Hierarchical Dynamic Parsing and Encoding for Action Recognition

Bing Su[1], Jiahuan Zhou[2], Hao Wang[1], Ying Wu[2]

[1]Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
subingats@gmail.com, wanghao@iscas.ac.cn
[2]Department of Electrical Engineering and Computer Science,
Northwestern University, Evanston, IL, 60208, USA
{jzt011, yingwu}@eecs.northwestern.edu

Abstract. A video action generally exhibits quite complex rhythms and non-stationary dynamics. To model such non-uniform dynamics, this paper describes a novel hierarchical dynamic encoding method to capture both the locally smooth dynamics and globally drastic dynamic changes. It provides a multi-layer joint representation for temporally modeling 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. The dynamics within each stage are encoded by mean-pooling or learning to rank based encoding. At the second layer, the temporal information of the ordered dynamics extracted from the previous layer is encoded again to form the overall representation. Extensive experiments on a gesture action dataset (Chalearn) and several generic action datasets (Olympic Sports and Hollywood2) have demonstrated the effectiveness of the proposed method.

Keywords: Action Recognition, Hierarchical Modeling, Dynamic Encoding

1 Introduction

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 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 HMM [2] and CRF [3] or exemplar-based such as 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

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.

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 the hierarchy of two layers into a joint representation for action recognition.In the first layer, we parse the sequence of frame-wide features into different stages and encode the dynamics and appearances into a feature vector within each stage. In the second layer, we extract a high-level dynamic encoding representation by pooling the 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, meanwhile, it also captures the high-level semantic information for a video. 2) We propose an unsupervised method for temporal clustering to achieve efficient dynamic parsing. 3) The extracted representations from multi-scale parsings provide complementary discriminative information and hence can be readily combined.

2 Related work

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, HOF and MBH are typical low-level descriptors used in video-based action recognition [6]. These descriptors can be computed either sparsely at local space-time cuboids [7] or by dense sampling scheme [6]. Various coding variants have also been proposed to encode these local descriptors, such as Fisher vector [8] and vector of locally aggregated descriptors [9, 10]. Efforts have also been made to construct hierarchical feature representations based on BoW to capture context information and high-level concepts [11, 12]. Besides these hand-crafted features, deep neural networks have also been applied to learn representations directly from videos, such as the trajectory-pooled deep-convolutional descriptors [13] and the pose-based convolutional neural network features [14].

Dynamic-based action modeling approaches. Both generative models and deterministic models have been studied to model and represent dynamics and motions in action recognition. Generative models are typically based on temporal (hidden) state-space, such as HMM [15, 2, 16], CRF [3, 17, 18], temporal AND-OR graph [19], and linear dynamic systems [20]. The dynamics in generative models refer to the internal hidden states and transitions, e.g., Multilevel motions of the bodies and parts are governed by such internal dynamics in [15]. A large amount of samples and complex computations are required to train such models. We use dynamics as a milder term to indicate the evolution of frame appearances. Our method directly captures such evolution from the single sequence in an unsupervised fashion.

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 [21] to learn temporal action alignments and phantom action templates. Actom sequence model [22] and graphs [23] are also used to model temporal structures and relationships among local features. Recently deep neural architectures are employed for modeling actions. In [24], spatial and temporal nets are incorporated into a two-stream ConvNet. In [25], salient dynamics of actions are modeled by the differential recurrent neural networks.

Temporal clustering. Aligned Cluster Analysis [26] 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 [27], 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.

3 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

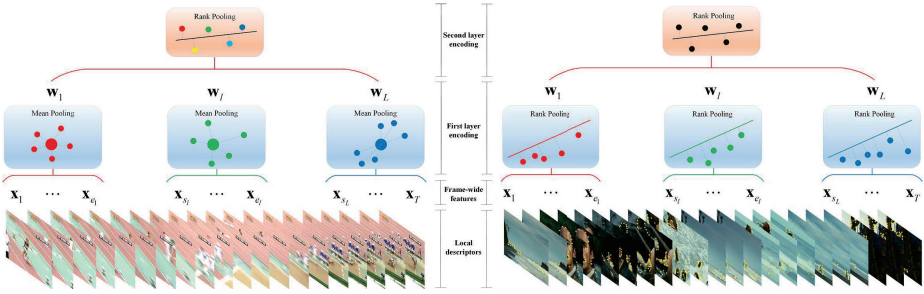


Fig. 2. The pipeline of the proposed method. The first layer can either adopt mean pooling (left) or rank pooling (right).

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. We construct the hierarchy with two layers in this paper, and note that it can be easily generalized to more layers.

3.1 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 partial sequences reach the end, the minimal distance along the optimal path is determined by $\min\limits_{l=1}^{f \cdot l_{ave}} d(T, L, l)$ 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:

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

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

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 (1). Hence both procedures reduce the objective of (1). (1) has a trivial lower bound $\sum_{j=1}^{L} \sum_{i=s_j}^{e_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2 \geq 0, \forall \mathbf{P}, \mathbf{U}$. Hence the partition learning algorithm will at least converge to a local minimum.

Computational complexity. The complexities of dynamic programming (2) and calculating (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.

3.2 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 $f(\mathbf{w}_l; \mathbf{v}_{s_l}) > f(\mathbf{w}_l; \mathbf{v}_{s_l+1}) > \cdots > f(\mathbf{w}_l; \mathbf{v}_{e_l})$.

$$\begin{aligned} &\arg\min_{\mathbf{w}_l} \tfrac{1}{2}\|\mathbf{w}_l\|^2 + C \sum_{0\le a<b\le e_l-s_l} \varepsilon_{ab} \\ &s.t.\ \mathbf{w}_l^T \cdot (\mathbf{v}_{s_l+a} - \mathbf{v}_{s_l+b}) \ge 1 - \varepsilon_{ab}, \\ &\varepsilon_{ab} \ge 0, \forall 0 \le a < b \le e_l - s_l \end{aligned} \tag{4}$$

$\mathbf{w}_l$ 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.

3.3 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 $f(\mathbf{y}; \mathbf{w}_l{}') > f(\mathbf{y}; \mathbf{w}_k{}'), \forall 1 \le k < l \le L$. 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 hierarchical dynamic parsing and encoding method are as follows. First, the method is totally unsupervised, simple and easy to perform. Both the parsing and the hierarchical encoding are 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.

4 Experiments

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

4.1 Datasets

ChaLearn Gesture Recognition dataset [28]. 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 [28, 5].

Olympic Sports dataset [17]. 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 [6] and the accuracy as in [21].

Hollywood2 dataset [29]. 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 [29, 6].

4.2 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 [6], 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. The square-root trick is applied on these descriptors except

trajectory descriptors. We learn a codebook with a size of $4,000$ for each type of descriptors by k-means clustering as in [6] 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 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.

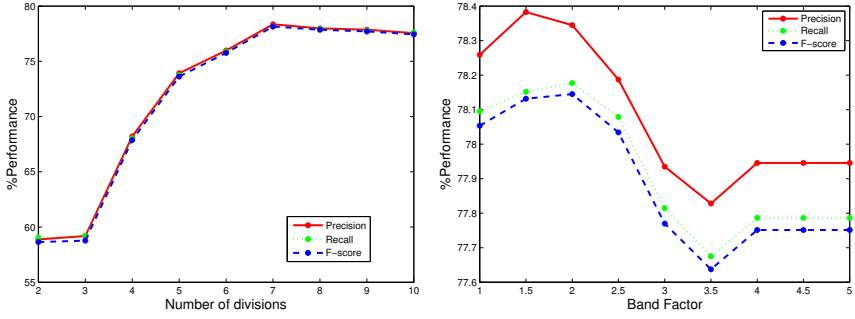
Implementation details. On the ChaLearn dataset and the Hollywood2 dataset, we apply chi-squared kernel map on each time varying mean vector when using rank pooling. On the Olympic Sports dataset, we apply chi-squared kernel map on the output representation of the second layer rather than in the second layer pooling, while the square-root trick is applied in the second layer pooling. The order in Eq. 4 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 [30] to solve Eq. 4 and fix the value of C to 1. When improved dense trajectory features are used, the features of different descriptors are concatenated. We apply L_2-normalization to the final representation and train linear SVMs for classification.

4.3 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 on the Chalearn Gesture recognition dataset. The average number of frames of the dataset is 39.7. We first fix f to be 2, and vary L from 2 to 10. The performances (the precision, recall and F-score) are shown in Fig. 3(a). We find that 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, 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. However, the decline of the performances is not significant with redundant divisions. Thus we will set a relatively larger value for L 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 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.



(a) (b)

Fig. 3. The performances with the increase of (a) the number of divisions and (b) the value of band factor on the Chalearn Gusture dataset.

We then fix the number of divisions 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. We find that 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.

4.4 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 3.2. We compare the two pooling methods on the Chalearn Gesture dataset, the Olympic Sports dataset and the Hollywood2 dataset in Tab. 1 and Tab. 2, respectively. M-HPDE and R-HPDE denote that the mean pooling and the rank pooling are used in the first layer modeling in HPDE, respectively. The mean pooling outperforms the rank pooling on the ChaLearn gesture dataset, while the rank pooling achieves better results on the Olympic sports and Hollywood2 datasets. This verifies the explanation in 3.2. 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. For generic and complex actions, the complex dynamics within divisions contain important discriminative information of the action and hence cannot be eliminated.

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

Table 1. Comparison of performances using the two pooling methods on the ChaLearn dataset.

Pooling Method	Olympic	hollywood2
M-HPDE	84.58	62.90
R-HPDE	87.66	63.51

Table 2. Comparison of MAPs using the two pooling methods on the Olympic Sports and Hollywood2 datasets.

4.5 Comparison with state-of-the-art

We compare the proposed HDPE with the improved dense trajectory features encoded by Bag-of-Words or Fisher Vector encoding [6] and learning to rank based temporal encoding (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. 3, Tab. 4 Tab. 5. For HDPE, the number of divisions for each video is set to be 7, 10 and 10 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. Note that these parameters for the Olympic Sports dataset and the Hollywood2 dataset are set by intuitively judging the dynamic complexity from the average length of videos. Carefully tuning these parameters may further improve the performances. Mean pooling is adopted for the ChaLearn dataset and Rank pooling is adopted for the Olympic Sports dataset and the Hollywood2 dataset in the first layer modeling.

Method	Precision	Recall	F-score
Wu et al. [31]	59.9	59.3	59.6
Yao et al. [32]	-	-	56.0
Pfister et al. [33]	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
HPDE	**78.34**	**78.18**	**78.15**

Table 3. Comparison of the proposed HPDE with state-of-the-art results on the ChaLearn gesture dataset.

From Tab. 3, 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.

Tab. 4 shows that our result is slightly worse than the best result reported in [6], which is achieved by using the advanced Fisher Vector encoding. We use the Bag-of-Words encoding because the dimensionality of the frame-wide features encoded by Fisher Vector per descriptor is about 25,600, which is much higher than the Bag-of-Words encoding (4,000), and this will greatly increase the computation time of the temporal clustering algorithm 1. [6] also reports their results with the Bag-of-words encoding, as denoted by "Local+BoW" in Tab. 4. Our method outperforms this method that encoding descriptors in all frames into a single representation without considering the temporal information by a margin of 4%. Since Fisher Vector encoding improves the mAP from 83.3% by BoW to 91.1% on this dataset [6], applying our method to frame-wide features with Fisher Vector encoding can also be expected to achieve much better result, with a cost of much more computation time. Carefully tuning the parameters L and f may also lead to performance improvement.

Method	Olympic Sports
Brendel et al. [23]	77.3
Gaidon et al. [34]	82.7
Jain et al. [10]	83.2
Wang et al. [6]	**91.1**
Local+BoW [6]	83.3
HPDE	87.66
HPDE+Rank pooling	**89.09**

Table 4. Comparison of the proposed HPDE with state-of-the-art results on the Olympic Sports dataset. mAP is used as the performance measure.

Method	Hollywood2
Jain et al. [10]	62.5
Wang et al. [6]	64.3
Hoai et al. [35]	73.6
Fernando et al. [5]	**73.7**
Local+BoW [6]	62.2
Rank pooling+BoW [5]*	62.19
HPDE	**63.51**

Table 5. Comparison of the proposed HPDE 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.

As shown in Tab. 5, on the Hollywood2 dataset, Fernando et al. [5] and Hoai et al. [35] achieve much higher mAPs. Besides the Fisher Vector encoding, both the two work also adopt the data augmentation technique proposed in [35], 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 and Fisher Vector encoding 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. On the basis of the same BoW feature encoding method, our method outperforms the "Local+BoW" method reported in [6].

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 "Local+BoW" method [6]. 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 perform preliminary experiments on the Olympic Sports dataset to verify this. As shown in Tab. 4, concatenating the representations of the proposed method and the rank pooling method leads to a improvement of about 2% in mAP. We also evaluate the multi-class accuracy in Tab. 6, the proposed HDPE representation itself significantly outperforms the reported results by a margin of 7.5%, and the combination of the "local", rank pooling and the proposed HDPE representations further extends the margin to about 10%.

Method	Accuracy
Laptev et al. [29]	62.0
Niebles et al. [17]	72.1
Tang et al. [36]	66.8
Wang et al. [21]	73.8
HPDE	**81.34**
HPDE+Rank Pooling+Local	**83.58**

Table 6. Comparison of the proposed HPDE with state-of-the-art results on the Olympic sports dataset. Accuracy is used as the performance measure.

5 Conclusions

In this paper we have presented a hierarchical dynamic parsing and encoding method for action recognition, which unsupervised learns higher-level representations 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.

Acknowledgements

This work was supported by National Basic Research Program of China (2013CB329305), Natural Science Foundation of China (61303164, 61402447, 61502466) and Development Plan of Outstanding Young Talent from Institute of Software, Chinese Academy of Sciences (ISCAS2014-JQ02). This work was also supported by National Science Foundation grant IIS-0916607, IIS-1217302.

References

1. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV. (2003)
2. Li, K., Hu, J., Fu, Y.: Modeling complex temporal composition of actionlets for activity prediction. In: ECCV. (2012)
3. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Conditional models for contextual human motion recognition. In: ICCV. (2005)
4. Yao, B., Zhu, S.C.: Learning deformable action templates from cluttered videos. In: ICCV. (2009)
5. Fernando, B., Gavves, E., M., J.O., Ghodrati, A., Tuytelaars, T.: Modeling video evolution for action recognition. In: CVPR. (2015)
6. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV. (2013)
7. Laptev, I.: On space-time interest points. IJCV **64**(2) (2005) 107–123
8. Perronnin, F., Snchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: ECCV. (2010)
9. Jgou, H., Douze, M., Schmid, C., Prez, P.: Aggregating local descriptors into a compact image representation. In: CVPR. (2010)
10. Jain, M., Jegou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR. (2013)
11. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: CVPR. (2010)
12. Peng, X., Zou, C., Qiao, Y., Peng, Q.: Action recognition with stacked fisher vectors. In: ECCV. (2014)
13. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: CVPR. (2015)

14. Chéron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: ICCV. (2015)
15. Bregler, C.: Learning and recognizing human dynamics in video sequences. In: CVPR. (1997)
16. Su, B., Ding, X.: Linear sequence discriminant analysis: a model-based dimensionality reduction method for vector sequences. In: ICCV. (2013)
17. Niebles, J.C., Chen, C.W., Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In: ECCV. (2010)
18. Song, Y., Morency, L.P., Davis, R.: Action recognition by hierarchical sequence summarization. In: CVPR. (2013)
19. Pei, M., Jia, Y., Zhu, S.C.: Parsing video events with goal inference and intent prediction. In: ICCV. (2011)
20. Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R.: Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: CVPR. (2009)
21. Wang, J., Wu, Y.: Learning maximum margin temporal warping for action recognition. In: ICCV. (2013)
22. Gaidon, A., Harchaoui, Z., Schmid, C.: Actom sequence models for efficient action detection. In: CVPR. (2011)
23. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV. (2011)
24. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS. (2014)
25. Veeriah, V., Zhuang, N., Qi, G.J.: Differential recurrent neural networks for action recognition. In: ICCV. (2015)
26. Zhou, F., De la Torre, F., Cohn, J.F.: Unsupervised discovery of facial events. In: CVPR. (2010)
27. Hoai, M., De la Torre, F.: Maximum margin temporal clustering. In: International conference on artificial intelligence and statistics. (2012)
28. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., , Blake, A.: Real-time human pose recognition in parts from a single depth image. In: CVPR. (2011)
29. Laptev, I., Marszaek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008)
30. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. JMLR **9** (2008) 1871–1874
31. Wu, J., Cheng, J., Zhao, C., Lu, H.: Fusing multi-modal features for gesture recognition. In: ICMI. (2013)
32. Yao, A., Gool, L.V., Kohli, P.: Gesture recognition portfolios for personalization. In: CVPR. (2014)
33. Pfister, T., Charles, J., Zisserman, A.: Domain-adaptive discriminative one-shot learning of gestures. In: ECCV. (2014)
34. Gaidon, A., Harchaoui, Z., Schmid, C.: Recognizing activities with cluster-trees of tracklets. In: BMVC. (2012)
35. Hoai, M., Zisserman, A.: Improving human action recognition using score distribution and ranking. In: ACCV. (2014)
36. Tang, K., Fei-Fei, L., Koller, D.: Learning latent temporal structure for complex event detection. In: CVPR. (2012)