e. or-
node, independently, with the fixed latent variable \( L_k^\ast = (V_k^\ast, \Delta_k^\ast) \). Note that each action part detected by one leaf-node is mapped to several feature bins at specific positions in the vector \( \Phi(X_k, y_k, L_k^\ast) \).

For each or-node \( U_k \), we apply its children leaf-nodes for detect-
ing parts in all positive samples. Assume that leaf-node \( L_j \) detects an action part on the \( k \)-th sample using the feature vector \( \pi_j^k \), which is a sub-vector of the complete feature vector of \( k \)-th sample, \( \Phi(X_k, y_k, L_k^\ast) \). And we obtain a set of feature vector \( \{\pi_j^k\} \) for all samples. The vectors detected by the same leaf-node are first grouped into one cluster, i.e. one cluster for one leaf-node. We de-
note the cluster for the \( j \)-th leaf-node as \( \Omega_j^k \). Then we perform the spectral clustering algorithm with the Euclidean distance on vectors of all leaf-nodes of or-node \( U_j \)’s children for all positive samples, and the similar vectors are grouped together. We re-arrange the feature vectors for all samples based on the newly generated partition. For example, if the feature vector \( \pi_j^k \) is grouped from \( \Omega_j^k \) into another cluster \( \Omega_j^{k'} \), we adjust the position of \( \pi_j^k \) in \( \Phi(X_k, y_k, L_k^\ast) \), i.e. by moving the feature bins into the position representing \( j \)-th leaf-node. If \( \Omega_j^{k'} \) is a newly generated cluster, we thus create a new leaf-node accordingly. By analogy, we remove one leaf-node if few samples are grouped into the corresponding cluster. In this way, the structure of \( U_j \) is reconfigured with the feature vector re-
arrangement. In practice, we constrain the extent of structural re-
configuration, i.e. only few leaf-nodes can be created or removed in one iteration. We present a toy example in Fig.5 for illustration. In Fig.5(a), a leaf-node associated with the or-node \( U_j \) is created to better handle the intra-class variance. A leaf-node is removed if there is another similar one, (e.g. the leaf-node associated with the
After clustering

**Figure 5:** Illustration of discriminative structural learning. We reconfigure the model structure by re-arranging the feature vector, as the example illustrates. Parts of the STAOG model reconfigured in two iterations are shown in (a), where the left one represents the original model and the other one the new model. During this step, a new leaf-node associated with \( U_7 \) and \( U_{12} \) is created and a leaf-node associated with \( U_8 \) is removed. Assume that we use 5 samples, \( X_1, \ldots, X_5 \), for the structure learning. (b) shows the feature vectors detected by the same leaf-node are first grouped into one cluster, i.e., one cluster for one leaf-node. (c) illustrates the feature rearrangement after clustering. For example, the feature vector of sample \( X_1 \) is grouped from cluster \( \Omega_9 \) into cluster \( \Omega_{12} \), we move the feature bins \( \phi_9 \) into the bins corresponding to the leaf-node \( L_6 \). Cluster \( \Omega_{14} \) is a newly generated cluster, we thus create a new leaf-node accordingly.

or-node \( U_8 \). The sub-vector of \( \phi_8 \) of sample \( X_1 \) is grouped from cluster \( \Omega_8 \) to cluster \( \Omega_{13} \), then the feature bins are moved from \( \phi_8 \) to \( \phi_{13} \) as Fig.5(b) shows.

After this structure reconfiguration, we obtain the new feature vector for each sample, \( \Phi(X_k, y_k, L_k) \), and the hyperplane is re-calculated as \( q_k^* = -C \sum_{h=1}^{N} \Phi(X_k, y_k, L_k) \), accordingly.

(III) The newly generated model structure can be represented by the feature vector \( \Phi(X_k, y_k, L_k) \). The model parameters can be learned by solving \( \psi_k = \arg \min \psi (f(\psi + \psi, q_k^*)) \). The optimization task in Eq.15 becomes,

\[
\min \psi \frac{1}{2} \| \psi \|^2 + C \sum_{k=1}^{N} \max_{y \in L} \left( \psi \cdot \Phi(X_k, y, L) + h(y_k, y) \right) - \psi \cdot \Phi(X_k, y_k, L_k) \quad (17)
\]

This is a standard structural SVM problem, which can be solved in the cutting plane method and Sequential Minimal Optimization. The energy Eq.17 can be calculated by \( E(\psi_k^*) = f(\psi_k^*) - g(\psi_k^*) \).

We accept the new model structure until \( E(\psi_k^*) < E(\psi_t) \) and \( \psi_{t+1} = \psi_k^* \). Otherwise, we keep the model structure as in the previous iteration and the parameter vector is calculated by \( \psi_{t+1} = \arg \min \psi_f (f(\psi + \psi, q_i)) \). Thus, we ensure that the optimization function will decrease in each iteration. We repeat the 3-step iteration until convergence. Algorithm 2 summarizes the overall algorithm of learning a STAOG model. In the case of multi-class classification, we use a one-against-rest approach and select the class with the highest score.

**6. EXPERIMENTS**

We test our STAOG model on two different action recognition databases: UCF YouTube [21] and Olympics Sports [24]. The video resolution is normalized to \( 320 \times 240 \). The YouTube dataset contains 11 action categories, which is challenging due to large variation in camera motion, object appearance, pose/view and large intra-class variability. We follow the standard setup using leave-one-out cross validation for a predefined set of 25 folds. Average accuracy over all classes is reported as performance measure. The Olympic Sports dataset consists of 16 different sports classes that contain complex motions going beyond simple punctual or repetitive actions. The challenges of Olympic sports arise from background clutters, viewpoints and complex sequence of primitive actions. Each action is performed only by a single actor and represents a temporal sequence of primitive actions (e.g., triple-jumping, pole-vault and diving). We use the same train-test split setup and the average precision (AP) for each of the action classes as in [35].

**6.1 Implementation**

We fix the number of and-nodes (anchor frames) in the STAOG model as \( T = 3 \) for UCF YouTube dataset, and \( T = 5 \) for Olympic Sports dataset empirically. We need more anchor frames for Olympic dataset because the actions are more complex and last over more frames. The parameter \( T \) can be roughly estimated by the action temporal complexity in general. The number of spatial layout for each anchor frame is fixed as \( 2 \times 2 \), and thus \( K = 4 \) or-nodes for each anchor frame. There are at most \( m = 4 \) leaf-nodes associated to each or-node. We extract the interest points described by the HOG and HOF features by utilizing the code published in [17] beforehand. The size of one action part (within a 3-D volume) is empirically set to \( 60 \times 60 \) pixels spanning \( \rho = 15 \) frames. The dimension of the generated dictionary is set at 300 for describing action parts in each anchor frame and the global features of the anchor frames. We set the discrete temporal steps searching for anchor frames: \( \Sigma = \{ \pm 2, \pm 4, \pm 6, \pm 8, \pm 10 \} \). The convergence of our learning algorithm usually takes \( 9 \sim 10 \) iterations.

The experiments are carried out on a PC with Core I5 3.0GHZ CPU and 4GB memory. The average CPU-time used to process a video from Olympic Sports dataset is 200 seconds and 150 for a video in the UCF YouTube dataset. In particular, it takes 7 seconds for processing one frame in Olympic Sports dataset, and 4 seconds for each frame in the UCF YouTube dataset on average. The efficiency of our method is slightly diverse with the densities of the feature points in the video sequence.
Liu et al [21]. Zhang et al [44]. Ikizler-Cinbis et al [12]. Dense trajectories [36].

|                      | Liu et al [21] | Zhang et al [44] | Ikizler-Cinbis et al [12] | Dense trajectories [36] | Ours-1 | Ours-2 | Ours(full) |
|----------------------|---------------|------------------|---------------------------|------------------------|--------|--------|------------|
| b_shoot              | 53.0%         | 98.0%            | 48.5%                     | 43.0%                  | 58.4%  | 62.0%  | 77.9%      |
| bike                 | 73.0%         | 74.0%            | 75.2%                     | 91.7%                  | 82.1%  | 87.3%  | 88.6%      |
| dive                 | 81.0%         | 80.0%            | 95.0%                     | 99.0%                  | 98.4%  | 98.6%  | 98.8%      |
| golf                 | 86.0%         | 68.0%            | 95.0%                     | 97.0%                  | 95.3%  | 95.3%  | 97.4%      |
| h_ride               | 72.0%         | 65.0%            | 73.0%                     | 85.0%                  | 81.3%  | 86.0%  | 88.0%      |
| s_juggle             | 54.0%         | 67.0%            | 53.0%                     | 76.0%                  | 66.0%  | 81.6%  | 82.2%      |
| swing                | 57.0%         | 71.0%            | 66.0%                     | 88.0%                  | 85.2%  | 84.1%  | 85.4%      |
| t_swing              | 80.0%         | 68.0%            | 77.0%                     | 71.0%                  | 69.6%  | 80.0%  | 80.7%      |
| t_jump               | 79.0%         | 80.0%            | 93.0%                     | 94.0%                  | 90.2%  | 94.7%  | 95.8%      |
| v_spike              | 73.3%         | 77.0%            | 85.0%                     | 95.0%                  | 89.7%  | 90.6%  | 96.4%      |
| walk                 | 75.0%         | 54.0%            | 66.7%                     | 87.0%                  | 85.2%  | 86.2%  | 87.4%      |
| Accuracy             | 71.2%         | 72.9%            | 75.2%                     | 84.2%                  | 82.0%  | 86.0%  | 88.9%      |

Table 1: Accuracy per action class and average accuracy for all classes on the YouTube dataset [21].

Figure 6: Example inference results on three different action models e.g. basketball-layup (a), clean-and-jerk (b) and triple-jump(c) learned on the Olympic Sports dataset and each of action category includes two instances. The red boxes in each frame represent the discovered discriminative action parts. Our model successfully localizes the accurate anchor frames across the instances in the long action videos. In addition, it is noticed that the large intra-class variabilities can be captured by our model.
6.2 Results and Comparisons

Compared with the recently proposed methods on the YouTube dataset, our model outperforms the state-of-the-art: we achieve the accuracy of 88.9% in YouTube dataset, the reported results of the competing algorithms are: 71.2% in [21], 72.9% in [44], 75.2% in [12] and 84.2% in [36]. The accuracy scores for all categories are reported in Table 1. Our model outperforms on 7 out of the 11 categories which have relatively large intra-class variance or background disturbance. In the Olympic Sports dataset, we obtain better AP scores for 10 out of the 16 categories, and overall AP score 71.4%, better than the previous methods [35, 24]. The competing method proposed by Tang et al. [35] utilizes the variable-duration HMM for learning the temporal structure in the video. Our results show that our compositional model with the explicit spatial and temporal relations can achieve better performance. The detailed results are reported in Table 2.

Figure 6 illustrates the inference results on three action categories from the Olympic Sports dataset, each of which includes two instances. Our model also localizes the action parts in the anchor frames, as they are discriminative in appearance and motion. These results demonstrate well the capability of our model, since the scenarios of actions contain the very realistic challenges in video action recognition. The spatial compositions defined over the and-nodes enable us to handle pose/view variations and background disturbances. The temporal compositions are effective to localize anchor frames in videos against various motion frequency, temporal locations and video length.

For further evaluation, we conduct three empirical analysis in different model settings as follows.

(I) We simplify the temporal compositions by discarding the displacement $\Delta t$ for each anchor frame, i.e. fixing the temporal structure. We report AP scores of this model setting in the fourth column of Table 1, named as “Ours-1”. The average accuracy is $82\%$, $6.9\%$ less than our complete model.

(II) We also evaluate the benefit of spatio-temporal contextual interactions. Our model can be simplified into a tree structure by removing the interactions. The accuracies are shown in the sixth column of Table 1, named as “Ours-2”. We can observe that the spatio-temporal contextual interactions make the accuracies increase $2.9\%$ on average. The increased performance of the interactions also speaks in favour for our model, as it shows that through better associations between the anchor frames and action parts, it is possible to achieve even better accuracies.

(III) One may be interested in how the performances are improved by introducing the or-nodes in spatial compositions, which is one of the key components in the STAOG model. In this experiment, we set different maximum numbers $m$ of leaf-nodes under the or-nodes, i.e. how many leaf-nodes at most can be created for the model. We compare the results on the YouTube dataset in Fig. 7, and observe that $m = 4$ achieves the better results in general than $m = 2$. $84.2\%$. In practice, the number $m$ is not sensitive, as the exact number of leaf-nodes is decided by the clustering on data during the structural learning.

7. CONCLUSION

This paper studies a novel hierarchical model for human action recognition, in the form of a configurable Spatio-Temporal And-Or Graph. This model is shown to handle well realistic challenges in action recognition. Moreover, we consider two aspects to improve our method. First, the model can be integrated with high level semantic information to represent multi-agent complex events. Second, we plan to speed up the algorithm for large-scale processing.

8. REFERENCES

[1] M. Albanese, R. Chellappa, N. P. Cuntoor, V. Moscato, A. Picariello, V. S. Subrahmanian, and O. Udrea. A constrained probabilistic petri net framework for human activity detection in video. *IEEE Transactions on Multimedia*, 10(6):982–996, 2008.

[2] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *J. Log. Comput.*, 4(5):531–579, 1994.

[3] 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 (4)*, pages 187–200, 2012.

[4] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, pages 778–785, 2011.

[5] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *CVPR*, pages 1998–2005, 2010.

[6] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, pages 3273–3280, 2011.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*,
