# Action Recognition with Unsupervised Hierarchical Dynamic Parsing and Encoding

Bing Su, *Member, IEEE,* Jiahuan Zhou, *Student Member, IEEE,* Xiaoqing Ding, *Fellow, IEEE,* and Ying Wu, *Senior Member, IEEE*

***Abstract*—Generally the evolution of an action is not uniform across the video, but exhibits quite complex rhythms and non-stationary dynamics. To model such non-uniform temporal dynamics, in this paper we describe a novel hierarchical dynamic parsing and encoding method to capture both the locally smooth dynamics and globally drastic dynamic changes. It parses the dynamics of an action into different layers and encodes such multi-layer temporal information into a joint representation for action recognition. At the first layer, the action sequence is parsed in an unsupervised manner into several smooth-changing stages corresponding to different key poses or temporal structures by temporal clustering. The dynamics within each stage are encoded by mean-pooling or rank-pooling. At the second layer, the temporal information of the ordered dynamics extracted from the previous layer is encoded again by rank-pooling to form the overall representation. Extensive experiments on a gesture action dataset (Chalearn Gesture) and two generic action datasets (Olympic Sports and Hollywood2) have demonstrated the effectiveness of the proposed method.**

***Index Terms*—Action Recognition, Temporal Clustering, Hierarchical Modeling, Dynamic Encoding.**

## I. Introduction

ACTION recognition is a fundamental yet challenging problem in computer vision and has become an essential component in a wide range of applications. Although many efforts has been dedicated to this area over decades and significant progresses have been made, action recognition still remains largely unsolved due to inherent difficulties such as high-dimensional video data, background dynamic clutters, fast irregular motion, large intra-class variations, partial occlusion and viewpoint changes.

The performance of action recognition methods depends heavily on the representation of video data. For this reason, many recent efforts focus on developing various action representations in different levels. The state-of-the-art action representation is based on the Bag-of-Visual-Words (BoW) [1] framework, which includes three steps: local descriptors extraction, codebook learning, and descriptors pooling or feature encoding. The raw local descriptors themselves are noisy and the discriminative power of the distributed BoW representation comes from the efficient coding of these local descriptors. As a result, the temporal dependencies and dynamics of the video are seriously neglected.

Dynamics characterize the inherent global temporal dependencies of actions. Existing dynamic-based approaches generally view the video as a sequence of observations and model it with temporal models. The models can either be state-space-based such as Hidden Markov Model (HMM) [2] and Conditional Random Field (CRF) [3] or exemplar-based such as Dynamic Time Warping (DTW) [4]. Such models generally not only require a large amount of training data to exactly estimate parameters, statistics and temporal alignments, but also cannot directly lead to vector representations with a fixed dimension. Recently, Fernando et al. [5] propose to pool frame-wide features via learning to rank within the BoW framework, which encodes the temporal evolution of appearances in a principled manner and results in a representation with the same dimension of the frame-wide features. The dynamics are considered as the ordering relations of frame-wide features and the changes of all successive frames are treated equally.

The dynamic behind an action is time-varying and not easy to be figuratively expressed. However, for a specific given action video, the dynamic does have some intuitive rhythms or regularities. One cue is that humans can recognize an action from some ordered key frames. Typically each frame captures a key pose, and the number of key poses is much smaller than the number of frames in the whole video. Taking an example of Fig. 1, a video recording an action "jump" may contain up to hundreds of frames, but only three key poses can represent the drastic changes in the dynamics: running approach, body stay flew in the air and touch down. There may be many similar frames corresponding to each key pose. These key poses segment the whole action into different divisions or stages, and each stage consists of the frames related to a key pose. Therefore, the dynamics of an action can also be viewed as a hierarchy. The dynamics within each stage are relatively stable, and the dynamics of the sequence of the stages or key poses represent the essential evolution of the action.

In this paper, we incorporate the dynamics in both levels into a joint representation for action recognition. We build an unsupervised hierarchical structure for each action video to parse the dynamic of appearances into different levels and encode them in different layers. In the first layer, we parse the sequence of frame-wide features into different stages and encode the dynamic and appearances into a feature vector within each stage. In the second layer, we extract high-level dynamic encoding representation by rank pooling the encoded

B. Su is with the Institute of Software, Chinese Academy of Sciences, Beijing, China, 100190. E-mail: subingats@gmail.com.

X. Ding is with the Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084. E-mail: dingxq@tsinghua.edu.cn.

J. Zhou and Y. Wu are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, 60208. E-mail: {jzt011, yingwu}@eecs.northwestern.edu.

Fig. 1. The action "jump" can be roughly parsed into three divisions: running approach, body stay flew in the air and touch down. Each division can also be parsed into different sub-divisions

features produced in the first layer.

The contributions of this work include: 1) The proposed hierarchical parsing and encoding is a new unsupervised representation learning method. It hierarchically abstracts the prominent dynamic and generates a representation that is robust to speed and local variations and captures the high-level semantic information for a video. 2) We propose an unsupervised temporal clustering method to achieve efficient dynamic parsing. It built on a single action sequence and no annotations or training are needed to perform parsing. 3) The extracted representations from multi-scale parsings provide complementary discriminative information and hence can be combined.

This paper is an extension of our previous conference paper [6]. The major extensions include: 1) For video-based action video, the sequence of frame-wide features is first modeled by linear dynamic system and the state sequence is used instead of the observation sequence to obtain the parsing segments, such that the temporal clustering can be performed faster; 2) The relationships and comparisons of the proposed method with rank pooling [5] and improved dense trajectories [7] are detailed discussed; 3) The influences of parameters and the effects of temporal clustering are experimentally evaluated on more datasets; 4) The proposed method is experimentally evaluated on frame-wide features with the Fisher Vector encoding in addition to the BOW encoding; and 5) "Deeper" hierarchical model is built with more than two layer and the representations from these layers are combined.

The rest of this paper is organized as follows: Section II briefly reviews the existing work on action recognition; Section III presents the proposed hierarchical dynamic parsing and encoding method; We evaluate the proposed method in Section IV and draw conclusions in Section V.

## II. Related work

Appearance and dynamics are two important aspects of actions. Previous work on action representation and modeling can be accordingly categorized into appearance-based approaches and dynamic-based approaches.

**Appearance-based action representation approaches.** BoW representation is widely used in appearance-based action representation approaches. Different methods differ in the local visual descriptors and the coding scheme. HOG [8], [9], HOF [10] and MBH [11] are typical low-level descriptors used in video-based action recognition. These descriptors can be computed either sparsely at local space-time cuboids [12] or by dense sampling scheme [7]. HOG/HOF descriptors are extracted around STIPs in [12]. Several descriptors such as HOG and MBH are fused to encode the densely sampled trajectories in [11], and the dense trajectories are improved to correct camera motion in [7]. Various coding variants have also been proposed to encode these local descriptors, such as local soft assignment [13], sparse coding [14], locality-constrained linear coding [15], super vector coding [16], multi-view super vector [17], super sparse coding with spatial-temporal awareness [18], Fisher vector [19] and vector of locally aggregated descriptors [20], [21].

Efforts have also been made to construct hierarchical feature representations based on BoW to capture context information and high-level concepts. Three levels of spatial-temporal context hierarchy are modeled in ascending order of abstraction in [22]. At multiple spatial-temporal scales, the most discriminative class-specific shapes of space-time feature neighborhoods are learned in [23] and the contextual interactions between interest points are encoded to augment local features in [24]. Mid-level information are preserved by stacking two-layer nested Fisher vector encoding in [25]. Mid-level representation can also be captured by mining discriminative action parts [26], [27], [28], [29], [30].

Besides these hand-crafted features, deep neural networks have also been applied to learn representations directly from videos. In [31], convolutional features generated by deep architectures are aggregated by trajectory-constrained pooling. In [32], appearance and motion-based convolutional neural network (CNN) features are computed from all the tracks of body joints. In [33], the sequence of convolutional net-based frame-wide features for a video is mapped by multi-layer Long Short Term Memory (LSTM) networks into a fixed length representation, which is decoded by another LSTM to produce a target sequence for unsupervised training.

**Dynamic-based action modeling approaches.** Both deterministic models and generative models have been studied to model and represent dynamics and motions in action recognition. For deterministic models, the temporal structures or alignments are explicitly modeled. Dynamic time warping (DTW) is used to align action sequences for recognition in [4]. Maximum margin temporal warping is proposed in [34] to learn temporal action alignments and phantom action templates. Actom sequence model [35] and graphs [36], [37] are also used to model temporal structures and relationships

among local features. Recently deep neural architectures are employed for modeling actions. In [38], spatial and temporal nets are incorporated into a two-stream ConvNet. In [39], salient dynamics of actions are modeled by the differential recurrent neural networks.

Generative models are typically based on temporal (hidden) state-space, such as HMM [2], [40], coupled HMM [41], semi-Markov model that incorporates prior knowledge on state duration [42], CRF [3], [43], HCRF [44], dynamic Bayes nets [45], temporal AND-OR graph [46], and linear dynamic systems [47]. Hierarchical sequence summarization is achieved in [48] by recursively learning hidden spatio-temporal dynamics based on latent variables of CRFs and grouping observations with similar latent states. The hierarchical combinatorial structures of cross-view actions are represented by a compositional multi-view AND-OR model in [49] via explicitly modeling the geometry, appearance and motion variations.

**Temporal clustering.** Aligned Cluster Analysis [50] divides a sequence by minimizing the similarities among the segments, where the similarity between two segments is measured by a dynamic time alignment kernel. As the dynamics of each segment may not be stable, the segments do not correspond to stable action stages. In contrast, our method divides a sequence into segments by minimizing the within-segment variances so that the frames within each segment are similar. As each segment shows a stable dynamic, it can be viewed as a stage of an action. In MMTC [51], features in sequences are clustered into several common clusters, and a multi-class SVM is trained to assign clusters using all training sequences. Our method acts on each individual sequence independently and no training is needed, and the segments from different sequences are different and only account for the evolution of the specific sequence.

## III. Hierarchical Dynamic Parsing and Encoding

Video-wide temporal evolution modeling method proposed in [5] aggregates the frame-wise features into a functional representation via a ranking machine. This representation captures the evolution of appearances over frames and hence provides the video-wide temporal information. However, the ranking function within the learning to rank machine attempts to rank all the frames in the video and these frames are equally treated, which ignores the non-stationary evolution of dynamic within different stages and cannot directly exploit the complex hierarchical temporal structures. Hierarchical architecture has the ability to learn a higher-level sematic representation by pooling local features in the lower layer and refining the features from the lower layer to the higher layer. In this section we propose a hierarchical temporal evolution modeling method, namely *Hierarchical Dynamic Parsing and Encoding or HDPE*, to take the rhythmic of stage-varying dynamic into account. The pipeline of HDPE is shown in Fig. 2.

### A. Unsupervised temporal clustering

In order to capture the temporal structures corresponding to relatively-uniform local dynamics, we first propose an unsupervised temporal clustering method that learns the parse of an action sequence only from the sequence itself.

For each action video, we extract a feature vector from each frame. Thus the action video can be represented as a sequence of such features. We denote the video by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$, where $\mathbf{x}_t$ the feature vector extract from the $t$-th frame, and $T$ is the number of frames in the whole video. We denote the partition of $\mathbf{X}$ by a segmentation path $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_L]$, where $L$ is the number of divisions, typically $L < T$. $\mathbf{p}_t = [s_t, e_t]^T$ provides the range $\{s_t, s_t + 1, \cdots, e_t\}$ of the $t$-th division, $s_t$ and $e_t$ are the start and end indexes of the frames in this division. The number of frames divided into the $t$-th division is $l_t = e_t - s_t + 1$. We hope that each division contains a set of steady evolving frames corresponding to the same key pose or temporal structure. We require $\mathbf{P}$ being a non-overlapping and completing partition that covers the whole video. Non-overlap means no frame can be simultaneously divided into two divisions, complete means that every frame in the sequence must be divided into one and only one division, hence the elements of $\mathbf{P}$ satisfy the following constraints: $s_1 = 1, e_L = T$, $s_{t+1} = e_t + 1, \forall t = 1, \cdots, L-1$, $e_t \geq s_t, \forall t = 1, \cdots, L$. There may be noisy or outlier frame in the sequence, which is significantly different with its successive neighbor frames. To avoid assigning such outlier frame into a separate division and prevent extremely unbalance divisions, we make the restriction on the number of elements in each division. Specifically, we limit the maximum number of elements within one division by $f \cdot l_{ave}$, where $f$ is the band factor, and $l_{ave} = \frac{T}{L}$ is the average number of elements in each division by uniform segmentation.

To parse the sequence $\mathbf{X}$ into different divisions, where each stage is related to a key pose, we define an essential sequence $\mathbf{U}$ of $\mathbf{X}$ as the sequence of key poses in $\mathbf{X}$: $\mathbf{U} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_L]$, where $\boldsymbol{\mu}_j$ is the mean of frame-wise features of the frames in the $j$-th division. Once $\mathbf{U}$ is given, the partition $\mathbf{P}$ can be obtained by computing the optimal alignment path along which the sum of distances between the aligned elements in $\mathbf{X}$ and the warped $\mathbf{U}$ is minimal among all possible paths:

$$\min_{\mathbf{P}} \sum_{j=1}^{L} \sum_{i=s_j}^{e_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2 \tag{1}$$

Consider a partial path that assigning the first $i$-th elements in $\mathbf{X}$ to the first $j$-th elements in $\mathbf{U}$, and the last $l$ elements of the first $i$-th elements in $\mathbf{X}$ are assigned to the $j$-th element of $\mathbf{U}$. We denote the sum of element-wise distances along this partial path by the partial distance $d(i, j, l)$. The minimal partial distance can be determined recurrently:

$$d(i,j,l) = \begin{cases} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2, l = 1, i = j = 1 \\ \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2 + \min\limits_{k=1}^{f \cdot l_{ave}} d(i-1, j-1, k), l = 1 \\ \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2 + d(i-1, j, l-1), l \leq f \cdot l_{ave} \\ Inf, otherwise \end{cases} \tag{2}$$

Eq. (2) does not have aftereffect, hence Eq. (2) can be effectively solved by dynamic programming. When both par-

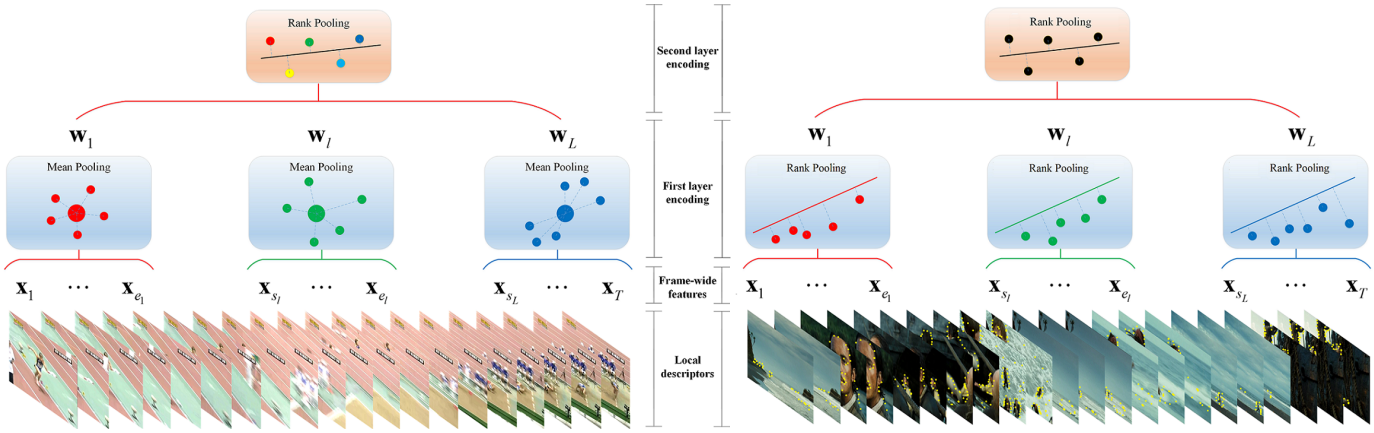



Fig. 2. The pipeline of the proposed method. The first layer can either adopt mean pooling (left) or rank pooling (right)

tial sequences reach the end, the minimal distance along the optimal path is determined by $\min_{l=1}^{f \cdot l_{ave}} d(T, L, l)$ and the optimal partition path $\mathbf{P}$ can be obtained by back tracking.

Given the partition $\mathbf{P}$ of the sequence $\mathbf{X}$, the essential sequence $\mathbf{U}$ can be obtained by computing the mean of each division. The essential sequence in turn can be used to parse the sequence $\mathbf{X}$ into different divisions. Determining the essential sequence $\mathbf{U}$ and computing the partition $\mathbf{P}$ rely on each other. We develop an unsupervised temporal clustering method to jointly mine temporal structures in the sequence $\mathbf{X}$ and learn the partition $\mathbf{P}$ that parses $\mathbf{X}$ into stages with respect to these temporal structures.

We first initialize the partition $\mathbf{P}$ to be a uniform partition that divides the sequence $\mathbf{X}$ into $L$ equal segments. For example, if $L = 3, T = 9$, i.e. we divide a sequence $\mathbf{X}$ with 9 elements into 3 segments, the initial partition $\mathbf{P} = [[1, 3]^T, [4, 6]^T, [7, 9]^T]$. Then we compute the essential sequence $\mathbf{U} = [\mu_1, \mu_2, \cdots, \mu_L]$, whose elements are the means of elements in the corresponding divisions:

$$\boldsymbol{\mu}_j = \frac{1}{l_j} \sum_{k=s_j}^{e_j} \mathbf{x}_k, j = 1, \cdots, L \tag{3}$$

After that, we update the partition $\mathbf{P}$ by aligning the elements in $\mathbf{X}$ to those in $\mathbf{U}$ to parse $\mathbf{X}$ using the dynamic programming algorithm. The essential sequence $\mathbf{U}$ is recomputed in turn with the updated $\mathbf{P}$. The two procedures are continued until the partition is unchanged with the previous iteration or a pre-fixed number of iterations is reached. We summarize the joint partition learning and temporal clustering algorithm in Alg. 1.

**Convergency.** Given $\mathbf{P}$, computing the essential sequence $\mathbf{U}$ by using Eq. (3) is equivalent to the solution of minimum mean square error problem: $\min_{\boldsymbol{\mu}} \sum_{i=s_j}^{e_j} \|\mathbf{x}_i - \boldsymbol{\mu}\|_2^2, j = 1, \cdots, L$. Given $\mathbf{U}$, computing $\mathbf{P}$ directly minimizes Eq. (1). Both procedures reduce the objective of Eq. (1). Eq. (1) has a trivial lower bound $\sum_{j=1}^{L} \sum_{i=s_j}^{e_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2 \geq 0, \forall \mathbf{P}, \mathbf{U}$. Thus the partition learning algorithm will at least converge to a local minimum.

**Algorithm 1** Unsupervised action parsing by temporal clustering

**Input:** a sequence $\mathbf{X}$, the number of divisions $L$, the maximal number of iterations $Ite$, the band factor $f$;
**Output:** the partition $\mathbf{P}$ of $\mathbf{X}$;
Initialize the partition path $\mathbf{P}$ to be a uniform partition;
**while** $\mathbf{P}$ has not converged and the number of iterations is less than $Ite$ **do**
  Compute the essential sequence $\mathbf{U}$ using (3);
  Update the partition path $\mathbf{P}$ by solving ref using the dynamic programming algorithm (2) with the band factor $f$;
**end while**

**Computational complexity.** The complexities of dynamic programming Eq. (2) and calculating Eq. (3) are $O(LNd)$ and $O(Ld)$, $L$, $N$ and $d$ are the number of segments, the length of the input sequence and the dimension of the frame-wide features. Hence the complexity of the temporal clustering Alg. 1 is $O(iLNd)$, $i$ is the number of iterations. As the method processes each sequence separately, parallel speedup can be easily performed.

### *B. State sequence extraction via linear dynamic systems*

For video-based actions, the state-of-the-art features extracted from each frame are generally very high-dimensional and contain redundant and noisy information. For example, for the improved dense trajectories-based feature [7] introduced in Sec. II, which is perhaps the most widely used hand craft features for action videos, when encoding a single type of descriptors, e.g. MBH, the dimensionality of frame-wide features is $4,000$ if the BoW encoding method is used with a codebook of $4,000$ visual words and is $2 * 0.5 * 192 * 256$ if the Fisher Vector encoding method is used with 256 number of Gaussians and a compression factor of $0.5$ for PCA. As analyzed in Sec. III-A, the computation complexity of the temporal clustering algorithm 1 is linearly proportional to the dimensionality of the frame-wide features $d$. Directly parsing the sequence of such high-dimensional features introduces a

heavy computation overhead. Moreover, the temporal clustering algorithm 1 relies on the Euclidean distance between frame-wide features, but the distance measure may be meaningless in such high-dimensional space.

A straightforward way to handle such high-dimensional data is to perform dimensionality reduction. However, supervised dimensionality reduction methods such as [40], [52] not only introduce supervision information, but also need large amounts of training samples and high space and time complexities to train the transformation. Even for the simple unsupervised PCA, a covariance matrix of huge size need to be calculated and processed to obtain the projection, the temporal dependencies of frame-wide features are totally lost and all the training sequence samples need to be available for training.

It has been shown in [47] that the motion dynamics of a feature sequence can be represented by the state trajectory sequence with only few dimensions per frame, where the state sequence can be obtained from the sequence itself by linear dynamic system (LDS). For an action video, the evolutions of its frame-wide features can be modeled by an LDS as follows:

$$\begin{cases} \mathbf{s}_{t+1} = \mathbf{A}\mathbf{s}_t + \gamma_t \\ \mathbf{x}_t = \mathbf{B}\mathbf{s}_t + \eta_t \end{cases} \tag{4}$$

where the sequence of frame-wide features $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$ is also called the observation sequence, and $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_T]$ is the latent stated sequence. $\mathbf{s}_t \in \mathbb{R}^{d'}$ is the state or latent variable corresponding to the frame-wide observation $\mathbf{x}_t \in \mathbb{R}^d$ at frame $t$. $d$ is the dimensionality of the frame-wide features, and $d'$ is the dimensionality of the frame-wide states. The state sequence is modeled as a first-order Markov process, that is, the next state $\mathbf{s}_{t+1}$ is determined by the current state $\mathbf{s}_t$, and the current observation $\mathbf{x}_t$ is determined by the current state $\mathbf{s}_t$. $\gamma_t \tilde{} N(0, \mathbf{\Sigma}_\gamma)$ and $\eta_t \tilde{} N(0, \mathbf{\Sigma}_\eta)$ are the system noise and the observation noise, respectively, which are modeled by two zero-mean i.i.d. Gaussian processes. $\mathbf{\Sigma}_\gamma$ and $\mathbf{\Sigma}_\eta$ are the co-variances of the corresponding Gaussian distributions, respectively.

Given the observation sequence $\mathbf{X} \in \mathbb{R}^{d \times T}$, the parameters in Eq. (4) and the state sequence $\mathbf{S} \in \mathbb{R}^{d' \times T}$ have closed-form least squares estimations [53]. The singular value decomposition (SVD) is first applied to $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{d \times d}$ and $\mathbf{V} \in \mathbb{R}^{T \times T}$ are orthogonal matrices, and $\mathbf{\Lambda} \in \mathbb{R}^{d \times T}$ is a rectangular diagonal matrix. The state sequence can be estimated as:

$$\mathbf{S} = \tilde{\mathbf{\Lambda}}\mathbf{V}^T \tag{5}$$

where $\tilde{\mathbf{\Lambda}} \in \mathbb{R}^{d' \times T}$ is the truncated rectangular diagonal matrix that only preserves the rows of $\mathbf{\Lambda}$ with respect to the $d'$ largest diagonal values. The other parameters can be estimated as:

$$\mathbf{B} = \mathbf{U}, \mathbf{A} = \mathbf{S}_{2:T}\mathbf{S}^+_{1:T-1}$$

where $+$ is the Moore-Penrose inverse. $\mathbf{A}$ is in fact the least squares estimation from: $\mathbf{A} = \arg\min_{\mathbf{A}} \|\mathbf{A}\mathbf{S}_{1:T-1} - \mathbf{S}_{2:T}\|_F^2$. $\mathbf{\Sigma}_\gamma$ and $\mathbf{\Sigma}_\eta$ can then be estimated from the residuals.

Even when the dimensionality $d'$ of the frame-wide states is set very small, the state sequence can still reflect the motion dynamic evolutions. In [47], only 3 dimensions are preserved. In this paper, we set $d'$ to 15 or 30 in our experiments and obtain the parsing segments by the temporal clustering algorithm 1 from the state sequence instead of the observation sequence.

### C. The first layer modeling

For an action sequence sample $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$, we first parse it into $L$ divisions using Alg. 1. We denote the parsing result of $\mathbf{X}$ by $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_L]$. The evolution within each division is relatively steady and hence the frames in each division can be equally treated. An abstract feature vector can be extracted from each division via mean pooling or rank pooling [5].

Mean pooling simply uses the mean of the frame-wide features as the output of the division. For the $l$-th division, we denote the segmentation fragment as $\mathbf{X}^{[l]} = [\mathbf{x}_{s_l}, \mathbf{x}_{s_l+1}, \cdots, \mathbf{x}_{e_l}]$. The mean pooling result of the division can be calculated as:

$$\mathbf{w}_l = \frac{1}{e_l - s_l + 1} \sum_{\tau=0}^{e_l - s_l} \mathbf{x}_{s_l+\tau}$$

Rank pooling learns a linear ranking function to order the frame-wise features in each division via learning to rank and uses the parameters of the function as the representation of the temporal structure associated with the division. A vector valued function that transforms each element $\mathbf{x}_{s_l+t}$ to the corresponding time varying mean vector $\mathbf{v}_{s_l+t} = \frac{\mathbf{u}_{s_l+t}}{\|\mathbf{u}_{s_l+t}\|}$, where $\mathbf{u}_{s_l+t} = \frac{1}{t+1} \sum_{\tau=0}^{t} \mathbf{x}_{s_l+\tau}.$, is first applied to $\mathbf{X}^{[l]}$, resulting in $\mathbf{V}^{[l]} = [\mathbf{v}_{s_l}, \mathbf{v}_{s_l+1}, \cdots, \mathbf{v}_{e_l}]$. A linear function $f(\mathbf{w}_l; \mathbf{v}) = \mathbf{w}_l^T \cdot \mathbf{v}$ is used to predict the ranking score for each $\mathbf{v}_{s_l+t}$. The parameters $\mathbf{w}_l$ of the linear function is learned to rank the orders of the elements in the division, such that $f(\mathbf{w}_l; \mathbf{v}_{s_l}) > f(\mathbf{w}_l; \mathbf{v}_{s_l+1}) > \cdots > f(\mathbf{w}_l; \mathbf{v}_{e_l})$.

$$\begin{gathered} \arg\min_{\mathbf{w}_l} \tfrac{1}{2}\|\mathbf{w}_l\|^2 + C \sum_{0 \le a < b \le e_l - s_l} \varepsilon_{ab} \\ s.t.\ \mathbf{w}_l^T \cdot (\mathbf{v}_{s_l+a} - \mathbf{v}_{s_l+b}) \ge 1 - \varepsilon_{ab}, \\ \varepsilon_{ab} \ge 0, \forall 0 \le a < b \le e_l - s_l \end{gathered} \tag{6}$$

$\mathbf{w}_l$ is used as the representation of the $l$-th temporal structure. After the first layer modeling, the original sequence $\mathbf{X}$ is mapped to the sequence of key temporal structures $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L]$, which contains high-level abstract information based on the original representation.

For simple actions and fine-grained actions, compared with the dynamic of divisions, the dynamic within each division is quite uniform and contributes little to the discrimination of the whole actions. Changing the orders of frames in a division does not influence the understanding of the action. Mean pooling is suitable for such cases, which is equivalent to extract key frames. The key frames are more robust to individual frames and local distortions since each key frame is the mean of a division. For complex activities, the dynamics in

divisions may be complex so that the orders of frames in each division cannot be changed, and hence it is better to apply rank pooling.

### D. The second layer modeling

The output sequence $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L]$ from the first layer reflects the essential temporal evolution of the sequence, which can be thought as the sequence of key poses, each pose is a pooling of the frames in the corresponding stage and captures the stage-wide temporal evolution. The second layer extracts the video-wide temporal evolution from these ordered stage-wide temporal evolutions. The learning-to-rank modeling used in each division of the first layer is applied to $\mathbf{W}$. A ranking function $f(\mathbf{y}; \mathbf{w}') = \mathbf{y}^T \cdot \mathbf{w}'$ that aims at providing the orders of the time varying mean vectors $\mathbf{w}_1', \mathbf{w}_2', \cdots, \mathbf{w}_L'$ by applying vector valued function to elements of $\mathbf{W}$ such that $f(\mathbf{y}; \mathbf{w}_l') > f(\mathbf{y}; \mathbf{w}_k'), \forall 1 \leq k < l \leq L$. The parameter vector $\mathbf{y}$ of $f(\mathbf{y}; \mathbf{w}')$ serves as the final representation of the video sequence $\mathbf{X}$.

Several advantages of the proposed HDPE method are as follows. First, the method is totally unsupervised, simple and easy to perform. The parsing, the state sequence extraction and the hierarchical encoding are all built on a single action sequence. No annotations are needed to perform parsing or encoding, and no labels or negative data are needed for training. Second, the method is robust to local distortions and individual outliers or noisy frames. The abstract feature produced by the first layer for each division is a pooling of all the frame-wide features in the division, and few outliers or distortions have little effect on the pooling result. Third, the learned representation implicitly combines local appearances and global dynamic in a principled hierarchical manner. The orders within the parsed divisions are not so important, hence the pooling of the first layer focuses on capturing the local averaged appearances. The temporal orders among the divisions are crucial and reflect the inherent dynamic of the video. The encoding of the second layer focuses on capturing such global high-level dynamic.

### E. Stacking more layers

We construct the hierarchy with two layers in this paper, and note that it can be easily generalized to more layers as shown in Fig. 3. An action video is first represented by a sequence of frame-wide features. The state sequence is extracted from the input sequence and temporal clustering is performed on the state sequence to parse the sequence into segments. The frame-wide features within each segments are encoded by mean pooling, and the encoded features of all the ordered segments form a new sequence, which serves as the input sequence of the next layer. The parsing and encoding process can be repeated for multiple layers. A vector representation can be extracted by rank pooling the encoded sequence of any layer. Either the rank pooled vector of the last layer or the combination of vectors of all the layers can by adopted as the representation of the video. The length of the encoded sequence of a layer is shorter then the input sequence of this layer. Therefore, for a video with $T$ frames, at most $T-1$ layers can be constructed.

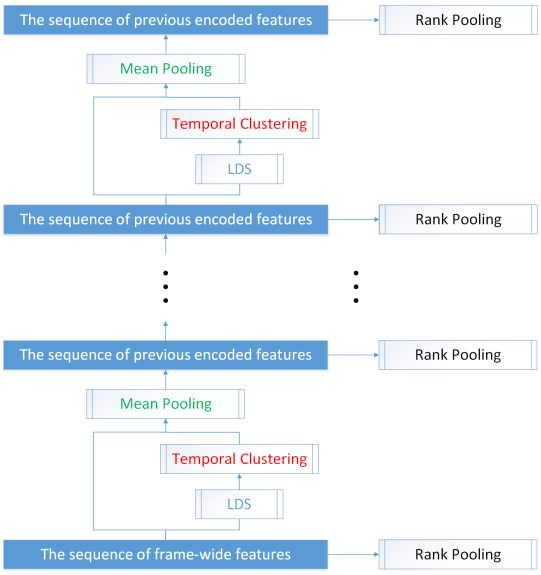



Fig. 3. The multi-layer HDPE model

## IV. Experiments

In this section we evaluate the performance of the proposed method on one gesture recognition dataset, i.e. the Chalearn gesture dataset, and two challenging generic action recognition datasets, including the Olympic Sports dataset and the Hollywood2 dataset.

### A. Datasets

**ChaLearn Gesture Recognition dataset [54].** This dataset consists of Kinect video data from 20 Italian gestures performed by 27 persons. There are 955 videos in total, and each video contains 8 to 20 non-continuous gestures with a length of 1 to 2 minutes. The overall length of the videos is about 23 hours, and the recordings and annotations include RGB, depth, foreground segmentation and Kinect skeletons. The dataset is split into training, validation and test sets. We report the multi-class (the mean over all classes) precision, recall and F-score measures on the validation set, as in [54], [5].

**Olympic Sports dataset [43].** This dataset contains 783 video sequences from 16 sports actions. The videos are collected from YouTube and annotated using Amazon Mechanical Turk. The dataset is split into training and test sets. The training set includes 649 video sequences and the test set includes the remaining 134 video sequences. We report the mean average precision over all classes (mAP) as in [7] and the accuracy as in [34] .

**Hollywood2 dataset [8].** This dataset contains RGB-video data from 12 generic action classes. There are in total 1,707 video clips in the dataset, which are collected from 69 different Hollywood movies. The dataset is split into training and test sets. The training set includes 823 videos and the test set includes the remaining 884 videos. The videos in the two sets are selected from different movies. We report mAP as in [8], [7].

### B. Experimental setup

**Frame-wide features.** For each action video, we extract a high-dimensional feature vector from each frame and represent the video by a sequence of frame-wide features. For the Olympic Sports dataset and the Hollywood2 dataset, we use the improved dense trajectories descriptors [7], which have achieved state-of-the-art results. We extract trajectory, HOG, HOF and MBH descriptors from the trajectories corresponding to a dense regular grid for all frames. We follow the same settings with [7] when extracting descriptors and the dimensionalites of the trajectory, HOG, HOF and MBH descriptors are 30, 96, 108 and 192, respectively. The square-root trick is applied on these descriptors except trajectory descriptors.

We use two methods to aggregate these descriptors: bog-of-visual-words (BoW) encoding and fisher vector (FV) encoding. For BoW, we learn a codebook with a size of $4,000$ for each type of descriptors by k-means clustering as in [7] and quantize the descriptors to their nearest visual words in the codebook. The histogram of the quantized descriptors in one frame is used as the frame-wide feature of the frame. Hence the dimensionality of the frame-wide features is $4,000$. For FV, we first reduce the dimensionality of each type of descriptors by a factor of two using PCA. We then train a Gaussian Mixture Model with $K = 256$ Gaussian components for each type of descriptors. The dimensionality of the frame-wide features by FV is $d = 2KD$ for a type of descriptors, where $D$ is the reduced dimension of the descriptors after PCA.

For the Chalearn Gesture recognition dataset, we employ the skeleton features provided by the authors of [5]. The normalized relative locations of body joints w.r.t the torso joints are calculated and clustered into a codebook with a size of 100. The histogram of the quantized relative locations in one frame is employed as the frame-wide feature with a dimensionality of 100.

**Implementation details.** The order in Eq. 6 can also be inverse, i.e., the rank value computed from the linear function of the previous frame is forced to be smaller than that of the current frame. If the first layer adopts rank pooling, the second layer encodes the results of the first layer with the same order and combines them together. If the first layer adopts mean-pooling, the second layer encodes the results of the first layer in both forward and inverse orders and combines them together. Following [5], we also use the SVR solver of liblinear [55] to solve Eq. 6 and fix the value of $C$ to 1.

On the ChaLearn dataset and the Hollywood2 dataset, when using the BoW encoding, we apply chi-squared kernel map on each time varying mean vector in the second layer and apply the $L_2$ normalization on the output representation of the second layer. The average kernel strategy is adopted to fuse the representations generated from different types of descriptors. When using the FV encoding, we apply the square-root trick on each time varying mean vector in the second layer. The representations of different descriptors are concatenated and we apply $L_2$-normalization to the final representation. For all datasets and both encoding methods, we train linear SVMs for classification.

TABLE I
COMPARISON OF PERFORMANCES USING THE TWO POOLING METHODS ON THE CHALEARN GESTURE DATASET

| Pooling Method | Precision | Recall | F-score |
|---|---|---|---|
| M-HDPE | **78.34** | **78.18** | **78.15** |
| R-HDPE | 75.95 | 75.83 | 75.79 |

TABLE III
COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE OLYMPIC SPORTS DATASETS WITH THE FV-BASED FRAME-WIDE FEATURES

| Pooling Method | $L = 10$ | $L = 20$ | $L = 30$ | $L = 40$ | $L = 50$ |
|---|---|---|---|---|---|
| M-HDPE | 89.69 | **90.37** | **89.75** | **90.37** | **89.44** |
| R-HDPE | **90.28** | 88.47 | 88.49 | 86.47 | 84.28 |

### C. Comparison of pooling in the first layer

In the first layer modeling, the encoding of each division could either be mean pooling or rank pooling as mentioned in III-C. We compare the two pooling methods on the Chalearn Gesture dataset, the Olympic Sports dataset and the Hollywood2 dataset with BoW and FV based frame-wide features, in Tab. I to Tab. V, respectively. M-HDPE and R-HDPE denote that the mean pooling and the rank pooling are used in the first layer modeling in HDPE, respectively.

Generally, the mean pooling outperforms the rank pooling on the ChaLearn gesture dataset and the Olympic Sports dataset, while the rank pooling achieves better results on the Hollywood2 dataset. This verifies the explanation in III-C. That is, for fine-grained actions such as gestures, since the evolution within each division is quite uniform, the within-division dynamic can be ignored, and the local appearance information is enhanced by mean-pooling. Each video in the Olympic Sports dataset contains only one single short sport action performed by one subject, and the divisions of such short action correspond to quite uniform stages of the action. Thus mean pooling within divisions performs better. However, the videos in the Hollywood2 dataset generally contain more than one scenes and actions, performed by multiple subjects. For such complex actions, the complex dynamics within divisions contain important discriminative information of the action and hence cannot be eliminated.

It can also be observed that for video-based generic actions, especially when the FV-based frame-wide features are used, the rank pooling outperforms the mean pooling when the number of divisions is small, while the mean pooling works better when more divisions are parsed. The more the parsed divisions, the finer each division is, and the more stable the dynamic within each division. Conversely, if few divisions are parsed from an action, each division is longer and the dynamic within it should be more complex. Rank pooling should then be used to capture such dynamic within each division.

In the following experiments, we use M-HDPE on the Chalearn gesture dataset and the Olympic Sports dataset, and adopt R-HDPE on the Hollywood2 dataset, unless otherwise specified.

TABLE II
COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE OLYMPIC SPORTS DATASETS WITH THE BOW-BASED FRAME-WIDE FEATURES

| Pooling Method | $L=5$ | $L=10$ | $L=15$ | $L=20$ | $L=25$ | $L=30$ | $L=35$ | $L=40$ | $L=45$ | $L=50$ |
|---|---|---|---|---|---|---|---|---|---|---|
| M-HDPE | **88.20** | **87.82** | **88.87** | **89.12** | **89.18** | **88.07** | **88.74** | **88.30** | **88.58** | **88.63** |
| R-HDPE | 87.45 | 87.64 | 87.26 | 85.35 | 85.60 | 85.51 | 83.89 | 83.62 | 85.54 | 83.11 |

TABLE IV
COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE HOLLYWOOD2 DATASETS WITH THE BOW-BASED FRAME-WIDE FEATURES

| Pooling Method | $L=10$ | $L=20$ | $L=30$ | $L=40$ | $L=50$ |
|---|---|---|---|---|---|
| M-HDPE | 59.07 | 60.99 | 61.68 | 61.36 | 61.72 |
| R-HDPE | **64.93** | **66.54** | **66.16** | **64.25** | **61.75** |

TABLE V
COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE HOLLYWOOD2 DATASETS WITH THE FV-BASED FRAME-WIDE FEATURES

| Pooling Method | $L=10$ | $L=20$ | $L=30$ | $L=40$ | $L=50$ |
|---|---|---|---|---|---|
| M-HDPE | 66.92 | 68.30 | **69.12** | **68.75** | **68.94** |
| R-HDPE | **67.43** | **69.22** | 67.29 | 66.07 | 64.27 |

TABLE VI
COMPARISON OF PERFORMANCES USING THE TWO POOLING METHODS ON THE CHALEARN DATASET

| Alignment Method | Precision | Recall | F-score |
|---|---|---|---|
| M-HDPE + uniform parsing | 77.85 | 77.68 | 77.60 |
| M-HDPE | **78.34** | **78.18** | **78.15** |
| R-HDPE + uniform parsing | 67.49 | 67.48 | 67.33 |
| R-HDPE | 75.95 | 75.83 | 75.79 |

TABLE VII
COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE OLYMPIC SPORTS AND HOLLYWOOD2 DATASETS

| Alignment Method | Olympic | hollywood2 |
|---|---|---|
| HDPE + uniform parsing | 88.41 | 64.69 |
| HDPE | **89.12** | **66.16** |

### *D. Effects of dynamic parsing*

To evaluate the effects of the dynamic parsing by temporal clustering algorithm 1, we compare the HDPE using the temporal clustering with HDPE using uniform parsing on the three datasets. "HDPE+uniform parsing" denotes that the action sequence is first uniformly parsed into divisions, and the two layer hierarchical dynamic encoding is applied. The uniform parsing can be viewed as the initialization of the temporal clustering. The numbers of divisions for both uniform parsing and temporal clustering are set to 7, 20 and 30 on the Chalearn gesture, Olympic Sports and Hollywood2 datasets, respectively. The band factor $f$ is set to 2 for all the datasets. The BoW-based frame-wide features are adopted for the Olympic Sports and Hollywood2 datasets. Both M-HDPE and R-HDPE are evaluated on the Chalearn dataset. The comparisons are shown in Tab. VI and Tab. VII. The proposed temporal clustering outperforms the uniform parsing on all the datasets. This indicates that temporal clustering is able to parse the action into dynamic-coherent divisions and the dynamic parsing did benefit the final performance.

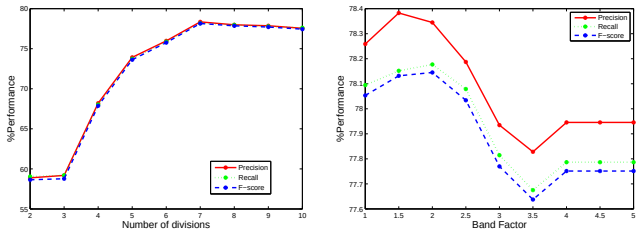
The improvements are more significant for R-HDPE, as reflected in the results on the Chalearn dataset and the Hollywood2 dataset. This suggests that more reliable parsings are required to use rank pooling in the first layer. For mean pooling, the output representation is not sensitive to the outliers in the division. Even for rough parsing, although the frames in some divisions may contain inconsistency outliers, the means of different divisions generally still reflect the evolution of the action. However, the non-linear rank pooling considers the relative ordering of all the frames in each division. If the parsing in the first layer leads to divisions with inconsistent dynamic, the resulted encodings of these divisions encode such non-smooth evolution and hence their ordering relationships are interrupted. It will be difficult to extract discriminative representation from these noisy encodings by rank pooling in the second layer.

### *E. Influence of parameters*

There are mainly two parameters of the proposed HDPE: the number of divisions $L$ for parsing the action sequence by temporal clustering and the band factor $f$ for aligning the sequence to the essential sequence by dynamic programming. We evaluate the influences of the two parameters on the final performance.

We first evaluate the influence of $L$. For the Chalearn Gesture recognition dataset, the average number of frames is 39.7. We fix $f$ to 2, and vary $L$ from 2 to 10. The performances (the precision, recall and F-score) are shown in Fig. 4(a). For the Olympic Sports dataset and the Hollywood2 dataset, we fix $f$ to 2, and vary $L$ from 10 to 50. The performances (mAP) of HDPE using BoW and FV based frame-wide features with the increasing values of $L$ are shown in Fig. 5(a), Fig. 5(b), Fig. 6(a) and Fig. 6(b), respectively. We find that on the Chalearn dataset and the Hollywood2 dataset, at first all performance measures improve with the increase of the number of divisions, because more temporal structures information can be captured. When $L$ is larger than 7 on the Chalearn dataset and 20 on the Hollywood2 dataset, the performances stop increasing. This may be because redundant divisions exist, which break the intrinsic temporal structures and slightly interfere the rank pooling of the second layer. The performances on the Olympic Sports dataset are quite oscillatory with the increasing of $L$, but generally inflection points exist and the amplitudes decay after these points. We set $L$ to be the value of the inflection point for the corresponding dataset and frame-wide features in the subsequent experiments.

HDPE also supports to set different $L$ for different sequences. For example, we can set $L$ as $N/r$, $r$ is a factor




(a) (b)

Fig. 4. The performances with the increase of (a) the number of divisions and (b) the value of band factor on the Chalearn Gesture dataset

measuring averagely how many frames a state should contain and can be estimated according to prior knowledge on the data. We set $L$ to be the same for all sequences, because as long as $L$ is large enough, the evolution of $L$ key stages should contain the information for discriminating different classes. Although the states of a more dynamic action are more complex, the local dynamics within these states are captured by the 1st layer modeling.

We then evaluate the influence of $f$. For the Chalearn dataset, we fix the number of divisions $L$ to be 7, and vary the band factor $f$ from 1 to 5 with a interval of 0.5. When $f = 1$, it means that the alignment is strictly restricted to the uniform alignment. When $f > 4$, the allowed maximal capacity of a division is larger than the length of the sequence, and it is equivalent to perform unconstrained dynamic time warping, which may mistake outliers as individual divisions and lead to extremely unbalanced alignment. The results are shown in fig. 4(b). For the Olympic Sports dataset and the Hollywood2 dataset, we fix $L$ to 20, and vary $f$ from 1.2 to 3. The performances (mAP) of HDPE using BoW and FV based frame-wide features with the increasing values of $f$ are shown in Fig. 5(c), Fig. 5(d), Fig. 6(c) and Fig. 6(d), respectively. For the Olympic Sports dataset with the BoW-based frame-wide features, relative balanced parsings with small amount of wrappings lead to better results. For the other two datasets, sufficient wrappings are required and applying appropriate constraints on the capacity of each division benefits the performances. We set $f$ in the range of 1.5 to 2 in the subsequent experiments. $f = 2$ means that the maximal number of elements within one division should not be larger than twice the average number of elements by uniform alignment.

### F. Effects of multi-layers

As introduced in Sec. III-E, HDPE can be generalized to multiple layers. To evaluate the effects of "deeper" hierarchy, we build HDPE model with three layers and extract the representations from all the layers. The number of divisions parsed in a higher layer is set to the half of the number of divisions in its lower layer. For the Chalearn dataset, the band factor is set to 2, and the number of divisions in the first ground layer is set from 8 to 20 with an interval of 2. The performances by using the representations of different layers and their combinations are shown in Tab. VIII, Tab. IX and Tab. X. For the Olympic Sports dataset, the band factor is set to 1.5, and the number of divisions in the first layer is fixed to 40. "1st+2nd" means that the presentations of the first layer and the second layer are concatenated. "All layers" means that the representations of all the three layers are combined by concatenation.

TABLE VIII
COMPARISON OF PRECISIONS USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE CHALEARN DATASET

| $L$ | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
|---|---|---|---|---|---|---|---|
| 1st layer | **77.70** | 77.53 | 77.42 | 76.54 | 76.24 | 76.36 | 75.52 |
| 2nd layer | 68.92 | 73.84 | 76.92 | **77.79** | 76.60 | 76.49 | 76.27 |
| 3rd layer | 57.01 | 59.30 | 62.10 | 68.90 | 68.74 | 70.34 | **72.63** |
| 1st+2nd | 78.76 | **79.09** | 78.44 | 78.10 | 77.24 | 76.76 | 76.53 |
| All layers | 78.13 | **78.78** | 77.95 | 77.36 | 78.20 | 77.36 | 76.30 |

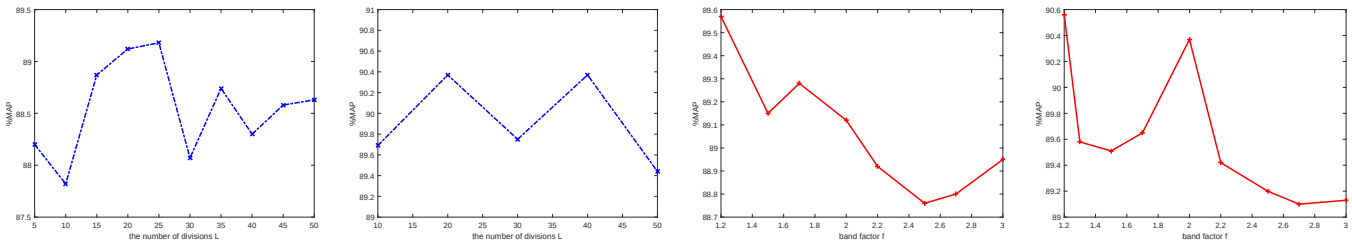
TABLE IX
COMPARISON OF RECALLS USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE CHALEARN DATASET

| $L$ | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
|---|---|---|---|---|---|---|---|
| 1st layer | **77.77** | 77.59 | 77.30 | 76.61 | 76.33 | 76.27 | 75.43 |
| 2nd layer | 68.88 | 73.60 | 76.71 | **77.48** | 76.60 | 76.45 | 76.16 |
| 3rd layer | 57.30 | 59.25 | 62.14 | 68.63 | 68.65 | 70.26 | **72.59** |
| 1st+2nd | 78.85 | **78.99** | 78.47 | 78.08 | 77.20 | 76.65 | 76.54 |
| All layers | 78.21 | **78.75** | 78.03 | 77.27 | 78.19 | 77.28 | 76.28 |

We can find that on both datasets, the performances of the third layer are worse than the first two layers. This may indicate that two layers w.r.t. key poses and samples of the pose are enough for simple actions such as gestures and single sport actions. However, when the number of divisions in the first layer increases, the performances of the third layer are continuously improved on the Chalearn dataset. This means that finer parsing in the first layer is necessary to build more layers for HDPE, because more layers aim to parse the dynamics into finer levels, and the more the number of layers, the more meticulous the first layer parsing should be. The combinations of the three layers outperform the combinations of only the first two layers when enough divisions are parsed in the first layer. This may imply that high layers encoding spectrum of dynamics contain discriminative information that benefits the final performance. For more complex activities, additional layers w.r.t. high-level semantic can be more beneficial.

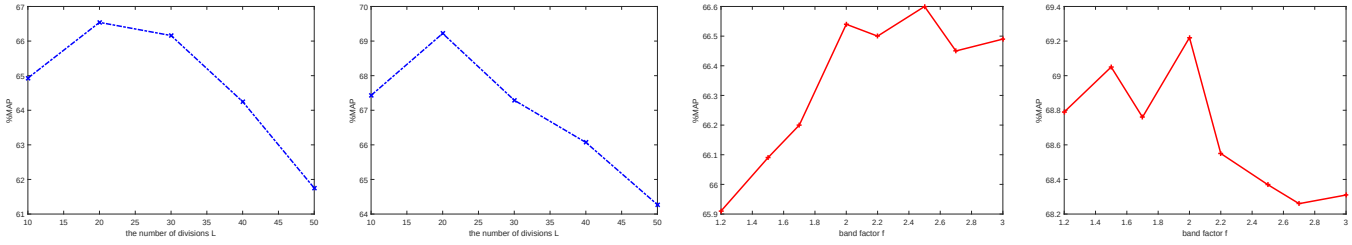
TABLE X
COMPARISON OF F-SCORES USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE CHALEARN DATASET

| $L$ | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
|---|---|---|---|---|---|---|---|
| 1st layer | **77.64** | 77.45 | 77.19 | 76.45 | 76.18 | 76.16 | 75.29 |
| 2nd layer | 68.75 | 73.53 | 76.67 | **77.51** | 76.49 | 76.37 | 76.08 |
| 3rd layer | 56.90 | 58.94 | 61.84 | 68.54 | 68.49 | 70.14 | **72.44** |
| 1st+2nd | 78.69 | **78.91** | 78.33 | 77.98 | 77.11 | 76.54 | 76.39 |
| All layers | 78.04 | **78.64** | 77.89 | 77.21 | 78.09 | 77.16 | 76.13 |




(a) (b) (c) (d)

Fig. 5. The performances of HDPE with the increase of (a) the number of divisions using the BoW-based frame-wide features (b) the number of divisions using the FV-based frame-wide features (c) the value of band factor using the BoW-based frame-wide features and (d) the value of band factor using the FV-based frame-wide features on the Olympic Sports dataset




(a) (b) (c) (d)

Fig. 6. The performances of HDPE with the increase of (a) the number of divisions using the BoW-based frame-wide features (b) the number of divisions using the FV-based frame-wide features (c) the value of band factor using the BoW-based frame-wide features and (d) the value of band factor using the FV-based frame-wide features on the Hollywood2 dataset

TABLE XI
COMPARISON OF MAPS USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE OLYMPIC SPORTS DATASET

| Layer | 1st | 2nd | 3rd | 1st+2nd | 1st+2nd+3rd |
|---|---|---|---|---|---|
| MAP | 90.52 | 89.90 | 88.89 | **90.67** | 90.64 |

TABLE XII
COMPARISON OF MAPS USING DIFFERENT COMBINATIONS ON THE OLYMPIC SPORTS AND HOLLYWOOD2 DATASETS

| Alignment Method | Olympic | hollywood2 |
|---|---|---|
| HDPE+IDT | 90.85 | 69.26 |
| HDPE+rank pooling | **91.18** | **70.40** |
| HDPE+both | 90.94 | **70.40** |

### G. Combinations

A potential advantage of the proposed method is the representations produced from different numbers of partitions in the first layer encode the temporal structures in different scales. If the number of divisions is set to 1, the temporal information is totally discarded and the proposed HDPE method boils down to the "IDT" method [7]. If the number of divisions is set to be the length of the sequence, no local appearances are smoothed and the proposed HDPE method boils down to the "rank pooling" method [5]. The more divisions are parsed from the action, the finer the scale of the captured temporal information is. The representations generated in different scales provide complementary information to each other. Combining them together incorporates multi-scale temporal information together. We evaluate the combinations of HDPE where 20 divisions are parsed in the first layer with either "local" or "rank pooling" or both on the Olympic Sports dataset and the Hollywood2 dataset. FV-based frame-wide features are used for both datasets. The comparisons are shown in Tab. XII. We find that the combinations of HDPE with "rank pooling" achieve the best results on both datasets. The combinations are achieved by simple concatenation. More advanced fusion methods and combinations with more middle temporal scale parsings in the first layer may further improve the performance.

### H. Comparison with state-of-the-art

It may be difficult to perform a fair comparison with state-of-the-art results because different methods use different components such as types of features and sample argument methods. The state-of-the-art results are usually achieved by fusing different types of features and adopting data augmentation techniques. We compare the proposed HDPE with the improved dense trajectory features (denoted by "IDT") encoded by Bag-of-Words (BoW) or Fisher Vector (FV) encoding [7] and learning to rank based temporal encoding (denoted by "rank pooling") [5] of the whole video as well as the several other state-of-the-art results on the three datasets, as shown in Tab. XIII, Tab. XIV Tab. XV. For HDPE, the number of divisions for each video is set to be 7, 20 and 20 for the ChaLearn Gesture dataset, the Olympic Sports dataset and the Hollywood2 dataset, respectively. The band factor is set to be 2 for all these datasets. Mean pooling is adopted for the ChaLearn dataset and the Olympic Sports dataset, while rank pooling is adopted for the Hollywood2 dataset in the first layer modeling.

TABLE XIII
COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE CHALEARN GESTURE DATASET

| Method | Precision | Recall | F-score |
|---|---|---|---|
| Wu et al. [56] | 59.9 | 59.3 | 59.6 |
| Yao et al. [57] | - | - | 56.0 |
| Pfister et al. [58] | 61.2 | 62.3 | 61.7 |
| Fernando et al. [5] | 75.3 | 75.1 | 75.2 |
| Rank pooling [5] | 74.0 | 73.8 | 73.9 |
| HDPE | **78.34** | **78.18** | **78.15** |
| 1st+2nd HDPE | **79.09** | **78.99** | **78.91** |

TABLE XIV
COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE OLYMPIC SPORTS DATASET. MAP IS USED AS THE PERFORMANCE MEASURE

| Method | Olympic Sports |
|---|---|
| Brendel et al. [36] | 77.3 |
| Gaidon et al. [59] | 82.7 |
| Jain et al. [21] | 83.2 |
| Wang et al. [7] (IDT+FV) | **91.1** |
| IDT(BoW) [7] | 83.3 |
| HDPE(BoW) | **89.12** |
| HDPE(FV) | **90.37** |
| HDPE+Rank pooling(FV) | **91.18** |

TABLE XV
COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE HOLLYWOOD2 DATASET. MAP IS USED AS THE PERFORMANCE MEASURE. * DENOTES THAT THE RESULT IS REPORTED BY OUR REPRODUCTION WITH THE BOW REPRESENTATION

| Method | Hollywood2 |
|---|---|
| Jain et al. [21] | 62.5 |
| Wang et al. [7] | 64.3 |
| Hoai et al. [60] | 73.6 |
| Fernando et al. [5] | **73.7** |
| IDT(BoW) [7] | 62.2 |
| Rank pooling(BoW) [5]* | 62.19 |
| HDPE(BoW) | **66.54** |
| IDT(FV) [7] | 64.3 |
| Rank pooling+IDT(FV) [5] | 70.0 |
| HDPE(FV) | **69.22** |
| HDPE+Rank pooling(FV) | **70.40** |

TABLE XVI
COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE OLYMPIC SPORTS DATASET. ACCURACY IS USED AS THE PERFORMANCE MEASURE

| Method | Accuracy |
|---|---|
| Laptev et al. [8] | 62.0 |
| Niebles et al. [43] | 72.1 |
| Tang et al. [61] | 66.8 |
| Wang et al. [34] | 73.8 |
| HDPE | **81.34** |
| HDPE+Rank Pooling+IDT | **83.58** |

From Tab. XIII, it can be observed that the proposed method outperforms the state-of-the-art method [5] on the ChaLearn gesture dataset. In [5], the results are achieved by combining the rank pooling representation with local method, and the results by rank pooling along are also reported, as denoted by "Rank pooling". Since we use the same frame-wide features provided by [5], the superior performance comes from the hierarchical parsing and modeling. Combination with the first layer pooling further improves the performances of HDPE.

Tab. XIV shows that the result of the single HDPE with FV-based frame-wide features is slightly worse than the best result reported in [7]. There is a certain randomness when extracting IDT descriptors and FV encoding. Our reproduction of IDT with FV is about 89%. Based on strictly the same features, HDPE outperforms IDT. When combined with rank pooling, HDPE outperforms the reported result. [7] also reports their results with the Bag-of-words encoding, as denoted by "IDT(BoW)" in Tab. XIV. Our method outperforms the IDT method that encoding descriptors in all frames into a single representation without considering the temporal information by a margin of 4%.

As shown in Tab. XV, on the Hollywood2 dataset, Fernando et al. [5] and Hoai et al. [60] achieve higher mAPs. Besides the Fisher Vector encoding, both the two work also adopt the data augmentation technique proposed in [60], which double the training data by flipping each video and average the classification scores of each test video and its mirrored version. The performance of our method may also be improved by applying such data augmentation to our method with the cost of time. We did not use this technique because both the time and space complexities are doubled, and we only focus on the evaluation of the proposed modeling method over other modeling method rather than the absolute performance. Even though, the combination of HDPE with rank pooling outperforms the combination of rank pooling and IDT with argumentation reported in [5]. On the basis of the same BoW feature encoding method, our method significantly outperforms the "IDT(BoW)" method reported in [7] and our reproduction of rank pooling by a margin of 4%.

We also evaluate the multi-class accuracy on the Olympic Sports dataset in Tab. XVI. The BoW based frame-wide features are used. The proposed HDPE representation itself significantly outperforms the reported results by a margin of 7.5%, and the combination of the "IDT", rank pooling and the proposed HDPE representations further extends the margin to about 10%.

## V. CONCLUSIONS

In this paper we have presented a hierarchical dynamic parsing and encoding method for action recognition, which unsupervised learns higher-level representation from a single action sequence by exploring the temporal structures and building the hierarchical architecture. The hierarchy disentangles the local appearances and the global dynamic into different layers. In the lower layer, the sequence is parsed into different divisions, and local appearance information within each uniformly-evolved division is captured via local mean or rank pooling. In the higher layer, the global dynamic of the appearances among the divisions is encoded. The learned representation is robust, because outliers or noisy frames cannot directly impact on the global dynamic since they must be assigned to a corresponding division, while their influence within a division is greatly diminished by pooling. Experimental results on several action datasets have demonstrated the potential of the proposed method. Our future

work involves exploring the fusion of multi-scale partitions to incorporate multi-scale temporal information.

## REFERENCES


[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings. IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1470–1477.

[2] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *European Conference on Computer Vision*. Springer, 2012, pp. 286–299.

[3] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition," in *IEEE International Conference on Computer Vision*, vol. 2. IEEE, 2005, pp. 1808–1815.

[4] B. Yao and S.-C. Zhu, "Learning deformable action templates from cluttered videos," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1507–1514.

[5] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5378–5387.

[6] B. Su, J. Zhou, X. Ding, H. Wang, and Y. Wu, "Hierarchical dynamic parsing and encoding for action recognition," in *European Conference on Computer Vision*. Springer, 2016.

[7] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[9] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference*. British Machine Vision Association, 2008, pp. 275: 1–10.

[10] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1932–1939.

[11] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 3169–3176.

[12] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[13] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *International Conference on Computer Vision*. IEEE, 2011, pp. 2486–2493.

[14] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1794–1801.

[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3360–3367.

[16] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *European conference on computer vision*. Springer, 2010, pp. 141–154.

[17] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 596–603.

[18] X. Yang and Y. Tian, "Action recognition using super sparse coding vector with spatio-temporal awareness," in *European Conference on Computer Vision*. Springer, 2014, pp. 727–741.

[19] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European conference on computer vision*. Springer, 2010, pp. 143–156.

[20] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[21] M. Jain, H. Jégou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2555–2562.

[22] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 2004–2011.

[23] J. Wang, Z. Chen, and Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 3185–3192.

[24] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2046–2053.

[25] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *European Conference on Computer Vision*. Springer, 2014, pp. 581–595.

[26] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1290–1297.

[27] ——, "Learning actionlet ensemble for 3d human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 914–927, 2014.

[28] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu, "Action recognition with actons," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3559–3566.

[29] M. Sapienza, F. Cuzzolin, and P. H. Torr, "Learning discriminative space–time action parts from weakly labelled videos," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 30–47, 2014.

[30] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis, "Representing videos using mid-level discriminative patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2571–2578.

[31] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314.

[32] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3218–3226.

[33] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, D. Blei and F. Bach, Eds. JMLR Workshop and Conference Proceedings, 2015, pp. 843–852.

[34] J. Wang and Y. Wu, "Learning maximum margin temporal warping for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2688–2695.

[35] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3201–3208.

[36] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 778–785.

[37] B. Wu, C. Yuan, and W. Hu, "Human action recognition based on context-dependent graph kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2609–2616.

[38] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[39] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4041–4049.

[40] B. Su and X. Ding, "Linear sequence discriminant analysis: a model-based dimensionality reduction method for vector sequences," in *Proc. IEEE Int'l Conf. Computer Vision*, 2013, pp. 889–896.

[41] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *Computer vision and pattern recognition, 1997. proceedings., 1997 ieee computer society conference on*. IEEE, 1997, pp. 994–999.

[42] S. Hongeng and R. Nevatia, "Large-scale event detection using semi-hidden markov models," in *Proceedings. IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1455–1462.

[43] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *European conference on computer vision*. Springer, 2010, pp. 392–405.

[44] Y. Wang and G. Mori, "Hidden part models for human action recognition: Probabilistic versus max margin," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1310–1323, 2011.

[45] T. Xiang and S. Gong, "Beyond tracking: Modelling activity and understanding behaviour," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 21–51, 2006.

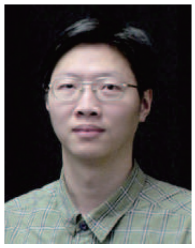
[46] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing video events with goal inference and intent prediction," in *International Conference on Computer Vision*. IEEE, 2011, pp. 487–494.
[47] G. Luo, S. Yang, G. Tian, C. Yuan, W. Hu, and S. J. Maybank, "Learning human actions by combining global dynamics and local appearance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2466–2482, 2014.
[48] Y. Song, L.-P. Morency, and R. Davis, "Action recognition by hierarchical sequence summarization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3562–3569.
[49] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2649–2656.
[50] F. Zhou, F. De la Torre, and J. F. Cohn, "Unsupervised discovery of facial events," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2574–2581.
[51] M. Hoai and F. De la Torre, "Maximum margin temporal clustering," in *Proceedings of international conference on artificial intelligence and statistics*, 2012, pp. 1–9.
[52] B. Su, X. Ding, C. Liu, and Y. Wu, "Heteroscedastic max-min distance analysis," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2015, pp. 4539–4547.
[53] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.
[54] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 1297–1304.
[55] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.
[56] J. Wu, J. Cheng, C. Zhao, and H. Lu, "Fusing multi-modal features for gesture recognition," in *Proceedings of the 15th ACM on International conference on multimodal interaction*. ACM, 2013, pp. 453–460.

**Bing Su** received the B.S. degree in information engineering from Beijing Institute of Technology, Beijing, in 2010, and the Ph.D. degree in Electronic Engineering from Tsinghua University, Beijing, in 2016. Currently, he is an Assistant Professor in Institute of Software Chinese Academy of Sciences, Beijing. His research interests include pattern recognition, computer vision and machine learning.

**Jiahuan Zhou** received his B.E. (2013) from Tsinghua University. He is currently working toward the PhD degree in the Department of Electrical Engineering & Computer Science, Northwestern University. His current research interests include computer vision, pattern recognition and machine learning.

[57] A. Yao, L. Van Gool, and P. Kohli, "Gesture recognition portfolios for personalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1915–1922.
[58] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *European Conference on Computer Vision*. Springer, 2014, pp. 814–829.
[59] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," in *British Machine Vision Conference*. BMVA Press, 2012, pp. 30–1.
[60] M. Hoai and A. Zisserman, "Improving human action recognition using score distribution and ranking," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 3–20.
[61] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1250–1257.

**Xiaoqing Ding** graduated from Tsinghua University, China, in 1962 won the graduate Golden Medal. Now she is a professor and PhD supervisor in Department of Electronic Engineering, Tsinghua University. She is an IEEE fellow and an IAPR Fellow. Her research interests include pattern recognition, image processing, characters recognition, biometrics, computer vision and video surveillance; she has won four the top prestigious National Scientific and Technical Progress Awards in China for multi-language character & document recognition and for face and writer recognition etc. in 1992, 1998, 2003 and 2008 respectively. She has published more than 550 papers, coauthored 2 books and holds 27 Invention Patents. She has served as program committee member of many international conferences and editor of international journals.



**Ying Wu** received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, the M.S. degree from Tsinghua University, Beijing, China, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, Illinois, in 1994, 1997, and 2001, respectively. From 1997 to 2001, he was a Research Assistant with the Beckman Institute for Advanced Science and Technology, UIUC. From 1999 to 2000, he was a Research Intern with Microsoft Research, Redmond, WA, USA. In 2001, he joined the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA, as an Assistant Professor. He was promoted as an Associate Professor in 2007 and a Full Professor in 2012. He is currently a Full Professor of Electrical Engineering and Computer Science with Northwestern University. His current research interests include computer vision, image and video analysis, pattern recognition, machine learning, multimedia data mining, and human-computer interaction. He serves as an Associate Editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Image Processing*, the *IEEE Transactions on Circuits and Systems for Video Technology*, the *SPIE Journal of Electronic Imaging*, and the *IAPR Journal of Machine Vision and Applications*. He received the Robert T. Chien Award by UIUC in 2001 and the NSF CAREER Award in 2003.