Human Action Recognition With Multiple-Instance Markov Model

Wen Zhou and Zhong Zhang

***Abstract*— Recognizing human actions in complex scenes is a challenging problem due to background clutters, camera motion, occlusions, and illumination variations. Markov models are widely used to model temporal statistical relationships among elementary actions for human action recognition. However, traditional Markov models cannot model long-range temporal relations for complex activities, and the states of elementary actions may be unstable due to unwanted background local features. In this paper, we propose a multiple-instance Markov model for human action recognition to address these issues. Our contributions are twofold. First, a novel representation for elementary actions is proposed to encode the movements of local parts. Based on this representation, our method selects elementary actions with stable states due to our multiple-instance formulation. Second, we build multiple Markov chains, which encode both local and long-range temporal information among elementary actions, to represent each video. Multiple-instance formulation allows our model to capture the most discriminative Markov chain for action representation. We evaluate the proposed model on a variety of data sets. Experimental results demonstrate its effectiveness for human action recognition.**

***Index Terms*— Action recognition, Markov chains, multiple-instance learning, max-margin method.**

I. Introduction

RECENTLY, human action recognition has received considerable attentions due to its wide applications, such as video indexing, multimedia retrieval and video surveillance. However, recognizing human action in complex scenes is still a challenging problem owing to camera motion, illumination variations, background clutters and occlusions in videos. One of the predominant approaches for human action recognition is to employ spatio-temporal interest points (STIPs) coupled with bag-of-words (BOW) model, and it successfully recognizes simple actions without any spatial and temporal information [1]–[3]. Exploring discriminative STIPs make it easy to distinguish periodic actions such as walking and running. However, these methods cannot handle complex activities since STIPs only encode local information and BOW model ignores spatial and temporal relationships among features.

To overcome this problem, two main directions have been followed to incorporate different global or context information for action representation [4]–[8]. First, much efforts have been put in enriching STIPs with high-level sematic information, *e.g.*, action bank [4], attributes [6], [7], and discriminative action parts [8]. These approaches aim at giving a robust representation for activity. Specifically, Sadanand and Corso [4] propose a novel high-level representation: *action bank*, to recognize human actions. It is comprised of many individual action detectors which provide high-level semantically rich features for action representation. Liu *et al.* [6] use high-level semantic concepts: *attributes* to recognize human action. Attributes are used to represent the complete pool of action classes, and they capture the inherent intra-class variations of each action class. Raptis *et al.* [8] incorporate appearance and motion constraints for the individual parts and the spatio-temporal dependencies among them to recognize human action. However, they cannot capture complex temporal structure for complex activities. Second, other approaches use Markov model, such as HMMs [9], semi-Markov Models [10], coupled HMMs [11] and coupled hidden semi-Markov models [12], to model the temporal statistical relations among different states of actions. In these methods, a human activity is represented by a sequence of elementary actions using the traditional Markov models. However, when a lot of unwanted background local features are extracted from videos, it makes the state variables of traditional Markov models unstable. In addition, traditional Markov models only encode temporal relations between each frame and its adjacent frames, or each video segment and its adjacent segments, while ignore long-range temporal relations among elementary actions.

In this paper, we propose a novel multiple-instance formulation for Markov model to overcome the aforementioned drawbacks. Our multiple-instance Markov model imposes weakly supervised information on a *bag* of Markov chains which represent a video using several sequences of elementary actions. The main idea is shown in Fig.1. This model imposes a constraint that at least one Markov chain can correctly represent human action using a sequence of elementary actions. These elementary actions are defined as the movements of large bodies, such as head, right arm, legs, and so on. One advantage of such definition is that it enables our method to select elementary actions with stable state variables, which is robust to unwanted background local features. Besides, the definition is different from traditional

Manuscript received March 19, 2013; revised December 3, 2013 and June 12, 2014; accepted July 20, 2014. Date of publication July 30, 2014; date of current version September 4, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Siome K. Goldenstein.

W. Zhou is with the Samsung Telecom Research and Development Center, Beijing 100190, China (e-mail: wen.zhou@ia.ac.cn).

Z. Zhang is with the College of Electron and Communication Engineering, Tianjin Normal University, Tianjin 300074, China (e-mail: zhong.zhang@ia.ac.cn).

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

Digital Object Identifier 10.1109/TIFS.2014.2344448

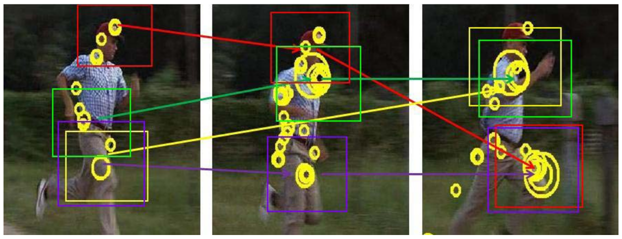

Fig. 1. An illustration of multiple-instance formulation for Markov model. Blocks denote sub-volumes which represent elementary actions, and different color arrows denote state transitions of Markov chains. We represent a video using a bag of multiple Markov chains as indicated by different colors.

Markov models which use whole movement of human (global video volumes) to represent elementary actions. It is more flexible than the elementary actions defined in traditional Markov models since it use the movements of large bodies (local video sub-volumes) to represent elementary actions.

Moreover, to generate multiple Markov chains, we build a temporal pyramid for image sequences to encode both local and long-range temporal information. In each layer of temporal pyramid, a sequence of STIPs is randomly chosen to encode one type of temporal relations, and sub-volumes centered at these STIPs are used to describe the appearance of the elementary actions. These sequences are combined to form a Markov chain. Thus, each Markov chain encodes both local and long-range temporal relations among elementary actions. Finally, since we formulate multiple chains in the framework of multiple-instance learning, our algorithm automatically selects elementary actions with stable state variables and discriminative Markov chains.

The rest of this paper is organized as follows. Sec.II describes related work. The proposed framework and saliency region detection are described in Sec. III. Details of our model, as well as model learning, and inference are elaborated in Sec.IV, V, and VI respectively. Experimental results are given in Sec.VII. We conclude this paper in Sec.VIII.

II. Related Work

The problem of action recognition has been widely explored in the computer vision community. Early action recognition methods focus on tracking and the analysis of trajectory. Recently, great progress has been made by introducing robust local descriptors: spatio-temporal features [1], [13], hierarchical invariant spatio-temporal features [3], motion and structure features [14], [15], and video representations: space-time pattern templates, trajectory-based representation [16], and bag-of-words representation [1], [17].

Bag-of-words models [1], [3], [13] are widely used in human action recognition due to their simplicity and effectiveness in action representation. However, it is beyond these methods' ability to model complex human activities as they ignore temporal relations among elementary actions.

For more complex human activities, temporal relations among elementary actions are discriminative to distinguish different human activities. Many methods have explored temporal relations, such as [12] and [18]. One major way for modeling temporal relations is using Markov models to capture the transitions between state variables [9]–[12], [19].

Oliver et al. [11] construct a coupled HMM to model human-human interactions, which addresses the limitation of the basic HMM. Since it can represent activities composed of motions of two or more agents, it successfully recognizes complex interactions between two persons.

The most relevant works are [10] and [20]. Both of these methods try to incorporate long-range dependencies. Sminchisescu et al. [20] use kernel principal component analysis to discover the intrinsic structure of the articulated action space, and they also explore factorial conditional random field for activity modeling and recognition. Factorial conditional random field models temporal sequences in multiple interacting ways. It relaxes independence assumption between observations and effectively incorporates both overlapping features and long-range dependencies. Shi et al. [10] propose a discriminative semi-Markov models for human action segmentation and recognition. It is an extension of HMM by allowing the underlying process to be a semi-Markov chain with a variable duration for each state. The modeling emphasis shifts towards individual segments as well as adjacent segments. Then, it incorporates both local and long-range information for human action recognition with discriminative manner. However, these models encode limit long-range information among elementary actions because they only consider states transition between adjacent segments. In addition, the state variables of these Markov models suffer from unwanted background STIPs. In contrast, on account of our instance generation for Markov model, our method encodes rich long-range information among elementary actions for action representation. Besides, since discriminative sub-volumes are selected to represent elementary actions because of the multiple-instance formulation for Markov model, the state variables of our algorithm are robust to unwanted background STIPs.

III. Proposed Framework

In this section, we present our framework for human action recognition. Fig.2 gives an overview of the proposed recognition framework. Given an input video, we first adopt interest point detector to detect STIPs. Then, salient region detection is applied to suppress unwanted background STIPs and obtain *valid* STIPs with high probability, where valid STIPs mean these STIPs represent local movements of objects. Following salient region detection, multiple Markov chains are generated as described in Sec.IV-A. After that, the parameters of our model are learned as described in Sec.V. Finally, given a test video, the input features are formulated as training process and the class label is inferred as described in Sec.VI.

A. Salient Region Detection

A lot of unwanted background STIPs are detected from complex videos due to background clutters and camera motion. Thus, we need generate huge instances to hit the valid instance, where valid instance denotes Markov chain that correctly represents human action. That will make our parameters

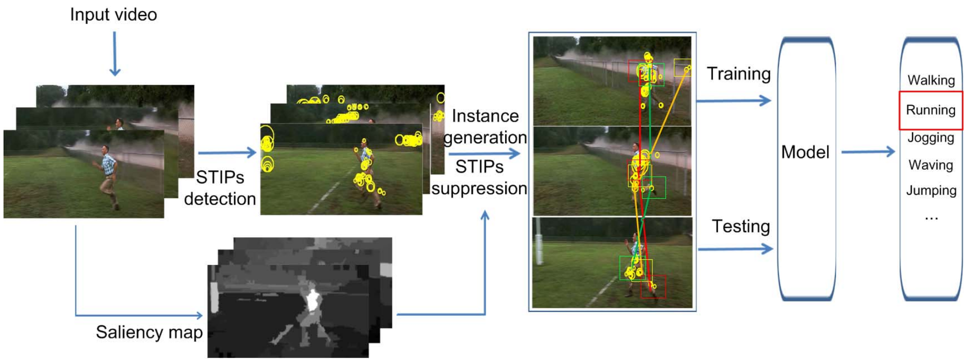

Fig. 2. Overview of proposed recognition framework.

learning inefficient. In order to prompt the efficiency of parameters learning, a saliency region detection method based on spatial prior is utilized to acquire saliency map that is used to suppress some unwanted background STIPs quickly.

We apply spatial prior based salient region detection to detect salient region in statistic image [21]. For an input video frame, we first segment it into several regions $\mathcal{R} = \{r_1, \ldots, r_c\}$. Then, the color histogram is built for each region to describe the appearance of region.

1) Region Contrast: We use spatially weighted region contrast [21] which incorporates spatial information by introducing a spatial weighting term to increase the effects of closer regions and decrease the effects of farther regions. As in [21], the spatially weighted region contrast based saliency of region r_k is defined as follows:

$$S(r_k) = \sum_{r_k \neq r_i} \exp(-D_s(r_k, r_i)/\sigma_s^2) w(r_i) D_r(r_k, r_i), \quad (1)$$

where $D_s(r_k, r_i)$ and $D_r(r_k, r_i)$ are the spatial distance and color distance between regions r_k and r_i respectively, and σ_s controls the strength of spatial weighting. $w(r_i)$ is the weight of region r_i which is defined as the number of pixels in r_i to emphasize color contrast to bigger regions.

2) Spatial Prior: Different from [21], we add spatial prior information to aid saliency region detection since we observe that actions are more likely to be spatially located in the center of video frames, especially for movie videos or videos that track objects. Then, the spatial prior is incorporated into (2). Our salient region detection is rewritten as follows:

$$S(r_k) = \sum_{r_k \neq r_i} \exp(-D_s(r_k, r_0)/\sigma_0^2) \exp(-D_s(r_k, r_i)/\sigma_s^2)$$
$$w(r_i) D_r(r_k, r_i), \quad (2)$$

where r_0 is the prior location of actions, and σ_0 controls the strength of spatial prior. We define $r_0 = [W/2, H/2]$, where W and H are the width and height of video frames respectively.

3) STIPs Suppression: As discussed before, due to background clutters and camera motion, many unwanted background STIPs are extracted from videos. Then, we need suppress STIPs to generate valid multiple instances. We suppress unwanted background STIPs using saliency maps S as follows: the STIP centered at (x, y) is selected as the final STIP if the saliency value at (x, y) is above the threshold T_h.

IV. Multiple-Instance Markov Model

A. Instance Generation

Instance generation aims at selecting elementary actions for multiple Markov chains to represent a video as shown in Fig.4. As mentioned before, sub-volume centered at a STIP is used to represent the elementary action. Specifically, we use the histogram of local features in a local sub-volume to represent the elementary action as shown in Fig.4. Then, our instance generation only needs to select sub-volumes to constitute Markov chains.

Given the i-th video $V^i (i = 1, .., N)$, STIPs $\{p_j^i\}_{j=1}^{n^i}$ are detected from this video, where n^i denotes the total number of STIPs. We use $\{t_j^i\}_{j=1}^{n^i}$ to denote the temporal location of these STIPs, and we use $B^i \in \mathcal{B}$ to represent a bag of Markov chains for the i-th video. B^i is constituted by several Markov chains $\{I_l^i | I_l^i \in B^i, l = 1, \ldots, L\}$, where L is the number of Markov chains. For clarity, we use $p(t)$ to denote the STIP p whose temporal location is t. Since the local sub-volume is centered at a STIP, we use p to denote the sub-volume centered at STIP p for notational clarity, and p is also used to denote the elementary action.

We first build temporal pyramid for image sequences as shown in Fig.3, where different layers of temporal pyramid give different temporal relations. Specifically, in layer 1, we use original sequence which only encodes temporal relations between elementary actions $p(t)$ and $p(t-1)$. Similarly, in layer 2, we select some sequences by down-sampling from original sequence with a factor of 1, which encode temporal relations between elementary actions $p(t)$ and $p(t-2)$. Down-sampling with a factor of 1 results in two different sub-sequences as shown in the left part and right part of layer 2. In the left and right part of layer 2, the $2{*}j$-th and $2{*}j + 1$-th video frames are selected to constitute sub-sequences respectively. In layer T, from the left to the right, each sub-sequence is constituted by the $T * j$, $T * j + 1, \ldots, T * j + T - 1$-th video frames respectively, which encodes temporal relations between elementary actions $p(t)$ and $p(t - T)$. For clarity, we use L_{ij} to denote the j-th sub-sequence in the i-th layer as shown in Fig.3.

The main reason for building temporal pyramid is to encode rich temporal information among elementary actions for action representation. After building temporal pyramid for videos, we build elementary actions pools for instance generation

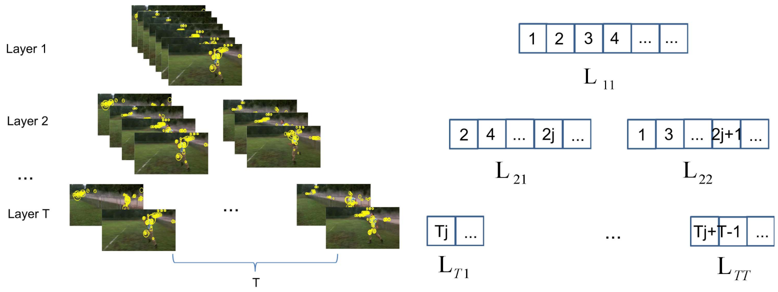

Fig. 3. An illustration of temporal pyramid. Each layer of temporal pyramid is constituted by a sequence of images, and different layers of temporal pyramid encode different temporal relations among elementary actions.

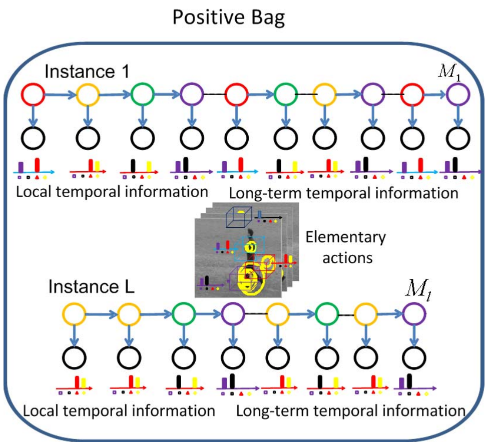
Positive Bag
Instance 1
M_1
Local temporal information
Long-term temporal information
Elementary actions
Instance L
M_l
Local temporal information
Long-term temporal information

Fig. 4. An illustration of positive bag for multiple-instance Markov model. Blue arrows denote the transition between states. Black lines denote fake transition between states since we combine STIPs from different layers of temporal pyramid into a chain structure, which do not affect the performance of our method due to the limited number $T-1$ ($M_l \gg T-1$).

where each pool contains T sub-sequences. Each sub-sequence is selected from one layer of temporal pyramid. For example, elementary actions pool can be constructed using all elementary actions from T sub-sequences $[L_{11}, L_{21}, \ldots, L_{T1}]$ or $[L_{11}, L_{22}, \ldots, L_{T1}]$. Then, each pool encodes both local and long-range temporal relations among elementary actions.

Finally, we choose elementary actions from these pools to generate instances. We randomly select one elementary action in each video frame from T sub-sequences, and combine these elementary actions into a chain structure. For example, instance $I = \{p(L_{11}), p(L_{21}), \ldots, p(L_{T1})\}$, where $p(L_{11})$ denotes elementary action sequence selected from images in sub-sequence L_{11}. $p(L_{11})$ encodes local temporal relations and $p(L_{T1})$ encodes long-range temporal relations. We can randomly generate instances several times to obtain more information for video representation. We use $\{I_l^i\}_{l=1}^L$ to denote L Markov chains that characterize i-th video.

B. Model Representation

Multiple-instance framework [22] is a generalization of supervised classification in which class labels are associated with sets of instances, in here, a set of Markov chains, instead of individual instance. In the case of binary classification, a bag is "positive" if at least one of its members is a positive example and it is "negative" if all its members are negative examples. Multiple-instance learning is widely used to handle the ambiguities of instances for different tasks, such as, classification of molecules in the context of drug design [22], content-based image retrieval [23], scene classification [24] and object tracking [25]. In this work, we will show that multiple-instance formulation for Markov model can be used to handle the ambiguities of temporal sequences.

Markov models have shown their advantages in exploring the sequential patterns, in here, statistical temporal relations, in the sequential data. We also use Markov model to encode statistical temporal relations and propose a multiple-instance formulation for Markov model to recognize human action. An illustration of positive bag for multiple-instance Markov model is shown in Fig.4. Each video is represented by a bag of Markov chains. Since we only know the label of bags, it is a weakly annotated data. Given the weakly annotated data, our model aims at exploring discriminative Markov chain from bags using maximum-bag-margin algorithm. For clarity, we first propose Markov model for one instance, and then provide Markov model for a bag of instances.

Note that our Markov model has several significant differences with traditional Hidden Markov Models (HMMs). First, traditional HMMs are used as full probabilistic models with unrestricted emission probabilities while our Markov model is a member of the exponential family with a log-linear model that is common in CRFs [26] and structural SVMs [27]. Besides, when traditional HMMs are used as full probabilistic models, they marginalize over the states as this makes inference less sensitive to noise. However, since our Markov model is learned and inferred in structural SVM, we perform inference by state maximization.

The goal of our Markov model for one instance is to find the label of instance by maximizing the probability of y_I given sequential data I and model's parameters $\mathbf{w}$, which is formulated as follows:

$$\begin{aligned} y_I &= \arg \max_{Y_I \in \{1,-1\}} \max_{\mathbf{s}} P(Y_I | I, \mathbf{w}) \\ &= \arg \max_{Y_I \in \{1,-1\}} \max_{\mathbf{s}} P(I | \mathbf{w}, Y_I) \\ &= \arg \max_{Y_I \in \{1,-1\}} \max_{\mathbf{s}} \sum_{k=1}^{M_l} \mathbf{w}^a_{s(t_{lk})} \Psi(p(t_{lk})) \\ &\quad + \sum_{i=1}^{K} \sum_{j=1}^{K} w^t_{ij} \Phi_{ij}(s(t_{l(k-1)}), s(t_{lk}), Y_I), \end{aligned} \tag{3}$$

where K ($K = |\mathcal{S}|$) is the number of state space, and it is also the number of cluster.

$\Phi_{ij}(s(t_{l(k-1)}), s(t_{lk}), Y_I)$ is a feature map of state transitions, and we define it as follows:

$$\Phi_{ij}(s(t_{l(k-1)}), s(t_{lk}), Y_I) = \mathbf{1}_{(s(t_{l(k-1)})=i)} \cdot \mathbf{1}_{(s(t_{lk})=j)} \cdot \mathbf{1}_{(Y_I=y_I)}, \quad (4)$$

where $\mathbf{1}_{(\cdot)}$ is an indicator function, e.g. $\mathbf{1}_{(s(t_{lk})=j)} = 1$ if $s(t_{lk})$ equals j, otherwise 0.

$\Psi(p(t_{lk}))$ characterizes the appearance of sub-volume centered at STIP $p(t_{lk})$. Specifically, it is a histogram of local STIPs extracted from the sub-volume. $\mathbf{w}^a_{s(t_{lk})}\Psi(p(t_{lk}))$ measures the similarity of the feature $\Psi(p(t_{lk}))$ to its assigned state $s(t_{lk})$.

Similar with Markov model for one instance, the Markov model for a bag of instances can be formulated as follows:

$$y_B = \max_{I \in B} y_I, \quad (5)$$

where B denotes a bag of instances.

V. Model Learning

In this section, we learn the parameters of multiple-instance Markov model using maximum-bag-margin algorithm. There are two sets of parameters that we must learn in our model, the appearance parameters $\mathbf{w}^a$, the transition parameters $\mathbf{w}^t$, which we can concatenate into a single weight vector:

$$\mathbf{w} = [\mathbf{w}^a \quad \mathbf{w}^t_1, \ldots, \quad \mathbf{w}^t_K], \quad (6)$$

where $\mathbf{w}^t_1 = [w^t_{11}, \ldots, w^t_{1K}]$. Similarly, we construct the feature vector $\Upsilon(I, \mathbf{s})$ for an instance as follows. For the $\mathbf{w}^a$ parameters we sum the feature histograms that are assigned to each state, and for the $\mathbf{w}^t$ parameters we count the number of times each state transition. We then normalize each of these features and concatenate them together to form the feature vector $\Upsilon(I, \mathbf{s})$.

Given a training set of N bags $\{B^i\}_{i=1}^N$ and their corresponding binary class labels $Y_{B^i} \in \{-1, 1\}$, we can compute their feature representations $\Upsilon(I^i_l, \mathbf{s})$. We consider linear classifier of the form:

$$f = \max_{I \in B^i} \max_{\mathbf{s}} \mathbf{w} \cdot \Upsilon(I, \mathbf{s}), \quad (7)$$

where $\mathbf{s}$ denotes the latent state variables. To the best of our knowledge, it is the first application of a generalized linear model to an HMM with unsupervised states and a bag of input sequences. Specifically, it is a multiple-instance formulation for Markov model that is a weakly-supervised framework for a bag of input sequences. That is based on an observation that it is easy to obtain the label of the bag while difficult to obtain the label for each input sequence in the bag.

We apply maximum-bag-margin algorithm and Latent SVM framework to learn the parameters $\mathbf{w}$. Generally, parameters learning leads to an iterative algorithm using Concave-Convex Procedure (CCCP) [28], where alternate between (**a**) optimizing the weight $\mathbf{w}$, and (**b**) inferring the latent state variables $\mathbf{s}$. This process is repeated for several iterations until convergence or maximum number of iterations is reached.

However, it is well known that latent SVM is a local minima learning algorithm and therefore initialization plays a fundamental role in performance. In order to avoid the local minimum, we initialize the latent state variables by performing k-means clustering algorithm on the elementary actions 10 times and keep the result with the lowest error. Specifically, we perform k-means clustering algorithm on the histogram of STIPs in a local sub-volume as shown in Fig. 1 to obtain the initial states of elementary actions, where sub-volumes centered at STIPs in $\{I^i_l\}_{l=1}^L$ are used to represent these elementary actions. We use $\mathbf{s} = \{s(t_{l1}), \ldots, s(t_{lk}) \ldots, s(t_{lM_l}) | s(t_{lk}) \in \mathcal{S}\}$ to denote the states of elementary actions, where M_l denotes the number of sub-volumes in the l-th instance, and t_{lk} denotes the temporal location of the k-th elementary action in the l-th instance.

a) Optimizing the weight: Once the latent variables are inferred, and the feature vectors $\Upsilon(I^i_l, \mathbf{s})$ are constructed for each examples. Then, optimizing the weight vector is formulated as follows:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{B^i} \xi_{B^i} \\ &\quad s.t. \quad \forall B^i : \quad Y_{B^i} \max_{I^i_l \in B^i} (\mathbf{w} \cdot \Upsilon(I^i_l, \mathbf{s})) \geq 1 - \xi_{B^i}, \\ &\qquad\qquad \exp(\mathbf{w}^t_k) \cdot \mathbf{1}^T = 1, \quad \xi_{B^i} \geq 0, \\ &\qquad\qquad k = 1, .., K, \quad i = 1, \ldots, N, \end{aligned} \quad (8)$$

where $\exp(\mathbf{w}^t_k) = [\exp(w^t_{k1}), \ldots, \exp(w^t_{kK})]$ and $\mathbf{1}^T = [1, \ldots, 1]^T$. In our experiments, we use the publicly available software CVX [29] which is a Matlab-based modeling system for convex optimization.

The first constraint states that positive bags should score better than 1, while negative bags score less than -1. The objective function penalizes violations of these constraints using slack variables ξ_{B^i}. Note that for a positive bag the margin is defined by the margin of the "most positive" instance, while the margin of a negative bag is defined by the "least negative" instance. The second constraint denotes the sum of $\exp(\mathbf{w}^t_k)$ should be one to guarantee that $\exp(\mathbf{w}^t_k)$ is valid transition probability. Parameter C controls the tradeoff between regularization and slack variables.

We also adopt [30] which employs heuristic learning algorithm. Their algorithm starts by an initialized SVM model, which is followed by a relabeling of the instances in positive bags using the learned model. If a positive bag contains no instances labeled as positive, then the instance with the largest score in that bag is relabeled as positive. The model is then retrained with the new labels. The process of relabeling and retraining is repeated until no labels are changed. However, this approach might often get trapped in local minimum, we randomly choose one instance for each bag several times, and initialize model with these instances to avoid local minimum. Algorithm 1 summarizes pseudo-code descriptions for the algorithms utilized in the experiments.

b) Inferring the latent state variables: Once the weight $\mathbf{w}$ is optimized, inferring the latent state variables $\mathbf{s}$ corresponds to performing MAP inference on the instance in Bag B^i to find the state. Due to the chain structure, exact maximum a posteriori inference for our model can be done efficiently

Algorithm 1: Pseudo-Code for Heuristic Multiple-Instance Learning Algorithm [30]

```
Input: $(\Upsilon_{B^i} = \{\Upsilon(I_l^i, \mathbf{s})\}_{l=1}^{L}, Y_{B^i}),\ i = 1, \ldots, N$
Output: $\boldsymbol{w}, b$
for $j=1:m$ do
    Random choose one instance $\Upsilon(I_l^i, \mathbf{s}) \in \Upsilon_{B^i}$ and compute QP solution $\mathbf{w}$ and b for dataset with positive examples $(\Upsilon_{B^i}, Y_{B^i})$.
    repeat
        compute outputs $f = \mathbf{w} \cdot \Upsilon(I_l^i, \mathbf{s})$ for all $\Upsilon(I_l^i, \mathbf{s})$ in positive bags, and set $\Upsilon(I_l^i, \mathbf{s}) = \Upsilon(I_k^i, \mathbf{s})$, where $k_i = \arg\max_{I_l^i \in B}(\mathbf{w} \cdot \Upsilon(I_l^i, \mathbf{s}))$,
    until no $k_i$ has changed during iteration;
end
```

Algorithm 2: Pseudo-Code for Parameters Learning

```
Input: $(\Upsilon_{B^i} = \{\Upsilon(I_l^i, \mathbf{s})\}_{l=1}^{L}, Y_{B^i}),\ i = 1, \ldots, N$
Initialize: $s = kmeans(\mathbf{x}, K)$
repeat
    optimize $\mathbf{w}$ and b using Algorithm 1 with $\Upsilon_{B^i}$;
    infer latent variables $\mathbf{s}$ for each instance in each $B^i$ using Viterbi algorithm;
    update $\Upsilon_{B^i}$ with the inferred $\mathbf{s}$.
until no $\mathbf{s}$ has changed during iteration;
```

using dynamic programming. Specifically, we perform Viterbi algorithm in logarithmic scale. For more details about Viterbi algorithm, we refer the reader to [31].

Algorithm 2 gives pseudo-code for parameters learning.

VI. Model Inference

In this section, we present the details of our model inference. Given STIPs $\{p_i^t\}_{i=1}^n$ extracted from test video $\mathcal{V}$, L instances $\{I_l\}_{l=1}^L$ are generated as defined in Section IV-A. For each instance, we compute the score of last elementary action and also infer the latent state variables $\mathbf{s}_l$ using Viterbi algorithm. $\Upsilon(I_l, \mathbf{s}_l)$ is formulated as training process. We define the inference process as follows:

$$y_B = sgn(\max_{I_l \in B} \mathbf{w} \cdot \Upsilon(I_l, \mathbf{s}_l)), \tag{9}$$

where $sgn(\cdot)$ is sign function.

1) Computational Complexity: For each possible state of elementary action, we need to enumerate $|\mathcal{S}|$ states of previous elementary action. Moreover, we need to compute $n_{max}L|\mathcal{S}|$ times at most, where n_{max} denotes the maximum number of elementary actions in L instances. Then, the runtime complexity for this inference algorithm is $\mathcal{O}(n_{max}L|\mathcal{S}|^2)$, where $|\mathcal{S}|$ is the number of states.

VII. Experiments

A. Primitive Features and Elementary Actions Representation

We adopt the STIP detector proposed in [32] to detect sparse 3D interest points from videos which denote the significant variation of cuboids. Similar with [1], we use HOG and HOF to characterize the appearance of these STIPs. We must point out that we apply k-means clustering two times with different purposes. For the first time, we apply k-means clustering with K' cluster centers, which is used to quantize local descriptor for STIPs. We simply use the assignments of STIPs to represent these STIPs. For the second time, we apply k-means clustering with $K(K \ll K')$ cluster centers to quantize histograms of elementary actions, which is used to initialize the states of elementary actions. Each elementary action is represented by a histogram of STIPs, where these STIPs are extracted from video volumes with size $w_v \times h_v \times t_v$ centered at a given STIP. In our experiments, w_v, h_v and t_v are set to 40, 40 and 10 respectively.

B. Datasets and Experimental Setup

1) KTH Actions Dataset: KTH actions dataset [33] contains six different actions. These video clips are acted under four different environments. We use the same experimental setup as [1]. All sequences are divided into a training set (8 persons), a validation set (8 persons) and a test set (9 persons). There are 2391 video clips in total, 1528 video clips for training and 863 video clips for testing. We use training set and validation set to train the model and present recognition accuracy on the test set. We apply one-against-all strategy for multi-class human action recognition.

2) UCF Sports Dataset: UCF sports dataset [34] contains 10 action categories. This dataset consists of a set of actions collected from various sports which are typically featured on broadcast television channels such as the BBC and ESPN. As in [2], we extend the dataset by adding a horizontally flipped version of each video. We use leave-one-out strategy for evaluation. This means that, for each clip in this dataset, we predict its label while training on all other clips, except for the flipped version of the tested video clip.

3) Youtube Actions Dataset: Youtube actions dataset [35] contains 11 action categories: basketball shooting, biking/ cycling, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. This dataset is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. The video clips in the same group may share some common features, such as the same actor, similar background, similar viewpoint, and so on. We use the experiment setting in [35], which takes part of the dataset (including all videos from biking and walking classes, and only videos from indexed 01 to 04 for the rest of classes) and obtain the average accuracy over 25-fold cross-validation.

4) Baselines: We use bag-of-words model coupled with linear SVM as our first baseline ($\mathbf{B}_1$) which ignores the temporal relations. We also use hidden Markov model ($\mathbf{B}_2$) which uses one instance to represent video and only encodes local temporal information. Both $\mathbf{B}_1$ and $\mathbf{B}_2$ use spatio-temporal features extracted from volumes centered at STIPs. These STIPs are selected using saliency map. We use comparison between $\mathbf{B}_1$ and $\mathbf{B}_1^*$ to demonstrate the advantage of STIPs suppression.

TABLE I

AVERAGE ACCURACY ON THE KTH ACTIONS DATASET

Method	Brief Description	Accuracy	Experimental Setting
[33]	SVM, local space time features	71.7%	split
[1]	SVM, spatio-temporal features	91.8%	split
[2]	SVM, dense spatio-temporal features	92.1%	split
[3]	SVM, learned hierarchical invariant spatio-temporal features	93.9%	split
[37]	Incremental learning, pyramid histograms of oriented gradient features	96.1%	split
[38]	Sparse modeling, dictionary learning, motion imagery	97.9%	split
[39]	Sparse representation, covariance manifolds, optical flow	97.4%	split
[4]	Action bank, linear SVM, spatiotemporal orientations	98.2%	split
[10]	Discriminative HMM, "cuboid" features	91.2%	split
[10]	Discriminative Semi-Markov Model, cutting plane learning, "cuboid" features	94.7%	split
[10]	BOW model, SVM, "cuboid" features	85.1%	split
[10]	Discriminative Semi-Markov Model, bundle method, "cuboid" features	95.0%	split
$\mathbf{B}_1$	BOW model, linear SVM, spatio-temporal features	91.4%	split
$\mathbf{B}_2$	Discriminative HMM, spatio-temporal features	92.3%	split
MIMM	Discriminative Multiple-instance Markov Model, "cuboid" features	98.0%	split
MIMM	Discriminative Multiple-instance Markov Model, spatio-temporal features	96.7%	split

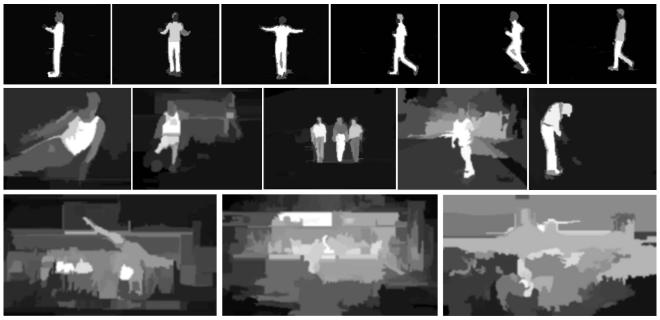

Fig. 5. Saliency maps for sample frames from KTH actions dataset and UCF sports dataset.

The only difference between $\mathbf{B}_1^*$ and $\mathbf{B}_1$ lies in that $\mathbf{B}_1^*$ uses original STIPs, while $\mathbf{B}_1$ use STIPs after suppression. To make direct comparisons with existing methods in literature presented in Table I, we also give experimental results by adopting a "cuboid" features [36] on KTH actions dataset. We use MIMM to represent multiple-instance Markov model for short.

We use cross validation to determine the optimal value of C. Specifically, we randomly divide the original dataset into two fold, and we perform two fold cross validation to search the optimal value of C where C range from 10^{-3} to 10^3. The optimal values of C are found at 10, 100 and 10 for KTH actions dataset, UCF sports dataset and Youtube actions dataset respectively.

C. Saliency Region Detection

Fig.5 gives the saliency maps on sample frames from KTH actions dataset (the first row) and UCF sports dataset (the second and third rows) respectively. Because of the clear background, global contrast based salient region detection gives the clear contour of human body. The second row of Fig.5 give fine saliency maps which clearly distinguish foreground and background. This is because of the large color histogram contrast between foreground and background. Although global contrast based salient region detection may fail to detect excellent saliency maps as shown in the third row of Fig.5, it reserves most STIPs due to our spatial prior for saliency region detection. We set $T_h = 0.5S_{max}$ where S_{max} is the maximum value of the saliency map.

D. Spatio-Temporal Interest Points Suppression

Fig.6 gives comparisons between original STIPs and STIPs after suppression on UCF sports dataset. A lot of unwanted background STIPs are detected from unconstraint videos especially when camera moves as shown in the first and third row of Fig.6. After suppression, many unwanted background STIPs are suppressed via saliency maps, as shown in the second and fourth row of Fig.6. It gives high probability of valid STIPs selection for instance generation where these valid STIPs represent the movement of objects, not background.

E. Experimental Results

1) KTH Actions Dataset: We found that STIPs after suppression make no difference with original STIPs on account of the clear background. Consequently, we use original STIPs to fairly compare with other methods on KTH actions dataset. The performance of our model on the KTH actions dataset is reported in Table I using the same experimental setting with other methods. First, we compare with our baselines $\mathbf{B}_1$ and $\mathbf{B}_2$. $\mathbf{B}_1$ ignores all temporal relations, while $\mathbf{B}_2$ only encodes local temporal relations among elementary actions. MIMM outperforms both $\mathbf{B}_1$ and $\mathbf{B}_2$. The main reason to obtain the improvement on KTH is that MIMM incorporates both local and long-range temporal relations among elementary actions. Castrodad and Sapiro [38] propose a sparse modeling pipeline for spatio-temporal features for human action recognition. Besides, they learn the inter-class correlations of actions to increase the discriminative power. Since they combine all the local information with the global information, they achieve 97.9% that is better than [1]–[3] which only use local features. Since our method also incorporate both local and long-range temporal information, our method achieves comparable result with [38]. Sadanand and Corso [4] propose a novel high-level action representation for human action recognition. Although they report a recognition accuracy of 98.2% on KTH actions dataset in [4], they use a richer representation for videos. Concretely, they first decompose the motion energies in different

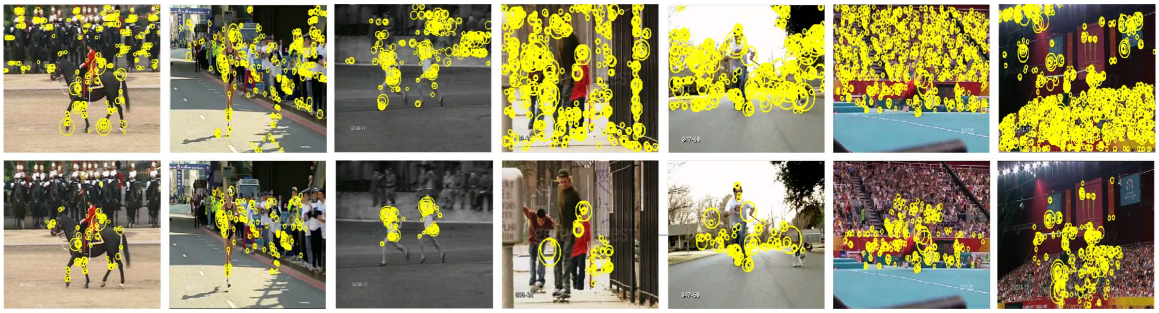

Fig. 6. Comparisons between original STIPs and STIPs after suppression on UCF sports dataset.

spatio-temporal orientations, which are realized using broadly tuned 3D Gaussian third derivative filters. Then, they correlate the action template with a query video to produce a correlation volume. Finally, they use volumetric max-pooling to extract a spatio-temporal feature vector for action representation. In a word, their low-level features are more powerful than ours. Besides, the action bank used for all experiments in [4] is the same that consists of 205 template actions collected from all 50 action classes in UCF50 and all six action classes from KTH. That means they use extra template actions (more training data) from UCF50 when they conduct experiments on KTH actions dataset. Thus, direct comparisons between these two methods would not be meaningful. The most related work is [10] which uses discriminative semi-Markov model to segment and recognize human action. Discriminative semi-Markov model also incorporates both local and long-range information. However, it encodes limit long-range information among elementary actions compared to MIMM. To fairly compare with [10], we also use "cuboid" features as primitive features to recognize human action. MIMM achieves superior results with [10] as shown in Table I. The main reason for this improvement is that MIMM encodes more long-range temporal information (temporal pyramid) for elementary actions to distinguish different actions than [10].

MIMM with "cuboid" features gives an improvement over MIMM with spatio-temporal features as it extracts more information from video using "cuboid" features than spatio-temporal features. [37] gives comparable results on KTH actions dataset compared to MIMM (spatio-temporal features). One reason is that [37] extracts pyramid histograms of oriented gradient features for shape of actor which is more discriminative than local features used in our method. However, they needs contour initialization as the lone preprocessing step which is difficult to obtain in realistic setting. Guo et al. [39] propose a sparse learning framework based on covariance manifolds of optical flow. Specifically, they capture action properties of an action segment by a covariance matrix. Then, they use a reconstruction residual error measure to decide the action class. Due to the effective action representation, they achieve 97.4%. However, since they use a global covariance matrix to represent actions, more subtle temporal information is ignored while our method captures that information using transition probability.

MIMM achieves 100%, 96.3%, 95.0%, 94% and 100% for boxing, hand clapping, hand waving, jogging, running and walking respectively. The major confusion occurs in running and jogging which shares similar STIPs due to the similar appearance. The model of "Jogging" is so strong that many videos with "Running" are wrongly classified into "Jogging".

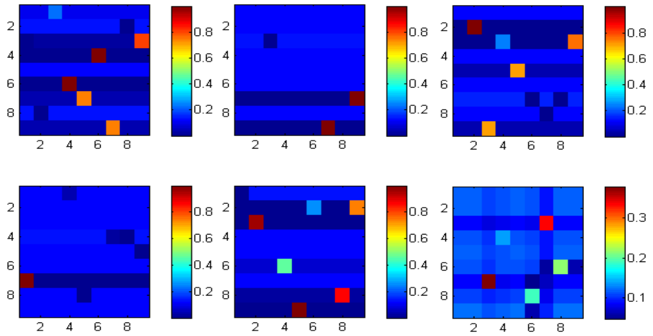

Fig. 7. Transition probability on KTH actions dataset for six classes. Red color indicates large transition probability in transition matrix, while blue color indicates low transition probability.

Fig.7 gives the transition probability on KTH actions dataset for boxing, hand clapping, hand waving, jogging, running and walking respectively, where the number of states K is set at 9. The element with i-th row and j-th col in transition matrix w^t_{ij} gives the probability that state j transferred from state i. From Fig.7 we can see that, these six transition matrixes are totally different on account of our discriminative parameters learning.

The recognition accuracy is a function of four parameters including: vocabulary size K', the number of states K, the layer number of temporal pyramid T, and the number of instances L. How these parameters affect the performance of our model? We perform four experiments to show the dependence of recognition accuracy on these four parameters.

Parameter K' varies from 600 to 4800, and K, T and L are set at 9, 4 and 40 respectively. Fig.8(a) gives the recognition accuracy of six action class with different K'. MIMM achieves the best mean recognition accuracy when $K' = 3000$. The mean recognition accuracy increases with K' when K' varies from 600 to 3000, and it slightly degrades when K' varies from 3000 to 4800. The main improvement of performance lies in "Running" in the first stage. One main reason for this improvement is that quantization for STIPs makes "Running" and "Jogging" easy to distinguish because of large K'. "Running" and "Jogging" contains many similar STIPs due to similar appearance of these two actions, and these STIPs are more likely to be quantized into the same cluster center when K' is small. However, when K' exceeds

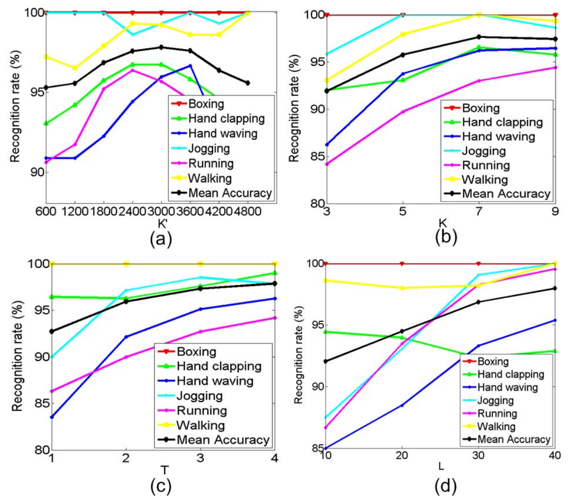

Fig. 8. (a)–(d) Give the recognition accuracies with different K', K, T and L respectively.

a value (here, 3000), STIPs may be over-quantized which results in representation with large variance. Then, it degrades the performance of MIMM in the second stage as shown in Fig.8(a). In a word, the recognition accuracy decreases with too small K' and too large K'.

Fig.8(b) shows the dependence of recognition accuracy on the number of states K, where K varies from 3 to 9, and K', T and L are set at 3000, 4 and 40 respectively. From Fig.8(b), the experiment results indicate that the recognition accuracy increases with K. The main reason for such improvement is that, MIMM encodes more temporal relations among elementary actions when K is large. As shown in Fig.7, only a small states have high transition probability by our discriminative learning. Thus, when K is large enough, the recognition accuracy does not increase.

To analyse the influence of T on recognition accuracy, we report the recognition accuracy with different T in Fig.8(c), where K', K and L are set at 3000, 9, and 40 respectively. The mean accuracy increases with rising T because of the incorporation of long-range temporal information. Incorporating long-range temporal information makes it easy to distinguish similar actions, such as "Jogging" and "Running", as indicated by Fig.8(c).

Fig.8(d) reports the recognition accuracy with different number of instances. From Fig.8(d) we can see that, the recognition accuracy increases with the number of instances as it results in high probability of hitting discriminative instances.

TABLE II

AVERAGE ACCURACY ON UCF SPORTS DATASET

Method	Accuracy	Experimental Setting
[2]	$78.1\%^{1}$	leave-one-out
[2]	$81.6\%^{2}$	leave-one-out
[2]	$85.6\%^{3}$	leave-one-out
[3]	86.8%	leave-one-out
[4]	95.0%	leave-one-out
[37]	87.1%	leave-one-out
[40]	88.2%	leave-one-out
MIMM	90.9%	leave-one-out

TABLE III

AVERAGE ACCURACY ON UCF SPORTS DATASET BY ACTION CLASS

Actions	$\mathbf{B}_1^*$	$\mathbf{B}_1$	$\mathbf{B}_2$	MIMM*	MIMM
Dive	92.9%	100%	100%	92%	100%
Golf swing	72.2%	80.4%	86.3%	85.3%	92.5%
Kick	100%	100%	100%	95%	100%
Lifting	33.3%	67%	73%	70%	78%
Ride horse	75%	80%	79%	80%	84%
Run	69%	73%	69%	80%	83%
Skateboard	75%	70%	67%	82%	81%
Swing-bench	85%	88%	90%	95%	100%
Swing-highbar	69%	72%	76%	85%	100%
Walk	81%	80%	85%	87%	91%
Average	75.2%	81%	82.5%	85.1%	90.9%

2) UCF Sports Dataset: Table II gives the recognition accuracy on UCF sports dataset and comparison with other methods under the same experimental setting. MIMM achieves better results than most of the methods reported in Table II. [2] and [3] use bag-of-words model coupled with SVM. The difference lies in that [2] extracts spatio-temporal features using dense sampling while [3] learns spatio-temporal features from data directly. For human action, discovering discriminative spatio-temporal features can prompt the performance without doubt, and discovering temporal information among elementary actions can prompt the performance further for complex activities. Consequently, MIMM gives superior performance than [2] and [3]. Different from [2], [3] which explore discriminative local features for action recognition, Sadanand and Corso [4] provide an action bank representation for human action recognition. Since action bank is robust to the large variation in actions, it achieves 95.0% that is better than our method. However, as discussed before, since they use extra template actions (more training data) from UCF50 when they conduct experiments on UCF actions dataset, it is unfair to directly compared with [4] in terms of recognition accuracy. However, our method has two main advantages compared to [4]. First, since the method described in [4] uses FFT-based convolution, it is computational inefficient compared to our method. Second, in order to obtain robustness for varied actions, the action bank detector used in [4] should cover the semantic space. Thus, action bank should be comprised of many individual action detectors sampled broadly in semantic space as well as viewpoint space. Then, one should carefully design the action bank detector. Thus, it heavily relies on manual intervention and it is difficult to generalize the action bank detector to more complex dataset. However, in our method, since both the weight for elementary actions and the transition probability are automatically learned from data, our method can be directly generalized to more complex dataset.

We also report the comparison between MIMM and baselines methods. Table III gives the recognition accuracy on UCF sports dataset by action class and also reports the comparison with our baseline methods $\mathbf{B}_1$ and $\mathbf{B}_2$, where K', K, T and L are set at 4000, 10, 5 and 60 respectively. From Table III we can see that, $\mathbf{B}_1$ performs better than $\mathbf{B}_1^*$ in mean accuracy due to the unwanted background STIPs suppression, which suppresses many unwanted background STIPs. $\mathbf{B}_2$ encodes local temporal information, and then it gives superior performance over $\mathbf{B}_1$. MIMM performs better than $\mathbf{B}_2$ since it not only encodes local temporal information,

TABLE IV

AVERAGE ACCURACY ON YOUTUBE ACTIONS DATASET BY ACTION CLASS

Actions	[35]	[3]	[40]	$\mathbf{B}_1^*$	$\mathbf{B}_1$	$\mathbf{B}_2$	MIMM
Cycle	73%	86.9%	91.7%	79.2%	88.6%	90.1%	93.8%
Dive	81%	93%	99%	80%	92%	90%	95%
Golf	86%	85%	97%	89%	84%	88%	87%
Juggle	54%	64%	76%	51%	60%	57%	70%
Jump	79%	87%	94%	76%	81%	85%	86%
Ride horse	72%	76%	85%	70%	75%	77%	82%
Basketball	53%	46.5%	43%	58%	55%	61%	65%
Volleyball	73.3%	81%	95%	73%	72%	71%	91%
Swing	57%	88%	71%	77%	70%	75%	87%
Tennis	80%	56%	88%	64%	66%	63%	76%
Walk	75%	78.1%	87%	55.4%	60.3%	61.5%	81.2%
Average	71.2%	76.5%	84.2%	70.2%	73.1%	74.4%	83.1%

but also long-range information. The only difference between MIMM* and MIMM lies in that, MIMM* use original STIPs while MIMM use STIPs after suppression. MIMM* gives superior performance than $\mathbf{B}_1^*$, $\mathbf{B}_1$ and $\mathbf{B}_2$. Due to the unwanted background STIPs, our instance generation may result in invalid instances which are constituted by unwanted background STIPs, and it degrades the performance of our method. Thus, MIMM* is outperformed by MIMM with large margin.

3) Youtube Action Dataset: Table IV gives average accuracy on Youtube action dataset by action class and comparisons with other methods and our baselines, where parameters are set the same as experiments on UCF sports dataset. $\mathbf{B}_1$ outperforms $\mathbf{B}_1^*$ mainly in cycling, diving, jumping, horse riding, and walking, where a lot of unwanted background STIPs are detected from unconstraint videos due to camera motion. The main reason is that $\mathbf{B}_1$ use valid STIPs after suppression, while $\mathbf{B}_1^*$ use original STIPs. Owing to the encoding of local temporal information, $\mathbf{B}_2$ achieves 74.4% and gives better performance than $\mathbf{B}_1$. Combining both local and long-range temporal information for elementary actions, the performance of MIMM is improved to 83.1%. [35] uses motion statistics to acquire stable motion features and clean static features, and [3] learns hierarchical invariant spatio-temporal features for human action recognition. Both of these methods ignore temporal relations and only rely on local features, and therefore MIMM gives superior performance than [3] and [35] because of the temporal information for mid-level features. We also report the result of [40] for comparison. In [40], Wang et al. use dense trajectories to represent human actions. Specifically, they sample dense points from each frame and track them based on displacement information from a dense optical flow field. Due to the rich trajectory information, they achieve 84.2% on Youtube actions dataset which is superior to our method. However, since they use BOW model to obtain the final representation for videos, they ignore the temporal information for action representation. Our method explores the temporal information of elementary actions for action recognition. We believe that the main improvement of [40] over MIMM is because of the strong local features. However, exploring the strong local features is beyond the scope of this work, and our method provides an effective way to incorporate complementary information (including temporal information and mid-level features) for local features.

VIII. CONCLUSION

In this paper, we propose a novel multiple-instance Markov model to overcome the disadvantages of traditional Markov model for human action recognition. First, a lot of unwanted background STIPs result in unstable state variables, and our multiple-instance formulation make our model select elementary actions with stable state variables. Second, our method gives a novel activity representation: a bag of Markov chains, which encodes both local and long-range temporal information among elementary actions. Our model explores the most discriminative Markov chain for action representation. Moreover, to efficiently generate valid instances, a spatial prior based saliency region detection method is proposed to suppress unwanted background STIPs. We evaluate our model on several actions datasets. Experimental results show the effectiveness of our model.

REFERENCES

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

[2] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 124.1–124.11.

[3] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3361–3368.

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

[5] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu, "Action recognition using context-constrained linear coding," *IEEE Signal Process. Lett.*, vol. 19, no. 7, pp. 439–442, Jul. 2012.

[6] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3337–3344.

[7] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu, "Attribute regularization based human action recognition," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 10, pp. 1600–1609, Oct. 2013.

[8] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1242–1249.

[9] A. F. Bobick and A. D. Wilson, "A state-based approach to the representation and recognition of gesture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1325–1337, Dec. 1997.

[10] Q. Shi, L. Cheng, L. Wang, and A. Smola, "Human action segmentation and recognition using discriminative semi-Markov models," *Int. J. Comput. Vis.*, vol. 93, no. 1, pp. 22–32, 2011.

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

[12] P. Natarajan and R. Nevatia, "Coupled hidden semi Markov models for activity recognition," in *Proc. WMVC*, Feb. 2007, p. 10.

[13] L. Shao and R. Mattivi, "Feature detector and descriptor evaluation in human action recognition," in *Proc. ACM Int. Conf. Image Video Retr.*, 2010, pp. 477–484.

[14] L. Shao, L. Ji, Y. Liu, and J. Zhang, "Human action segmentation and recognition via motion and shape analysis," *Pattern Recognit. Lett.*, vol. 33, no. 4, pp. 438–445, Mar. 2012.

[15] X. Zhen, L. Shao, D. Tao, and X. Li, "Embedding motion and structure features for human action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1182–1190, Jul. 2013.

[16] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," in *Proc. 11th Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 577–590.

[17] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi, "Cross-view action recognition via a continuous virtual path," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2690–2697.

[18] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1593–1600.

[19] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1250–1257.

[20] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2. Oct. 2005, pp. 1808–1815.

[21] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 409–416.

[22] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, Jan. 1997.

[23] O. Maron and A. Ratan, "Multiple-instance learning for natural scene classification," in *Proc. 15th ICML*, vol. 15. 1998, pp. 341–349.

[24] Z. H. Zhou and M. L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19. 2007, p. 1609.

[25] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 983–990.

[26] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2011, pp. 282–289.

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," in *Proc. J. Mach. Learn. Res.*, 2005, pp. 1453–1484.

[28] A. L. Yuille, A. Rangarajan, and A. Yuille, "The concave-convex procedure (CCCP)," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2. 2002, pp. 1033–1040.

[29] M. Grant and S. Boyd. (2014, Mar.). *CVX: Matlab Software for Disciplined Convex Programming, Version 2.1*. [Online]. Available: http://cvxr.com/cvx

[30] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 561–568.

[31] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[32] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2, pp. 107–123, Sep. 2005.

[33] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th ICPR*, vol. 3. Aug. 2004, pp. 32–36.

[34] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[35] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos 'in the wild'," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1996–2003.

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

[37] R. Minhas, A. A. Mohammed, and Q. M. J. Wu, "Incremental learning in human action recognition based on snippets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 11, pp. 1529–1541, Nov. 2011.

[38] A. Castrodad and G. Sapiro, "Sparse modeling of human actions from motion imagery," *Int. J. Comput. Vis.*, vol. 100, no. 1, pp. 1–15, Oct. 2012.

[39] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2010, pp. 188–195.

[40] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3169–3176.

Wen Zhou received the B.S. degree in automation from Beijing Business and Technology University, Beijing, China, in 2009, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2014. In 2014, he joined the Samsung Telecom Research and Development Center, Beijing, as a Researcher. His current research interests include pattern recognition, computer vision, and machine learning.

Zhong Zhang received the B.E. (Hons.) degree from Harbin Engineering University, Harbin, China, in 2009, and the Ph.D. (Hons.) degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2014. In 2014, he joined Tianjin Normal University, Tianjin, China, where he is currently an Assistant Professor. He has authored more than 30 papers in the areas of computer vision and pattern recognition in the international journals and conferences, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the Computer Vision and Pattern Recognition Conference, the International Conference on Pattern Recognition, and the International Conference on Image Processing. His current research interests include pattern recognition, computer vision, and machine learning.