Latent Hierarchical Model of Temporal Structure for Complex Activity Classification

Limin Wang, Yu Qiao, *Member, IEEE*, and Xiaoou Tang, *Fellow, IEEE*

***Abstract*— Modeling the temporal structure of sub-activities is an important yet challenging problem in complex activity classification. This paper proposes a latent hierarchical model (LHM) to describe the decomposition of complex activity into sub-activities in a hierarchical way. The LHM has a tree-structure, where each node corresponds to a video segment (sub-activity) at certain temporal scale. The starting and ending time points of each sub-activity are represented by two latent variables, which are automatically determined during the inference process. We formulate the training problem of the LHM in a latent kernelized SVM framework and develop an efficient cascade inference method to speed up classification. The advantages of our methods come from: 1) LHM models the complex activity with a deep structure, which is decomposed into sub-activities in a coarse-to-fine manner and 2) the starting and ending time points of each segment are adaptively determined to deal with the temporal displacement and duration variation of sub-activity. We conduct experiments on three datasets: 1) the KTH; 2) the Hollywood2; and 3) the Olympic Sports. The experimental results show the effectiveness of the LHM in complex activity classification. With dense features, our LHM achieves the state-of-the-art performance on the Hollywood2 dataset and the Olympic Sports dataset.**

***Index Terms*— Activity classification, hierarchical model, deep structure, latent learning, cascade inference.**

I. Introduction

HUMAN activity classification is an important yet difficult problem in computer vision [1]–[3], whose aim is to determine what people are doing given an observed video. It has wide applications in video surveillance [4], [5], human-computer interface [6], sports video analysis [7], and content based video retrieval [8]. The challenges of activity classification come from many aspects. Firstly, there always exist large intra-class appearance and motion variations within the same activity class. Background clutter, illumination and viewpoint changes, and activity speed variations also increase the complexity and difficulty of classification. Secondly, compared with still image, activity video has a higher dimension. The high dimensionality of video increases not only computational cost but also difficulty to develop robust classification algorithm. Finally, human activity always consists of a sequence of sub-activities. Each sub-activity further includes gestures and motions of different body parts.

While activity exhibits complex temporal structure, its sequential decomposition yields an important cue for activity recognition. Complex activity usually is composed of several phases (see Fig. 1). Each phase corresponds to a relatively simple sub-activity, and there exists a temporal order among these phases. The importance of *temporal structure* in activity classification has been demonstrated in previous works [9]–[15]. However, the effective modeling of temporal structure is still challenging due to the following two problems.

The first problem is that "sub-activity" usually has no precise definition given a complex activity type. Sub-activity is a relatively "simple" part of a "complex" activity. Its definition depends on the temporal scale we are considering, which can be ambiguous. For example (see Fig. 1), high-jump in a long temporal scale can be divided into three sub-activities, namely running, jumping, and landing. However, in a finer temporal scale, running can be further decomposed into several primitive sub-activities, such as waiting, starting running, and speeding up. The decomposition of complex activity corresponds to a coarse-to-fine process.

The second problem is how to automatically decompose complex activity into several sub-activities given a specific video. It is a difficult problem because the sub-activities usually have various durations and temporal displacements due to the speed variations of motion. For instance, in the activity of basketball-layup, some may have a long running time before they layup the basketball, while others may have a short running time. Therefore, classification algorithm needs to automatically determine the starting and ending time points of each sub-activity.

In order to address both of the problems effectively, we propose a Latent Hierarchical Model (LHM) for complex activity recognition. LHM makes use of its tree structure to decompose activity into sub-activities automatically, and allows us to deal with the ambiguity of sub-activity. Nodes at the high layer correspond to the activities in a long temporal scale. Each activity is divided into several sub-activities at the next layer with a relatively shorter temporal scale.

Manuscript received February 22, 2013; revised July 17, 2013 and October 2, 2013; accepted December 3, 2013. Date of publication December 20, 2013; date of current version January 9, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61002042, in part by the Shenzhen Basic Research Program under Grants JC201005270350A, JCYJ20120903092050890, and JCYJ20120617114614438, in part by the 100 Talents Programme of Chinese Academy of Sciences, and in part by the Guangdong Innovative Research Team Program under Grant 201001D0104648280. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Andrea Cavallaro. *(Corresponding Author: Y. Qiao.)*

L. Wang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: 07wanglimin@gmail.com).

Y. Qiao is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 100190, China (e-mail: yu.qiao@siat.ac.cn).

X. Tang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: xtang@ie.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2013.2295753

Fig. 1. Sub-activity decomposition is related to temporal scale. High Jump can be divided into running, jumping, and landing from a long temporal scale. However, running is further composed of waiting, start-running, and speeding up, if it is observed in a short temporal scale.

The decomposition repeats recursively until it reaches leaf nodes. For each video segment, we use Bag of Visual Words (BoVW) representation, for its simplicity and compactness, to summarize motion and appearance features. Besides, the locations of all sub-activities are specified by latent variables. The latent variables are adapted to different videos, which makes our model flexible and effective to deal with the duration variations and temporal displacements of sub-activities. We formulate the learning and inference problem of LHM in the latent SVM framework [16]. Since LHM has a deeper structure with more latent variables, it is infeasible to traverse all possible configurations of sub-activities during classification process. We develop a cascade inference algorithm based on dynamic programming and prune techniques, which greatly reduces the computational cost.

The main contributions of this paper can be summarized as follows:

- We propose a latent hierarchical model (LHM), which describes the temporal structure of activity in a coarse-to-fine manner. It introduces two latent variables to denote the starting and ending time points of each sub-activity. Thus, LHM is flexible in dealing with duration variation and temporal displacement (Section III).
- We formulate the learning problem of LHM in the latent SVM framework, and we extend the traditional linear latent SVM by introducing non-linear kernels. Therefore, we can use χ^2 kernel for BoVW representation, which plays an important role in final recognition performance (Section IV-A).
- Due to a lot of possible configurations for latent variables, we develop a cascade inference algorithm to improve classification efficiency based on dynamic programming and pruning techniques (Section IV-B).
- We conduct experiments on the challenging Hollywood2 and Olympic Sports Datasets, and achieve recognition performance superior or comparable to that of the state-of-the-art approaches. Our experimental results also exhibit the effectiveness of hierarchical structure and latent variables (Section V).

II. Related Work

Human activity classification has been studied extensively in recent years. In this paper, complex activities refer to those with long temporal structures such as Sports actions [12], Cooking Composite actions [17], and so on. Here we only overview a few related works and readers can refer to [1]–[3] for good surveys.

Video Representation. Video representation has been a central issue of activity recognition. Low-level local features turn out to be effective in action recognition [18]. In recent years, researchers have developed many different spatiotemporal detectors for video, such as 3D-Harris [19], 3D-Hessian [20], Cuboids [21], and Dense [22]. Then, a local 3D-region is extracted around the interested points and a histogram descriptor is computed to capture the appearance and motion information. There were some typical descriptors such as Histogram of Gradient and Histogram of Flow (HOG/HOF) [23], Histogram of Motion Boundary (MBH) [22], 3D Histogram of Gradient (HOG3D) [24], Extended SURF (ESURF) [20], Co-occurrence descriptor [25], and so on. Finally, a global representation is obtained for each video clip via a statistical model.

Among these statistical models, Bag of Visual Words (BoVW) is a common choice in action recognition [26]. Based on local features, BoVW construction usually is composed of two steps: (i) encoding of the local features, (ii) feature pooling and normalization. There were a large body of researches on the encoding methods such as Vector Quantization (VQ) [27], Soft-assignment Encoding (SA) [28], Fisher Vector (FV) [29], Sparse Coding (SPC) [30], Locality-constrained Linear Encoding (LLC) [31], and so on. These methods focus on minimizing information loss and improving encoding efficiency. For pooling method, there were usually two typical methods, sum pooling [27] and max pooling [30], and for normalization method, typical choices include ℓ_1-normalization, ℓ_2-normalization, and power-normalization [29].

In addition to these low-level local features and BoVW representation, there were some research works on mid-level and high-level representations such as motionlet [32], motion atom and phrase [15], action bank [33], and so on.

Temporal Structure. The importance of temporal structures in recognizing human activity has been studied in previous researches [9]– [15] and [34]. Probabilistic graphical models were usually adopted to model the temporal structure of human activity or motion trajectories, such as Hidden

Markov Models (HMMs) [5], [9], Hidden Conditional Random Fields (HCRFs) [10], [11], and Dynamic Bayesian Networks (DBNs) [4], [34]. The learning and inference of graphical model were usually conducted by some approximate methods such as Expectation Maximum, Variational Methods, and Sampling Methods [35]. The learning process is complex and usually needs a large mount of data to avoid overfitting. In addition to graphical models, some research works resorted to Max-Margin Methods [12], [14]. They formulated the learning problem using Latent SVM [16], which has been shown to be effective in object detection. These methods maked use of Latent SVM to estimate the model parameters and conduct inference. The learning of LHM is formulated in the same latent SVM framework with these methods. But our model focuses on decomposing complex activity into sub-activities in a hierarchical manner. From our experimental results, the hierarchical structure plays an important role to improve the recognition performance.

Hierarchical Model. Hierarchical tree-structured model is biologically inspired by the brain architecture and vision system [36], [37]. It has been widely used in computer vision and achieved successes on various tasks, such as learning feature hierarchies [38], [39], object detection [16], [40], [41], human body parsing [42], image parsing [43], and video understanding [44]. Our model is partially inspired by the work of [40] in which Zhu *et al.* developed a hierarchical model with deep structure for object detection. In their method, an object was represented by a mixture of hierarchical tree models whose nodes represent object parts. The experimental results indicated that deep structures can convey rich descriptions of shape and appearance features. Similarly, we model human activity in a tree-structured manner and the root corresponds to the whole activity, while the other nodes represent sub-activities at different temporal scales. We find that the deep structure yields much better results than a single-layer one in our activity classification experiments, which agrees with [40]'s conclusion on object detection.

III. Latent Hierarchical Model for Activity Classification

In this section, we firstly develop a Latent Hierarchical Model (LHM) to describe the temporal structure of activity video in a coarse-to-fine manner in Section III-A. Then, we summarize the key properties of LHM in Section III-B.

A. Latent Hierarchical Model

Latent Hierarchical Model (LHM) is a tree-structured model to capture the hierarchical decomposition of complex activity into sub-activities. As shown in Fig. 2, LHM can be seen as a tree decomposition of complex activity and each node represents a video segment (activity or sub-activity) at certain temporal scale. The root node describes the whole activity (e.g. long jump) in a rough manner. The root node is divided into several sub-activities in the next layer (e.g. run, jump, land). Each sub-activity can be further decomposed recursively until leaf node, which represents the atomic activity (e.g. start run, speed up, jump up, rolling). In essence, LHM is a generalization of STAR model [45] with the independence assumption that child nodes are independently placed in a coordinate system determined by their parent node. This generalization provides more descriptive capacity to LHM and yet allows for efficient inference algorithms due to the independence assumption.

The parameters to describe the structure of LHM include the depth of tree d and the number of nodes in each layer $\{n_1, \ldots, n_d\}$. In the example of Fig. 2, the depth is set to 3 and each non-leaf node has 3 children. In principle, the structure is flexible and can be set to any others. In default we adopt the $1-3-9$ structure and we will explore other structures in experiments. LHM enables us to divide each video into N segments in different temporal scales and each segment S_i is specified by a pair $z_i = (s_i, e_i)$, where s_i is the starting time point of segment and e_i is the ending time point of segment in video V. In practice, $\mathbf{s} = \{s_i\}$ and $\mathbf{e} = \{e_i\}$ are called *latent variables* because they are not specified in the training set, and we denote $\mathbf{h} = \{\mathbf{s}, \mathbf{e}\}$.

For activity classification, we define a discriminant function of LHM for each video V given the configuration of latent variables $\mathbf{h}$:

$$f(V, \mathbf{h}) = \sum_{i=1}^{N} \Phi_i(V, z_i) + \sum_{(i,j)\in E} \Psi_{i,j}(z_i, z_j), \qquad (1)$$

where $\Phi_i(V, z_i)$ is the localized *segment model*, measuring the compatibility between video feature and segment model; E denotes a set of pairs of parent and child node; $\Psi_{i,j}(z_i, z_j)$ is the *temporal deformation model*, incorporating the structural constraints between the parent and child segments. We would like to maximize the discriminant function over all possible configurations of latent variables for each video V, then our model can find the best location for each segment:

$$f^*(V) = \max_{\mathbf{h}\in\mathbb{H}(V)} f(V, \mathbf{h}), \qquad (2)$$

where $\mathbb{H}(V)$ denotes the set of all possible configurations for latent variables $\mathbf{h}$ in video V.

Segment Model. We denote $\phi(V, z_i)$ as a feature representation extracted from segment z_i of video V. Then we can linearly parameterize the segment model as $\Phi_i(V, z_i) = \omega_i \cdot \phi(V, z_i)$. [1] In this way, each segment model acts like a linear classifier. Due to the popularization of local low-level features and bag of visual words (BoVW) representation [26], we make use of them as our features. Specifically, we use the spatiotemporal interest points (STIPs) [19] with HOG/HOF descriptors [23]. Then, we choose the vector quantization encoding and sum pooling to construct BoVW representation. Besides, in the further exploration part of Section V, we also use Dense Trajectories [22] as low-level features of LHM due to their good performance. We observe that using the dense features enables us to further boost the recognition performance of LHM.

Temporal Deformation Model. We denote $(ds_i, de_i) = (s_i, e_i) - ((s_j, e_j) + v_i)$ as the temporal displacement of a child

[1] Note that we can incorporate non-linearity by kernel tricks and the details can be found in Section IV-A.

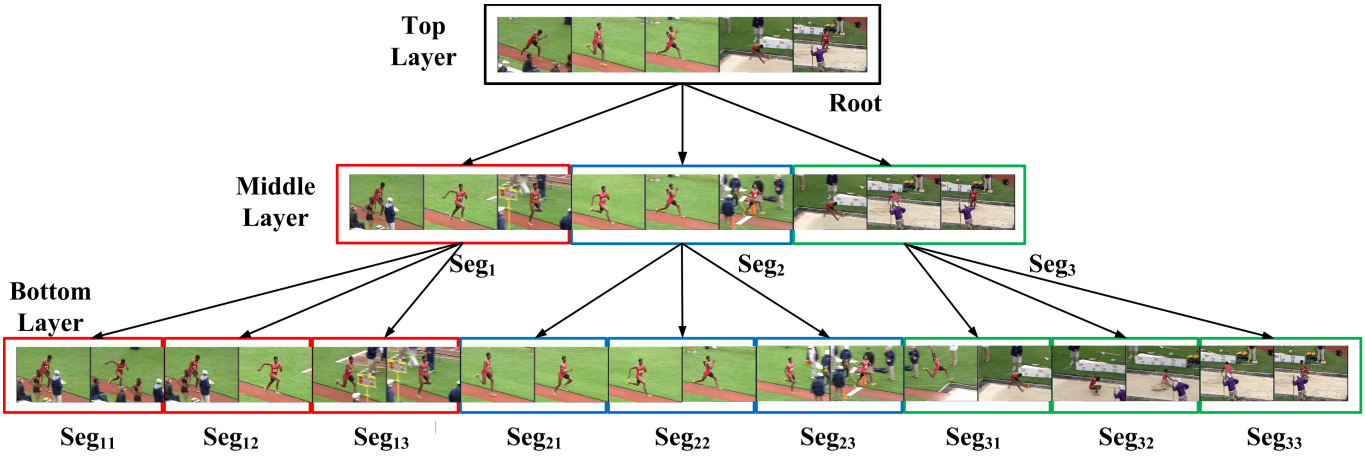


Fig. 2. An example of latent hierarchical model for activity video. In this example, LHM has a tree structure with three layers. The top layers has only one node (i.e. Root) and the middle layer has three nodes (i.e. $\text{Seg}_i, i \in \{1,2,3\}$). There are in total nine nodes (i.e. $\text{Seg}_{ij}, i, j \in \{1,2,3\}$) at the bottom layer. Nodes of different layers correspond to sub-activities in different temporal scale. Note that, we choose $1-3-9$ structures in this example and we can also resort to other structures for LHM in practice.

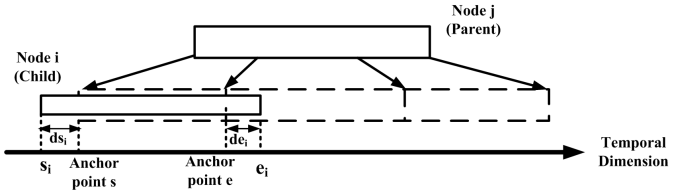


Fig. 3. Illustration of the temporal displacement between child node and the anchor point determined by its parent node.

node relative to its anchor point determined by parent node (see Fig. 3). Then we can define the temporal deformation model as $\Psi_{i,j}(z_i, z_j) = \omega_{i,j} \cdot \psi_{i,j}(z_i, z_j)$ with notation $\psi_{i,j}(z_i, z_j) = (ds_i, ds_i^2, de_i, de_i^2)$. This can be interpreted as a flexible term which allows the child node shift from its anchor point and will give penalty to large deformation. In fact, this term can be interpreted as a Gaussian distribution of child node relative to its anchor point:

$$P(z_i|z_j + v_i) = \mathcal{N}(z_i; \mu_i + (z_j + v_i), \Sigma_i), \tag{3}$$

where covariance Σ_i is set to a diagonal matrix,

$$\Sigma_i = \begin{bmatrix} \sigma_{i,s}^2 & 0 \\ 0 & \sigma_{i,e}^2 \end{bmatrix}. \tag{4}$$

Then for the log probability $P(z_i|z_j + v_i)$, we get

$$\begin{aligned} &\log|\Sigma|^{-\frac{1}{2}} - \frac{(z_i - (z_j + v_i) - \mu_i)^\top \Sigma^{-1} (z_i - (z_j + v_i) - \mu_i)}{2} \\ &= -\frac{1}{2}\left[\frac{(ds_i - \mu_{i,s})^2}{\sigma_{i,s}^2} + \frac{(de_i - \mu_{i,e})^2}{\sigma_{i,e}^2}\right] - \log(\sigma_{i,s}\sigma_{i,e}) \\ &= -\left[\frac{\mu_{i,s}}{\sigma_{i,s}^2}, \frac{1}{2\sigma_{i,s}^2}, \frac{\mu_{i,e}}{\sigma_{i,e}^2}, \frac{1}{2\sigma_{i,e}^2}\right] \cdot (ds_i, ds_i^2, de_i, de_i^2) + \text{const.} \end{aligned} \tag{5}$$

Thus this can provide a probabilistic explanation for our temporal deformation model.

B. Model Properties

LHM considers the hierarchical decomposition of complex activity into sub-activities in a recursive manner. There are several key properties about LHM which can be summarized as follows:

- *Hierarchical Structure.* LHM is a hierarchical model and has a deep structure. It can provide more descriptive power for complex activities and capture activity temporal structure in a coarse-to-fine way. In root, we provide a global BoVW to describe the whole activity roughly. In the next several layers, we focus on modeling the sub-activities in a finer manner. In addition to rich descriptive power, hierarchical structure can prune many unreasonable structures and allow us to design an efficient cascade inference algorithm, which will be discussed in Section IV-B.
- *Temporal Structure.* In addition to hierarchical structure, LHM also models the temporal structure among different sub-activities. Each sub-activity occurs at different temporal location in the whole activity and there exits an order among them. LHM exhibits temporal constraints among sub-activities, and provides rich information for complex activity recognition.
- *Flexibility.* LHM introduces two latent variables to indicate the starting and ending time points of sub-activity for each video. The latent model not only reduces the human annotation work during training period, but also increases the flexibility of our approach. During the inference phase, our model is capable of searching for a best match for each sub-activity and thus, the temporal location is adaptive to each specific video. Our model is very effective in dealing with the intra-class variation and is able to align the location of each sub-activity automatically.
- *Independence on Low-level Representation.* LHM is a general model concentrating on modeling the hierarchical structure and temporal structure of complex activity based on latent variable. LHM does not depend on

specific video representation. In experiment, we resort to bag of visual words (BoVW) representation of local spatial temporal features. Currently, we firstly use 3D Harris detector and HOG/HOF descriptor [23] for fair comparison with other methods. Then, we explore dense trajectory features [22] to boost the recognition performance of LHM. In addition to BoVW representations, we can also use other mid-level and high-level features such as Motionlet [32], Motion Atom and Phrase [15], and Action Bank [33]. Furthermore, some detection and tracking techniques can be incorporated into LHM to help determine the spatial location of activity. These extensions are out the scope of this paper.

IV. Latent Learning and Cascade Inference of LHM

In this section, we investigate how to learn the model parameters from a set of weakly labeled training samples (i.e. each training sample is only with a category label, without the detailed annotation of each sub-activity), and formulate the learning problem in a latent kernelized SVM framework in Section IV-A. Then we consider the inference problem of how to determine the locations of all sub-activities for each given video in Section IV-B. We design a cascade inference algorithm to search for the best match for each sub-activity given a video. Finally, we provide the implementation details of learning and inference algorithm in Section IV-C.

A. Latent Learning

The *learning task* is to estimate the model parameters in Equation (1) from a set of training videos $\mathbb{V} = \{V_m, y_m\}_{m=1}^{M}$, where $y_m \in \{+1, -1\}$ is the class label. We formulate the learning problem in a Max-Margin manner:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{m=1}^{M}\xi_m \\ \text{s.t.}\quad &f^*(V_m) \geq 1 - \xi_m, \text{if } y_m = 1, \\ &f^*(V_m) \leq -1 + \xi_m, \text{if } y_m = -1, \\ &\xi_m \geq 0,\ \forall\, m \in \{1, 2, \cdots, M\}, \end{aligned} \tag{6}$$

where C is a hyper parameter to balance between regularization term and loss term, $\|\cdot\|$ denotes the ℓ_2 norm, $f^*(V_m)$ is the maximum of discriminant function in Equation (1):

$$\begin{aligned} f^*(V_m) &= \max_{\mathbf{h}\in\mathbb{H}(V_m)} \sum_{i=1}^{N}\Phi_i(V_m, z_i) + \sum_{(i,j)\in E}\Psi_{i,j}(z_i, z_j) \\ &= \max_{\mathbf{h}\in\mathbb{H}(V_m)} \sum_{i=1}^{N}\omega_i\cdot\phi_i(V_m, z_i) + \sum_{(i,j)\in E}\omega_{i,j}\,\psi_{i,j}(z_i, z_j) \\ &= \max_{\mathbf{h}\in\mathbb{H}(V_m)} \mathbf{w}\cdot\Upsilon(V_m, \mathbf{h}), \end{aligned} \tag{7}$$

where $\mathbf{w}$ and $\Upsilon(V_m, \mathbf{h})$ are the concatenation of model parameters and video features:

$$\begin{aligned} \mathbf{w} &= (\omega_1, \ldots, \omega_N, \ldots, \omega_{i,j}, \ldots) \\ \Upsilon(V_m, \mathbf{h}) &= (\phi_1(V_m, z_1), \ldots, \phi_N(V_m, z_N), \\ &\qquad \ldots, \psi_{i,j}(z_i, z_j), \ldots). \end{aligned} \tag{8}$$

During training process, each training sample V_m just have class label y_m. Unlike traditional SVM [46], the problem (Equation (6)) is not convex since $f^*(V_m)$ contains an maximum operation over $\mathbf{h}$, which is called Latent SVM in [16]. It can be shown that the problem will become convex for the model parameters $\mathbf{w}$ when latent variables $\mathbf{h}$ are fixed. Thus, this allows us to develop an iterative learning algorithm between estimating latent variables $\mathbf{h}$ and optimizing model parameters $\mathbf{w}$ alternatively. In practice, we optimize the learning problem in a "coordinate decent" approach:

- *Step 1.* we initialize the model parameter $\mathbf{w}$ by a simple method, which will be discussed in Section IV-C.
- *Step 2.* we estimate latent variables for each training video:
  - For each positive example V_m, we estimate $\mathbf{h}_m^* = \arg\max_{\mathbf{h}\in\mathbb{H}(V_m)} f(V_m, \mathbf{h})$.
  - For each negative example V_m, we try to find all $\mathbf{h}'_m$ with $f(V_m, \mathbf{h}'_m) \geq -1$.
- *Step 3.* we solve the standard SVM problem when fixing the latent variables of all training samples based on the estimation of Step 2.

We firstly initialize the our model parameter with a simple scheme in *Step 1* (Details can be found in Section IV-C). Then we estimate the latent variables of training samples given the model parameters estimated in *Step 2*. A latent SVM is semi-convex in the sense that the training problem becomes convex if we fix the latent variables of positive training samples [16]. Thus we try to find the latent variables to maximize the score function for each positive training samples. The constraint of negative training samples is convex because $f^*(V_m)$ is the maximum of a set of convex functions. In principle, we can consider all possible latent variables for each negative training sample and put them in the constraints of Equation (6). In practice, however, when training a model for certain class, we often have a large number of negative training samples, for each of which we have many possible configurations of latent variables $\mathbf{h}$. Thus we cannot afford to put all possible configurations of negative samples into the learning problem. We choose to mine the hard negative instances $\mathbf{h}'_m$ with $f(V_m, \mathbf{h}'_m) \geq -1$. How to efficiently determine the locations of latent variables for each training sample can be found in Section IV-B. Finally, we solve the standard SVM problem when fixing the latent variables of each training sample.

Note that there are many optimization algorithms to solve the convex problems in *Step 3*. In [16], the author develops an algorithm of stochastic gradient descent to solve prime problem. This algorithm is efficient but can not deal with non-liner kernels. Although there are a large number of works on kernel extension for traditional SVM [46], few works have been done for latent SVM. Here, we propose to solve the dual problem of *Step 3* in order to incorporate non-linear kernel into latent SVM framework. Specifically, based on the estimated latent variables, we transform the learning problem (Equation (6)) into the following form:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{m=1}^{M}\xi_m$$

$$\begin{aligned}\text{s.t.}\quad & f(V_m, \mathbf{h}_m^*) \geq 1-\xi_m, \text{if } y_m = 1,\\ & f(V_m, \mathbf{h}'_m) \leq -1+\xi_m,\ \forall \mathbf{h}'_m \in \mathbf{H}'_m \text{ if } y_m = -1,\\ & \xi_m \geq 0,\ \forall m \in \{1,2,\ldots,M\},\end{aligned} \tag{9}$$

where $\mathbf{h}_m^*$ is the latent variable maximizing the score of positive sample V_m, and $\mathbf{H}'_m = \{\mathbf{h}'_m | f(V_m, \mathbf{h}'_m) \geq -1\}$ denotes a set of hard negative instances from sample V_m. By using Lagragian function, we can get its dual form:

$$\begin{aligned}&\max_{\alpha} \sum_{m:y_m=1} \alpha_{m,\mathbf{h}_m^*} + \sum_{m:y_m=-1} \sum_{\mathbf{h}'_m \in \mathbf{H}'_m} \alpha_{m,\mathbf{h}'_m}\\ &-\frac{1}{2}\left[\sum_{m:y_m=1} \alpha_{m,\mathbf{h}_m^*} y_m \Upsilon(V_m, \mathbf{h}_m^*) + \right.\\ &\left.\sum_{m:y_m=-1} \sum_{\mathbf{h}'_m \in \mathbf{H}'_m} \alpha_{m,\mathbf{h}'_m} y_m \Upsilon(V_m, \mathbf{h}'_m)\right]^2,\\ &\text{s.t. } 0 \leq \alpha_{m,\mathbf{h}_m^*} \leq C, \text{ if } y_m = 1,\\ &\sum_{\mathbf{h}'_m \in \mathbf{H}'_m} \alpha_{m,\mathbf{h}'_m} \leq C,\ \alpha_{m,\mathbf{h}'_m} \geq 0, \text{ if } y_m = -1,\end{aligned} \tag{10}$$

where α are the dual variables and their relationship with $\mathbf{w}$ is determined by:

$$\begin{aligned}\mathbf{w} = &\sum_{m:y_m=1} \alpha_{m,\mathbf{h}_m^*} y_m \Upsilon(V_m, \mathbf{h}_m^*)\\ &+ \sum_{m:y_m=-1} \sum_{\mathbf{h}'_m \in \mathbf{H}'_m} \alpha_{m,\mathbf{h}'_m} y_m \Upsilon(V_m, \mathbf{h}'_m).\end{aligned} \tag{11}$$

In the dual problem (10), we can replace the dot product $\Upsilon(V_m, \mathbf{h}_m) \cdot \Upsilon(V_n, \mathbf{h}_n)$ with non-linear kernel $\mathcal{K}(\Upsilon(V_m, \mathbf{h}_m), \Upsilon(V_n, \mathbf{h}_n))$. In practice, we use linear kernel for temporal placement model and χ^2 kernel for BoVW representation, defined as follows:

$$\mathcal{K}_{\chi^2}(S_1, S_2) = \exp\left\{-\frac{1}{2S}\sum_{r=1}^{D} \frac{(S_{1,r} - S_{2,r})^2}{S_{1,r} + S_{2,r}}\right\}, \tag{12}$$

where S denotes the mean distance among training samples, $S_{1,r}$ denotes the $r-$th element of histogram S_1 and D is the dimension of BoVW histogram. Then, the kernel for two training instances is defined as:

$$\begin{aligned}\mathcal{K}(\Upsilon(V_m, \mathbf{h}_m), \Upsilon(V_n, \mathbf{h}_n)) = &\sum_{i=1}^{N} \mathcal{K}_{\chi^2}(\phi_i(V_m, z_i^m), \phi_i(V_n, z_i^n))\\ &+ \sum_{(i,j)\in E} \psi_{i,j}(z_i^m, z_j^m) \cdot \psi_{i,j}(z_i^n, z_j^n).\end{aligned} \tag{13}$$

Note that, due to non-linear kernel, we can not calculate $\mathbf{w}$ explicitly and the calculations of $\mathbf{w} \cdot \Upsilon(V_k, \mathbf{h}_k)$ are replaced by the following formula:

$$\begin{aligned}\mathbf{w} \cdot \Upsilon(V_k, \mathbf{h}_k) = &\sum_{m:y_m=1} \alpha_{m,\mathbf{h}_m^*} y_m \mathcal{K}(\Upsilon(V_m, \mathbf{h}_m^*), \Upsilon(V_k, \mathbf{h}_k))\\ &+ \sum_{m:y_m=-1} \sum_{\mathbf{h}'_m \in \mathbf{H}'_m} \alpha_{m,\mathbf{h}'_m} y_m \mathcal{K}(\Upsilon(V_m, \mathbf{h}'_m), \Upsilon(V_k, \mathbf{h}_k)).\end{aligned} \tag{14}$$

Algorithm 1: Cascade Inference of LHM.

Data: Testing examples:V, Learned Model $\mathbf{w}$ and Thresholds $\{t_i, t'_i\}$.
Result: Maximum of score.
forall *possible location* z_i *of root* n_0 **do**
  $F[n_0, z_i] \leftarrow$ `ComputeMax(`n_0,z_i,V,$\{t_i, t'_i\}$`)`.
- Return maximum of score: `MAX(`$F[n_0, z]$`)`.

Function `ComputeMax(`n_i,z_i,V,$\{t_i, t'_i\}$`)`.
if *X has no child* **then**
  return $\Phi_i(V, z_i)$.
else
  foreach *child node* n_j *of* n_i **do**
    foreach *possible location* z_j *of* n_j **do**
      //Deformation pruning
      if $\Psi_{j,i}(z_j, z_i) \leq t'_i$ **then**
        skip z_j .
      else
        $F[n_j, z_j] \leftarrow \Psi_{i,j}(z_i, z_j)$.
        $q \leftarrow$ `ComputeMax (`n_j,z_j,V,$\{t_i, t'_i\}$`)`.
        $F[n_j, z_j] \leftarrow F[n_j, z_j] + q$.
    $s[n_j] \leftarrow$ `MAX (`$F[n_j, z]$`)`.
    //Hypothesis pruning
    if $s[n_j] \leq t_i$ **then**
      return $-\infty$.
    else
      $total \leftarrow total + s[n_j]$.
  $total \leftarrow total + \Phi_i(V, z_i)$.
  return $total$.

B. Cascade Inference

The *inference task* of LHM is to predict class label y and latent variables $\mathbf{h}$ given the video V and model parameters $\mathbf{w}$. The main challenge comes from the fact that the number of possible configurations for latent variables $\mathbf{h}$ is large, which prevents us from using brute force approach to calculate the discriminant function over all possible $\mathbf{h}$. In [12], Niebles *et al.* used the dynamic programming and distance transform techniques in a similar fashion to [16]. They claim that this matching scheme is efficient once the appearance similarities between the video sequences and each motion segment classifiers are computed. However, in our problem, evaluating the appearance similarities is the bottleneck due to χ^2 kernel calculation. Besides, our LHM is a deep structure model and introduces two latent variables for each segment, thus it is very time-consuming to calculate the appearance similarities of all possible configurations in advance. Inspired by the method of cascade object detection in [47], we design a cascade inference algorithm for LHM. The core idea of our algorithm is to make use of dynamic programming and prune techniques to constrain the search space of $\mathbf{h}$ and accelerate the process of inference.

First, we convert the inference problem into the following subproblem using dynamic programming techniques. For a node n_i at location z_i specified by starting and ending time point pair (s_i, e_i), its largest discriminant value $F(n_i, z_i)$ can be calculated by the following recursive function:

$$F(n_i, z_i) = \sum_{(j,i)\in E} \max_{z_j}\{F(n_j, z_j) + \Psi_{j,i}(z_j, z_i)\} + \Phi_i(V, z_i). \tag{15}$$

Then, we evaluate the score of each node in a depth-first-search (DFS) order. The cascade inference algorithm for a tree-structured model with $n+1$ nodes has $2n$ intermediate thresholds for two kinds of pruning techniques. As shown in Algorithm 1, during the DFS process, we use two kinds of pruning techniques, namely *deformation pruning* and *hypothesis pruning*:

- Deformation pruning: we will skip the segment specified by z_j if the temporal deformation term $\Psi_{j,i}(z_j, z_i)$ is smaller than a threshold t'_j. Intuitively, the total score will decrease greatly if it is plus the temporal deformation term. This pruning technique enables us to constrain child node to move in a reasonable interval.
- Hypothesis pruning: if the score maximum of a child node $s[n_j]$ is less than a threshold t_j, then we will prune its parent node n_i at location z_i. Intuitively, if the parent node n_i is located correctly, then the maximum of its child node score would not be smaller than a threshold. So, the small score of its child node may indicate the location of parent node is not correct.

During the DFS process, once we evaluate the response of node n_i at location z_i, we will store its value to avoid calculating it again. Using the cascade inference algorithm, we can find the maximum of score for each video V efficiently. Besides, during the inference process, we can keep the location of each node, thus we can find the best configuration of latent variables $\mathbf{h}$ effectively.

C. Implementation Details

Initialization. Unlike the heuristic initialization of [12], we propose a simple method to initialize our model structure and training samples. We set the anchor point of child node relative to parent node in a regular grid layout. For training samples, we initialize latent variable $\mathbf{h}$ according to the model structure i.e. $ds = 0$, $de = 0$. Then we get a set of instances for the first round of standard SVM training.

Updating Latent Variables. During the step to estimate latent variables $\mathbf{h}$, the duration of root node is restricted to cover at least 80% of the whole video. The positions of the child nodes are ensured to overlap with the corresponding reference box. These restrictions can suppress some unreasonable structures and improve search efficiency.

Thresholds of Cascade Inference. During training process, we search all possible configurations for latent variables $\mathbf{h}$ without using prune techniques. For each node, we keep the minimum score of its child node over all positive samples. The hypothesis pruning thresholds t_j will be the minima multiplied by a ratio β_1 ($\beta_1 = 0.5$ in experiments). We also store the values of temporal deformation term for different parent and child node pairs. The deformation pruning thresholds t'_j is set to be the minima of the deformation term multiplied by a ratio β_2 ($\beta_2 = 1.3$ in experiments) over positive training samples. Note that the deformation term is usually negative.

V. Experiments

We firstly conduct experiments on three public action datasets: the KTH [48], the Hollywood2 [49], and the Olympic Sports Dataset [12]. Then we further explore some important aspects of LHM. For the three datasets, we use LIBSVM package [50] to solve the standard SVM problem in the learning framework of Section IV-A. For multi-class classification, we apply the one-vs-all training scheme.

A. KTH Dataset

The KTH is a relatively simple dataset among the three and it contains 6 action classes: boxing, hand-clapping, hand-waving, jogging, running, and walking [48]. [2] Each action is performed by 25 actors in four controlled environments: outdoors, outdoors with scale variation, outdoors with different cloths, and indoors. There is no camera motion in these videos and the intra-class variations are relatively small compared with other datasets. Some video frames and their detected STIPs are shown in Fig. 4. We follow the experimental settings described in [48] and the codebook size is 1,000.

Experimental results are shown in Fig. 5 and Table I. From the results, we see that our method can achieve high accuracy rates for the actions of boxing, hand-waving, hand-clapping and walking. But for the action of running and jogging, the performance of our method decreases because the two actions are similar to each other and there is a strong confusion between these two kinds of action.

Comparison with Other Methods. We compare LHM with three other methods in Table I. The method of [48] is based on spatiotemporal jets at the center of each detected interest point using normalized derivatives, and use BoVW representation and SVM classifier. The other two methods [23], [23] are both based on HOG/HOF features. The method of [23] uses the traditional BoVW and the method of [12] uses a single-layer segment model. From the comparison, we find the three methods using HOG/HOF features obtain similar performance, which are much better than spatiotemporal jets. LHM is comparable to other methods using local features. The actions in KTH are relatively simple, and the detected local spatial-temporal features provide sufficient information for activity recognition.

B. Hollywood2 Dataset

The Hollywood2 action dataset [49] is collected from 69 different Hollywood movies. [3] In total, there are 1,707 action samples, which is composed on 823 training samples and 884 testing samples. The authors provide the clean and noisy versions of the dataset and we use the clean version. There are 12 action classes: answer-phone, drive-car, eat, fight-person, get-out-car, hand-shake, hug-person, kiss, run, sit-down, sit-up, and stand-up. Some video frames and their detected STIPs are shown in Fig. 4. As all the video clips are segmented from movies, the video quality is very high and there is no camera shaking. The performance is evaluated by average precision according to paper [49] and the codebook size is set as 4,000.

The final recognition results are shown in Table II. We see that the Hollywood2 dataset is more difficult than the KTH

[2] Available at http://www.nada.kth.se/cvap/actions/

[3] Available at http://www.di.ens.fr/ laptev/actions/hollywood2/

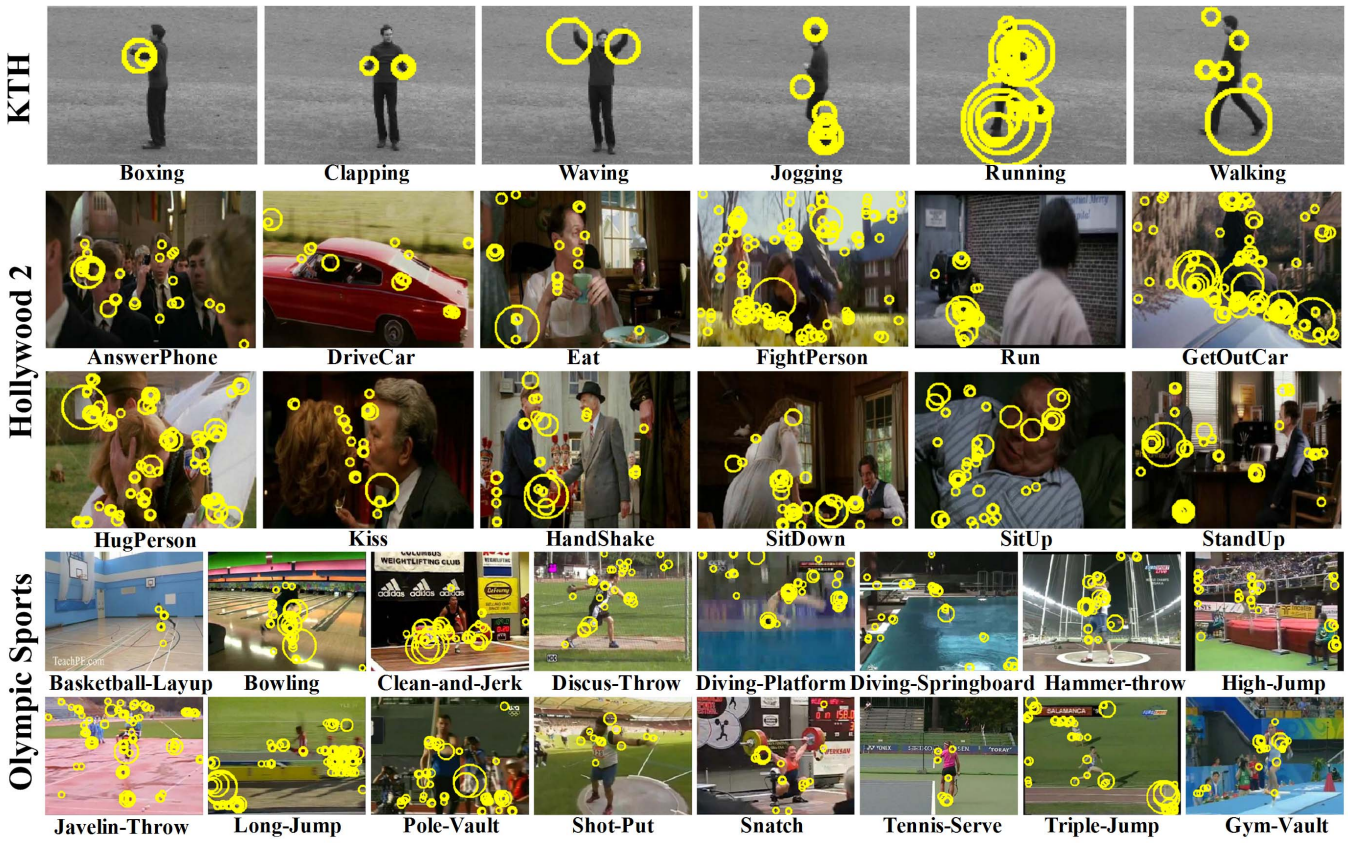


Fig. 4. Sample frames from the three datasets: KTH, Hollywood2, and Olympic Sports. The detected spatio-temporal interest points (STIPs) are drawn on the frames by yellow circles.

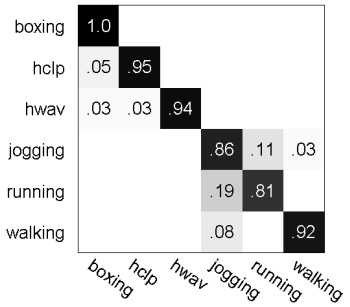


Fig. 5. The confusion matrix of LHM method in KTH dataset.

dataset and our method obtains average precision as 48.1%. For action classes such as drive-car and fight-person, LHM can perform relatively well and get average precision larger than 70%. However, for the rest of action class, the recognition rate is relatively low. The videos are all extracted from realistic movies and the intra-class variance is very large compared with the KTH dataset.

Comparison with Other Methods. We compare our method with three other methods: the BoVW model [19] (baseline), the context model [49], and Convolutional Gated RBM (GRBM) [51]. The BoVW model uses the same features and codebook size and the context model exploits the static scenes as a cue for action recognition. The Convolutional Gated RBM aims to learn the features directly from the video intensity with some deep models. The BoVW is implemented by ourselves and use the same codebook with LHM. We find the result is similar to a recent empirical study of local features [18]. From the comparison, we observe that our method outperforms the other two methods in 8 action classes. For mean of average precision, our method achieves higher rate than traditional BoVW by 2.8% and than GRBM by 1.5%.

C. Olympic Sports Dataset

The Olympic Sports Dataset is collected by [12] and has 16 sports classes: basketball-layup, bowling, clean-and-jerk, discus-throw, diving-platform, diving-springboard, hammer-throw, high-jump, javelin-throw, long-jump, pole-vault, shot-put, snatch, tennis-serve, triple-jump, and gym-vault. All the videos are from YouTube and each activity class contains a complex temporal structure compared with the activities in the KTH and Hollywood2 dataset. Note that the authors only release part of their dataset on their website. [4] There are 649 sequences for training and 134 sequences for testing. We conduct experiments according to the settings released on their website. In order to compare our method with those proposed by [12] and [14], we use the same feature representation and the codebook size is 1,000. The final performance is evaluated by computing the average precision (AP) for each of the action classes and reporting the mean AP over all the class (mAP).

[4] Available at http://vision.stanford.edu/Datasets/OlympicSports/

TABLE I

COMPARISON OF LHM METHOD WITH OTHER ACTIVITY CLASSIFICATION APPROACHES EVALUATED USING CLASSIFICATION ACCURACY ON THE KTH DATASET. "OVERALL" INDICATES THE AVERAGE ACCURACY OVER ALL THE CLASSES. THE BOLD FONTS INDICATES THE BEST PERFORMANCES

Class	Schuldt [48]	Laptev [23]	Niebles [12]	LHM
boxing	97.9%	97.0%	99.2%	**100.0%**
clapping	59.7%	95.0%	**96.5%**	95.0%
waving	73.6%	91.0%	**99.9%**	94.0%
jogging	60.4%	**89.0%**	78.2%	86.0%
running	54.9%	80.0%	79.5%	**81.0%**
walking	83.8%	**99.0%**	94.4%	92.0%
overall	71.7%	**91.8%**	91.3%	91.4%

TABLE II

COMPARISON OF LHM METHOD WITH OTHER ACTIVITY CLASSIFICATION APPROACHES EVALUATED USING THE AVERAGE PRECISION (AP) ON THE HOLLYWOOD2 DATASET. "OVERALL" INDICATES THE MEAN AVERAGE PRECISION (MAP) OVER ALL THE CLASSES. THE BOLD FONTS INDICATES THE BEST PERFORMANCES

Class	Marszalek [49]	Laptev [23]	Taylor [51]	LHM
answer-phone	10.7%	12.5%	-	**21.5%**
drive-car	75.0%	**81.4%**	-	79.8%
eat	28.6%	57.2%	-	**62.8%**
fight-person	57.1%	70.5%	-	**72.7%**
get-out-car	11.6%	27.0%	-	**39.1%**
hand-shake	14.1%	**21.1%**	-	19.2%
hug-person	13.8%	37.3%	-	**38.1%**
kiss	**55.6%**	54.4%	-	52.1%
run	56.5%	**66.1%**	-	65.4%
sit-down	27.8%	52.5%	-	**53.4%**
sit-up	7.8%	15.1%	-	**19.3%**
stand-up	32.5%	48.2%	-	**53.3%**
overall	32.6%	45.3%	46.6%	**48.1%**

Our experiment results are shown in Table III and Fig. 8. From the results we see that our method obtains a relatively high performance in the Olympic Sports Dataset with mAP = 69.2%. For some activity categories such as basketball-layup, diving-springboard, diving-platform, gym-vault, our method performs pretty well and achieves average precisions larger than 90%. See Fig. 8, our model can automatically divide gym-vault into three sub-activities: running, rolling in the air, and landing; long-jump into three sub-activities: starting running, speeding up, and jumping; clean-and-jerk into three sub-activities: beginning, clean phase, and overhead jerk phase. The duration of each sub-activity varies and adapts to each activity video. Each sub-activity is further decomposed into more primitive sub-activities in the bottom layer.

However, for some activity categories such as tennis-serve, high-jump, triple-jump, and discus-throw, our method performs poorly and the average precision is low. We analyze the reasons as follows. Firstly, we find there exist strong confusions among some activity categories. For example, the similarity among triple-jump, long-jump, and high-jump is very high. The three activities share some sub-activities such as running and jumping. For activities such as hammer throw, discus-throw, and shot-put, the whole processes of activities

TABLE III

COMPARISON OF OUR METHOD WITH OTHER ACTIVITY CLASSIFICATION APPROACHES EVALUATED USING THE AVERAGE PRECISION (AP) ON THE OLYMPIC SPORTS DATASET. "OVERALL" INDICATES THE MEAN AVERAGE PRECISION (MAP) OVER ALL THE CLASSES. THE BOLD FONTS INDICATES THE BEST PERFORMANCES

Class	Laptev [23]	Niebles [12]	Tang [14]	LHM
basketball-layup	69.9%	82.1%	85.5%	**93.5%**
bowling	37.9%	53.0%	**64.3%**	57.5%
clean-jerk	71.8%	70.6%	78.2%	**80.5%**
discus-throw	40.3%	47.3%	**48.9%**	38.7%
diving-platform	86.9%	95.4%	93.7%	**96.0%**
diving-springboard	71.2%	84.3%	79.3%	**91.3%**
hammer-throw	63.1%	71.2%	70.5%	**86.9%**
high-jump	26.1%	27.0%	18.4%	**32.0%**
javelin-throw	**90.0%**	85.0%	79.5%	85.5%
long-jump	79.4%	71.7%	81.8%	**85.8%**
pole-vault	68.8%	**90.8%**	84.9%	84.5%
shot-put	**52.2%**	37.3%	43.3%	47.4%
snatch	52.2%	54.2%	**88.6%**	58.3%
tennis-serve	27.0%	33.4%	49.6%	**51.0%**
triple-jump	8.7%	10.1%	16.1%	**25.7%**
gym-vault	84.9%	86.1%	85.7%	**91.4%**
overall	58.2%	62.5%	66.8%	**69.2%**

are almost the same, namely firstly moving in rhythm and then delivering. Secondly, for some activities such as tennis-serve with short duration, their temporal structures are not as complex as the other. Thus, our model structure may be a bit complex than the activity class.

Comparison with Other Methods. We compare our LHM with three other methods: the BoVW model (baseline) [23] and two kinds temporal models [12], [14] in Table III. The method of [12] models the temporal structure of decomposable motion segments and formulates the problem in a similar framework. The model of [14] is based on the variable-duration hidden Markov model and it gets the state-of-the-art performance with local features in this dataset.

From the comparison, the proposed LHM achieves higher average precision for 10 of the 16 activity classes. For mean average precision, our LHM is higher than the baseline by 11% and than the state-of-the-art by 2.4%. These results exhibit that hierarchical decomposition of sub-activities and automatic adaptation of starting and ending time points is effective for complex activity classification.

D. Further Explorations

Our LHM provides a general framework for hierarchical modeling the temporal structure of complex activity. In this section, we study the different aspects of LHM in a more detailed way. Firstly, we explore the different structure settings and their influences on final recognition performance. Secondly, we investigate the effectiveness of latent variables by comparing the recognition performance of LHM with temporal pyramids [52]. Temporal pyramids decompose each video into segments of equal duration, while LHM automatically aligns video by efficient search in latent variable space. Then, we investigate the inference efficiency of the proposed cascade algorithm. Finally, we incorporate denser and richer

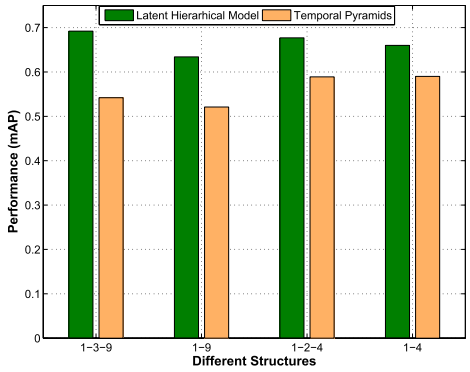


Fig. 6. Explorations of the performance for different hierarchical structures and comparison of Latent Hierarchical Model and Temporal Pyramids on the Olympic Sports Dataset.

local features based on dense trajectories [22] into LHM to boost final recognition performance.

Hierarchical Model is Better. In order to explore the performance of LHM with respect to model structure, we conduct additional experiments on the Olympic Sports Dataset. We choose three other structure settings, $1-2-4$, $1-4$, $1-9$, and the results are shown in Fig. 6. From the results, we see that the structure $1-3-9$ obtains the best performance (69.2%) and the second one is $1-2-4$ (67.7%). The deep structures are better than the shallow ones: $1-9$ (63.4%) and $1-4$ (66.0%). We conclude that deep structure is useful for the complex activity classification. LHM models the decomposition of complex activity into sub-activities in a coarse-to-fine manner. The deep structure provides extra descriptive power to LHM and contributes for more accurate alignment of different video samples. The comparison results indicate that hierarchical model is better for activities with complex and long temporal structure.

Latent Model is Better. We also implement the temporal pyramid representations on the Olympic Sports Dataset. For different structures, we compare the recognition performance of LHM and temporal pyramids [52], which uses fixed temporal segmentation, and the results are depicted in Fig. 6. From the experimental results, we observe that LHM performs much better than temporal pyramids: $1-3-9$ (69.2% vs. 54.2%), $1-9$ (63.4% vs. 52.1%), $1-2-4$ (67.7% vs. 58.9%), $1-4$ (66.0% vs. 59.0%). All these results indicate that model with latent variables, which are determined adaptively for different videos, can describe the complex activity more effectively. Besides, we observe that the recognition rates of temporal pyramids representation are similar to or even lower than those of the traditional BoVW method. It implies that if there exist strong temporal displacements among different videos, the temporal pyramids representation may harm the final performance. This observation can be ascribed to the fact that its assumption of approximate temporal correspondence in the temporal pyramid may not hold for the training and testing samples.

Efficiency of Cascade Inference. We explore the efficiency of cascade inference. For 300-frame length video, the number of segments needed to be calculated for inference with cascade and without cascade for $1-3-9$ structure is shown in Fig. 7. For top layer, we need to calculate the same number of segment response for inference both with and without cascade in top layer. However this number is relatively small due to we have the 80% overlap constraints. For the second and the third, cascade inference with algorithm only needs to calculate much less segment response. Totally, the number of calculated segment response for usual inference algorithm is 15 times of cascade inference. We implement LHM in matlab and run on a PC with E5645 CPU(2.4GHZ) and 8G RAM. We test 100 videos randomly and the average time for each video is 6s for cascade inference and 70s for usual bottom-up inference without pruning techniques. Our cascade inference algorithm can improve the time efficiency about 10 times without influencing the recognition performance.

TABLE IV
RESULTS OF LHM WITH DENSE TRAJECTORY ON THE OLYMPIC SPORTS DATASET AND THE HOLLYWOOD2 DATASET. WE COMPARE OUR RESULTS WITH THAT OF THE STATE-OF-THE-ART APPROACH [53]. THE BOLD FONTS INDICATES THE BEST PERFORMANCE

Method	Hollywood2	Olympic Sports
BoVW+STIPs+HOG/HOF [23]	45.3%	58.2%
BoVW+Dense Trajectory [53]	58.2%	74.1%
BoVW+Dense Trajectory+STP [53]	**59.9%**	77.2%
LHM+Dense Trajectory	**59.9%**	**83.2%**

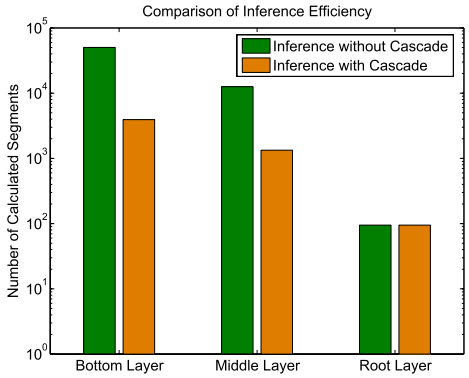


Fig. 7. Comparison of efficiency between inference with cascade and without cascade.

Dense Features are Better. Spatiotemporal interest points (STIPs) [19] with HOG/HOF descriptor [23] is a common choice for local features. However, Wang *et al.* propose a much denser and richer feature called dense trajectory [22], which turns out to be effective in capturing the motion and appearance information for human activity recognition. From the experiment results shown in Table IV, BoVW with dense trajectory features obtains much better results than with STIPs features.

In this part, we explore incorporating dense features into LHM, which combines the richness of low level features with the descriptive and flexible power of LHM to further boost recognition performance. In experiment, we use four kinds of descriptors: HOG, HOF, MBHX, MBHY, and the codebook size is 4,000. We obtain recognition performance of 59.9% for the Hollywood2 Dataset and 83.2% for the Olympic Sports

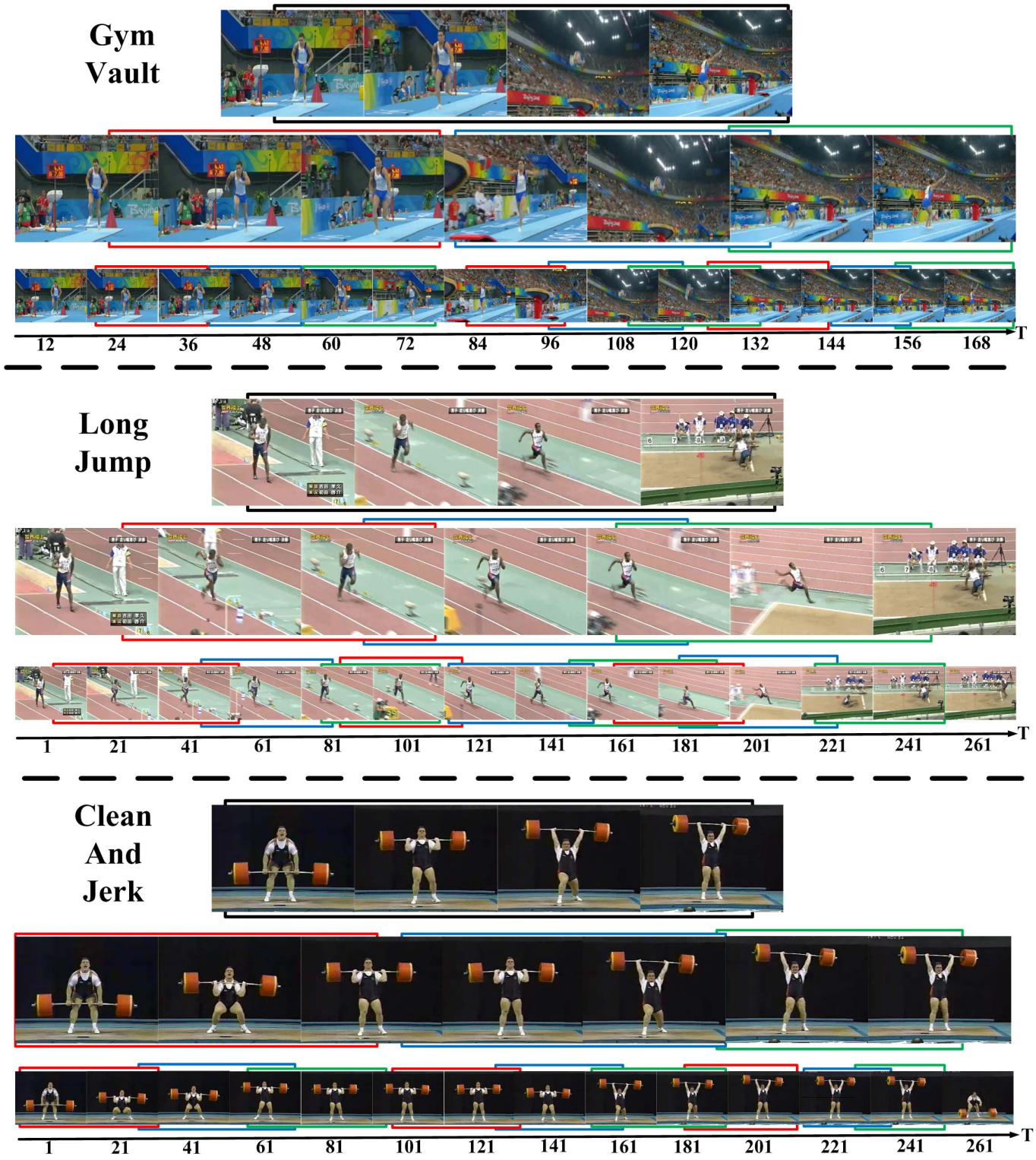


Fig. 8. Three examples of LHM's learning results on the Olympic Sports Dataset. The numbers denote the indexes of frames in the video. Each video is decomposed into sub-activities in a $1-3-9$ structure. In each layer, video is divided into several segments, whose durations are determined in inference. Each segment correspond to a learnt sub-activity, where the color lines indicate the durations of sub-activities. From the result, we see that our LHM can automatically decompose complex activity into several sub-activities hierarchically. Each complex activity video is represented as a whole segment in root node and it is divided into several sub-activities in the middle layer. Each sub-activity is further decomposed into more primitive actions in the bottom layer.

Dataset. For the Hollywood2 Dataset, the action types are relative simple and there are no complex temporal structures in them. Thus, there is only slight improvement for LHM compared with BoVW. However, for the Olympic Sports Dataset, the advantage of LHM is more evident and LHM obtains considerable performance improvement. Currently,

Wang *et al.* [53] further improve their recognition performance by incorporating structural information into BoVW framework with spatiotemporal pyramids (STP) and obtain the best results on the two datasets. Our LHM with dense trajectory features are comparable to the best results on the Hollywood Dataset and much better than the best results on the Olympic Sports Dataset, even though we don't consider any spatial information in our model. In conclusion, dense features are more rich and effective than STIPs. With dense features, we can further boost the recognition performance of LHM and obtain the state-of-the-art results on the challenging Hollywood2 Dataset and Olympic Sports Dataset.

VI. Conclusion

This paper has proposed a Latent Hierarchical Model (LHM) for classifying complex activities. LHM is a hierarchical model with deep structure, which decomposes activity into sub-activities in a coarse-to-fine manner. We develop the latent learning algorithm to estimate the parameters of LHM. We also present a cascade inference algorithm to improve activity classification efficiency. The starting and ending time points of each sub-activity indicated by latent variables, are determined automatically in inference process. LHM is flexible and effective to deal with the duration variation and temporal displacement of each sub-activity. The experimental results show that the proposed method with dense features achieves recognition performance superior or comparable to that of the previous methods on two challenging action datasets: the Hollywood2 and the Olympic Sports. In particular, LHM is more suitable for activities with longer and more complex temporal structure and gains considerable recognition performance improvement.

References

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, p. 16, 2011.

[2] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[3] D. A. Forsyth, O. Arikan, L. Ikemoto, J. F. O'Brien, and D. Ramanan, "Computational studies of human motion: Part 1, tracking and motion synthesis," *Found. Trends Comput. Graph. Vis.*, vol. 1, nos. 2–3, pp. 77–254, 2005.

[4] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, "Trajectory classification using switched dynamical hidden Markov models," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1338–1348, May 2010.

[5] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden Markov models," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912–1919, Jul. 2007.

[6] A. Jaimes and N. Sebe, "Multimodal human-computer interaction: A survey," *Comput. Vis. Image Understand.*, vol. 108, nos. 1–2, pp. 116–134, 2007.

[7] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 796–807, Jul. 2003.

[8] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Found. Trends Inf. Retr.*, vol. 2, no. 4, pp. 215–322, 2009.

[9] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[10] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2006, pp. 1521–1527.

[11] Y. Wang and G. Mori, "Hidden part models for human action recognition: Probabilistic versus max margin," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1310–1323, Jul. 2011.

[12] J. C. Niebles, C.-W. Chen, and F.-F. Li, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proc. ECCV*, 2010, pp. 392–405.

[13] J. Wang, Z. Chen, and Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3185–3192.

[14] K. Tang, F.-F. Li, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 1250–1257.

[15] L. Wang, Y. Qiao, and X. Tang, "Mining motion atoms and phrases for complex action recognition," in *Proc. IEEE Conf. ICCV*, Dec. 2013, pp. 2680–2687.

[16] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[17] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, "Script data for attribute-based recognition of composite activities," in *Proc. 12th ECCV*, 2012, pp. 144–157.

[18] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2009, pp. 1–11.

[19] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[20] G. Willems, T. Tuytelaars, and L. J. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. ECCV*, 2008, pp. 650–663.

[21] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. 2nd Joint IEEE Int. Workshop VS-PETS*, Oct. 2005, pp. 65–72.

[22] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3169–3176.

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[24] A. Kläser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. BMVC*, 2008, pp. 1–10.

[25] X. Peng, Y. Qiao, Q. Peng, and X. Qi, "Exploring motion boundary based sampling and spatial-temporal context descriptors for action recognition," in *Proc. BMVC*, 2013, pp. 1–11.

[26] X. Wang, L. Wang, and Y. Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition," in *Proc. ACCV*, 2012, pp. 572–585.

[27] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop Statist. Learn. Comput. Vis.*, vol. 1. 2004, pp. 1–22.

[28] J. V. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *Proc. ECCV*, 2008, pp. 696–709.

[29] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek, "Image classification with the fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.

[30] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1794–1801.

[31] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3360–3367.

[32] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2674–2681.

[33] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 1234–1241.

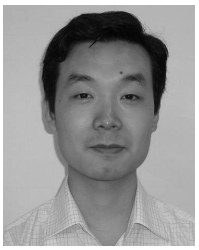
[34] B. Laxton, J. Lim, and D. J. Kriegman, "Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[35] C. Bishop, *Pattern Recognition and Machine Learning*, vol. 4. New York, NY, USA: Springer-Verlag, 2006.

[36] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio, "A quantitative theory of immediate visual recognition," *Progr. Brain Res.*, vol. 165, pp. 33–56, 2007.

[37] N. Krüger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. H. Piater, *et al.*, "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1847–1871, Aug. 2013.
[38] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[39] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2046–2053.
[40] L. Zhu, Y. Chen, A. L. Yuille, and W. T. Freeman, "Latent hierarchical structural learning for object detection," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 1062–1069.
[41] Y. Chen, L. Zhu, and A. L. Yuille, "Active mask hierarchies for object detection," in *Proc. ECCV*, 2010, pp. 43–56.
[42] L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. L. Yuille, "Max margin AND/OR graph learning for parsing the human body," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.
[43] S. C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends Comput. Graph. Vis.*, vol. 2, no. 4, pp. 259–362, 2006.
[44] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 2012–2019.
[45] B. Leibe, A. Leonardis, and B. Schiele, "An implicit shape model for combined object categorization and segmentation," in *Toward Category-Level Object Recognition*. New York, NY, USA: Springer-Verlag, 2006, pp. 508–524.
[46] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
[47] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester, "Cascade object detection with deformable part models," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2241–2248.
[48] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th ICPR*, Aug. 2004, pp. 32–36.
[49] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 2929–2936.
[50] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
[51] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. ECCV*, 2010, pp. 140–153.
[52] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 2847–2854.
[53] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.

Limin Wang received the B.Sc. degree in computer science and technology from Nanjing University, Nanjing, China, in 2011. He is currently pursuing the Ph.D. degree with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include computer vision and machine learning.



Yu Qiao (S'05–M'07) received the Ph.D. degree from the University of Electro-Communications, Japan, in 2006. He was a JSPS Fellow and Project Assistant Professor with the University of Tokyo from 2007 to 2010. He is currently a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include pattern recognition, computer vision, multimedia, image processing, and machine learning. He has published more than 90 papers. He received the Lu Jiaxi Young Researcher Award from the Chinese Academy of Sciences in 2012.

Xiaoou Tang (S'93–M'96–SM'02–F'09) received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, the M.S. degree from the University of Rochester, Rochester, NY, USA, in 1991, and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1996.

He is a Professor with the Department of Information Engineering and an Associate Dean (Research) of the Faculty of Engineering, Chinese University of Hong Kong. He was the Group Manager with the Visual Computing Group, Microsoft Research Asia, from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing.

Dr. Tang received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition in 2009. He is a Program Chair of the IEEE International Conference on Computer Vision in 2009, and an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*.