Int J Comput Vis (2016) 119:254–271
DOI 10.1007/s11263-015-0859-0

MoFAP: A Multi-level Representation for Action Recognition

Limin Wang[1] · **Yu Qiao**[2] · **Xiaoou Tang**[1]

Received: 16 June 2014 / Accepted: 21 September 2015 / Published online: 7 October 2015

Abstract This paper proposes a multi-level video representation by stacking the activations of motion features, atoms, and phrases (MoFAP). Motion features refer to those low-level local descriptors, while motion atoms and phrases can be viewed as mid-level "temporal parts". Motion atom is defined as an atomic part of action, and captures the motion information of video in a short temporal scale. Motion phrase is a temporal composite of multiple motion atoms defined with an AND/OR structure. It further enhances the discriminative capacity of motion atoms by incorporating temporal structure in a longer temporal scale. Specifically, we first design a discriminative clustering method to automatically discover a set of representative motion atoms. Then, we mine effective motion phrases with high discriminative and representative capacity in a bottom-up manner. Based on these basic units of motion features, atoms, and phrases, we construct a MoFAP network by stacking them layer by layer. This MoFAP network enables us to extract the effective representation of video data from different levels and scales. The separate representations from motion features, motion atoms, and motion phrases are concatenated as a whole one, called *Activation of MoFAP*. The effectiveness of this representation is demonstrated on four challenging datasets: Olympic Sports, UCF50, HMDB51, and UCF101. Experimental results show that our representation achieves the state-of-the-art performance on these datasets.

Keywords Action recognition · Motion Feature · Motion Atom · Motion Phrase

Communicated by Ivan Laptev, Josef Sivic, and Deva Ramanan.

✉ Limin Wang
07wanglimin@gmail.com

Yu Qiao
yu.qiao@siat.ac.cn

Xiaoou Tang
xtang@ie.cuhk.edu.hk

[1] Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong

[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

1 Introduction

Human action recognition is an important problem in the field of computer vision and recently has received extensive research interests (Aggarwal and Ryoo 2011; Forsyth et al. 2005; Turaga et al. 2008). State-of-the-art methods (Wang and Schmid 2013a; Wang et al. 2013a, 2015) have achieved satisfying performance for action recognition in videos recorded under constrained environment, such as the datasets of KTH (Schüldt et al. 2004) and Weizmann (Gorelick et al. 2007). However, human action is extremely complex in realistic scenarios, for instance, the datasets of Olympic Sports (Niebles et al. 2010), HMDB51 (Kuehne et al. 2011), and UCF101 (Soomro et al. 2012). First, videos from the same action class always exhibit large intra-class variations, caused by background clutter, viewpoint change, and scale difference. Furthermore, an action contains complex temporal structure, and it is composed of several atomic actions. Like many other problems, an effective *visual representation* is very crucial to deal with these problems in action recognition from videos.

In the past several years, a great amount of research works have been devoted to developing robust video representation. Among those works, the most popular is Bag of Visual Words (BoVW) model (Csurka et al. 2004; Sivic and Zisserman 2003) and its variants (Wang et al. 2012) with low-level

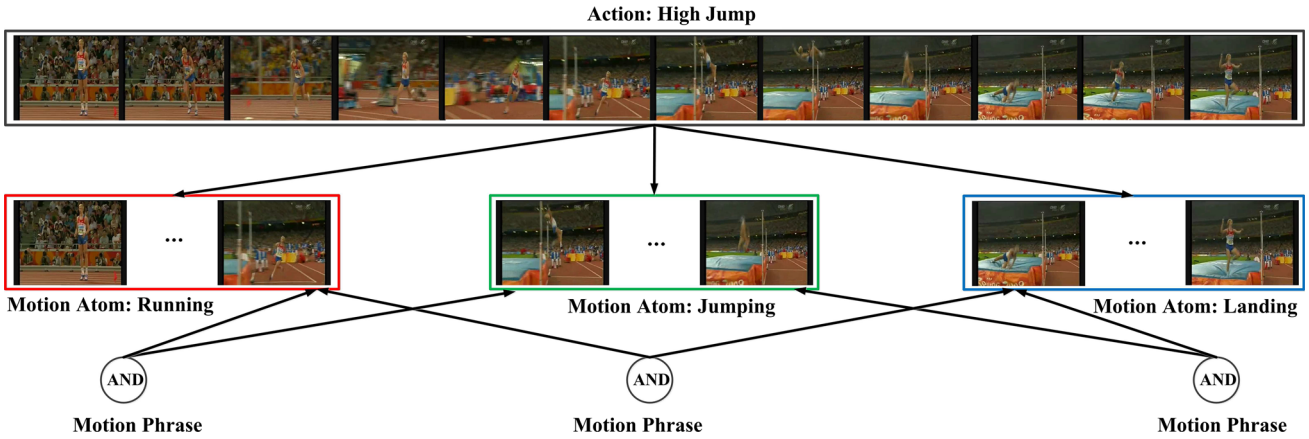

Fig. 1 Illustration of motion atoms and phrases. Actions can be decomposed into several motion atoms of short duration. For example of high jump, it contains atoms: running, jumping, and landing. Meanwhile, there exists temporal structure among multiple atoms in a long temporal scale, which can be described by motion phrase

features such as Space-Time Interest Points (STIPs) (Laptev 2005) and Dense Trajectories (DTs) (Wang et al. 2013a). Although these methods have achieve good performance in action recognition, there still exists a huge "semantic gap" between low-level video features and high-level action concepts. One way to bridge this semantic gap is to design sophisticated models by incorporating spatial and temporal relations among these low-level features, such as Temporal Structure Model (Niebles et al. 2010), Variable-duration HMM (Tang et al. 2012), Latent Hierarchical Model (Wang et al. 2014a), and Segmental Grammar Model (Pirsiavash and Ramanan 2014). Most of these statistical models resort to iterative algorithms to estimate model parameters and approximate inference techniques to speed up action recognition. In practice, however, these sophisticated models are less effective and efficient on the large-scale action recognition datasets.

Inspired by the recent works in image classification that build representation using mid-level "attributes" (Berg et al. 2010) or "parts" (Singh et al. 2012), we propose to discover "temporal parts" to represent action videos in this paper. These "temporal parts" are capable of modeling temporal structure for action understanding. Furthermore, compared with these complex models as described above, the proposed mid-level "temporal parts" share the following benefits: (i) it is efficient to compute the responses of mid-level feature units, and they can be used for action recognition on the large-scale datasets. (ii) the mid-level representation is independent with the final action recognition classifier and may be easily combined with other methods.

As shown in Fig. 1, our temporal part representation has two components: *motion atom* and *motion phrase*. Motion atoms aim to describe the simple and atomic motion patterns of short video segments. These motion atoms act as mid-level feature units to bridge the semantic gap between low-level features and high-level action concepts. To this end, we propose a discriminative clustering algorithm to discover a set of motion atoms from training videos. Specifically, after initialization step, our approach alternates between training classifier for each cluster and detecting top activations with this classifier. On convergence, each cluster corresponds to a motion atom and the classifier serves as atom detector. Motion phrase is a temporal composite of multiple motion atoms at different locations as shown in Fig. 1. A single motion atom describes the visual information in a short temporal scale, and thus its discriminative capacity is limited by its temporal duration. Motion phrase is expected to describe long-scale motion information by sequentially or hierarchically combining multiple motion atoms. Specifically, we adopt the AND/OR structure to define the temporal composition of multiple motion atoms at different locations. We design an efficient bottom-up mining algorithm and a greedy selection method to obtain a set of representative and discriminative motion phrases.

In oder to obtain an effective video representation, we stack the motion features, atoms, and phrases in a network manner as shown in Figure 3. Each layer outputs a video representation, corresponding to activation of **mo**tion **f**eatures, **a**toms, and **p**hrases, respectively. These multi-level representations are concatenated as the final video representation, called activations of **MoFAP**. MoFAP can be viewed as a hybrid representation, containing the low-level feature histogram, the mid-level motion atom activation, and even higher-level motion phrase activation. We observe that this representation is very effective in handling the complexity of realistic videos from our experimental results. Although the construction of motion atoms and phrases makes use of low-level features, they are able to describe the video data in different levels and extract extra visual information during their construction procedure. Therefore, these mid-level rep-

resentations are able to provide complementary information to low-level features. In summary, this work is composed of four contributions as follows:

- We propose a discriminative clustering algorithm to discover a set of motion atoms from unlabeled videos. These motion atoms acts as mid-level feature units to bridge the semantic gap between low-level features and high-level action concepts.
- We design motion phrases to further enhance the representative and discriminative capacity of motion atoms. These motion phrases are able to effectively deal with small temporal displacement and capture long-scale temporal structure.
- We present a multi-level video representation by stacking the encoding layers of motion features, atoms, and phrases. This hybrid representation aggregates the information from low-level to high-level cues, which are complementary to each other for the task of action recognition.
- We conduct experiments on four challenging datasets of the Olympic Sports (Niebles et al. 2010), the UCF50 (Reddy and Shah 2013), the HMDB51 (Kuehne et al. 2011), and the UCF101(Soomro et al. 2012). The experimental results demonstrate that our method achieves the state-of-the-art recognition performance.

This paper is an extension of our conference work (Wang et al. 2013b). The following describes the extensions of this paper from its conference version:

- We extend the single-scale motion atoms into multi-scale motion atoms, which can help address the issue of ambiguity in atomic actions. Meanwhile, this extension turns out to be effective in practice.
- Based on multi-scale motion atoms, we propose to mine not only temporal motion phrases, but also hierarchical motion phrases. These hierarchical motion phrases are complementary to the original temporal motion phrases.
- Based on motion features, atoms, and phrases, we design a new multi-level representation by stacking them in a network manner. This multi-level representation aggregates the information from low-level to high-level cues and is helpful to improve the recognition performance.
- We present new experimental results on two large datasets: HMDB51 and UCF101, and obtain the state-of-the-art performance. Furthermore, we conduct additional experiments to study different aspects of our method such as: (i) usage of new low-level representation (iDTs+FV); (ii) detailed and extensive exploration of the effectiveness of motion atoms and phrases; (iii) cross-dataset evaluation of motion atoms; (iv) performance evaluation on the multi-level MoFAP representation.

The remainder of this paper is organized as follows: Sect. 2 reviews related works on action recognition and mid-level representations. We present the unsupervised discovery of motion atoms in Sect. 3. Section 4 gives the description of mining motion phrases. We present the multi-level representation MoFAP in Sect. 5. The experimental evaluation and exploration is described in Sect. 6. Finally, we give a discussion and conclusion about our method in Sect. 7.

2 Related Work

Action recognition has been studied extensively in recent years and readers may refer to Aggarwal and Ryoo (2011); Forsyth et al. (2005); Turaga et al. (2008) for good surveys. Here, we only cover the works related to our method.

Mid-Level Representations in Action Mid-level representation such as parts, attributes and discriminative patches originated from image based tasks such as object recognition and scene classification (Berg et al. 2010; Bourdev and Malik 2009; Doersch et al. 2013; Parikh and Grauman 2011; Singh et al. 2012). Recently, these works have been extended to video domain and demonstrated its effectiveness in the task of action recognition (Jain et al. 2013a; Liu et al. 2011; Raptis et al. 2012; Sapienza et al. 2012; Wang et al. 2013c; Zhang et al. 2013; Zhu et al. 2013). Liu et al. (2011) first introduced the concept of attribute to represent video data for action recognition. They proposed a unified framework that utilizes both manually specified attributes and data driven attributes, and resorted to a latent SVM framework to explore the importance of different attributes. Sapienza et al. (2012) proposed to learn discriminative space time action parts in a multiple instance learning framework. Then they used a local deformable spatial BoVW to capture the spatiotemporal structure. Raptis et al. (2012) grouped similar trajectories into clusters, each of which was regarded as an action part. Then they used graphical model to capture both both information of each part and pairwise relations between parts. Jain et al. (2013a) extended the idea of discriminative patches into videos and proposed discriminative spatio-temporal patches for representing videos. Wang et al. (2013c) designed an action part called *motionlet*. Assuming motion is an important cue for action recognition, they proposed a data-driven approach to discover those effective parts with high motion salience. Zhang et al. (2013) proposed to discover a set of mid-level patches in a strongly-supervised manner. Similar to 2-D poselet (Bourdev and Malik 2009), they tightly clustered action parts using human joint labeling, called *acteme*. Zhu et al. (2013) proposed a two-layer *acton* representation for action recognition. The weakly-supervised actons were learned via a max-margin multi-channel multiple instance learning framework.

Our motion atoms and phrases can be viewed as "temporal parts" for representing action videos. However, there are several significant differences between our temporal parts and other mid-level parts. Motion atom aims to describe the visual pattern in a short temporal scale, and corresponds to a temporal atomic action. It focuses on the whole actor rather than a local spatial region. Meanwhile, motion phrase is a structured action part, where temporal relations among motion atoms is captured in an AND/OR structure. These previous mid-level action parts lack such structure information. Finally, we stack motion features, atoms, and phrase in a network manner to propose a multi-level representation.

Temporal Structure Actions, such as Olympic Sports actions (Niebles et al. 2010), and Cooking Composite actions (Rohrbach et al. 2012), can be usually decomposed into several atomic actions. Many previous research works have been proposed to model the *temporal structure* of segments for action recognition (Gaidon et al. 2013, 2014; Laxton et al. 2007; Niebles et al. 2010; Oliver et al. 2000; Pirsiavash and Ramanan 2014; Tang et al. 2012; Wang et al. 2014a, b, 2006). Many research works used state-observation sequential models, such as Hidden Markov Models (HMMs) (Oliver et al. 2000), Hidden Conditional Random Fields (HCRFs) (Wang et al. 2006), and Dynamic Bayesian Networks (DBNs) (Laxton et al. 2007), to model the temporal structure of action. These models are the probabilistic graphical models, where model parameters estimation and inference is conducted by some approximate methods, for example, Expectation Maximum Algorithm, Variational Methods, and Sampling Methods (Bishop 2006). Gaidon et al. (2013) annotated each atomic action for each video data and proposed Actom Sequence Model (ASM) for action detection. Niebles et al. (2010), and Tang et al. (2012) proposed to use latent variables to model the temporal decomposition of complex actions, and resorted to the Latent SVM (Felzenszwalb et al. 2010) to learn the model parameters in an iterative approach. Wang et al. (2014a) and Pirsiavash and Ramanan (2014) extended the temporal decomposition of complex action into a hierarchical manner by using Latent Hierarchical Model (LHM) and Segmental Grammar Model (SGM) respectively. These two methods aimed to capture the temporal structure of action in a coarse-to-fine way. Gaidon et al. (2014) introduced a spectral divisive clustering algorithm to extract a hierarchy over a large number of *tracklets*. Then, they used this structure to represent a video as an unordered binary tree for action recognition. Wang et al. (2014b) designed a sequential skeleton model (SSM) to capture the relations among *dynamic-poselets*, and performed action spatio-temporal detection from videos.

Our work is along the research line of using temporal structure as an effective cue for action understanding. However, we take a different perspective over this issue. We focus on learning a set of representation units, namely motion atoms and phrase, to represent video of complex action. This representation is flexible with the classifier used for recognition. Meanwhile, our representation is easily combined with other level features to boost final recognition performance.

AND/OR Model AND/OR structure has been successfully used in various vision tasks such as object recognition, image parsing, and action recognition (Amer et al. 2012; Chen et al. 2007; Si and Zhu 2013; Yao and Li 2010; Zhao and Zhu 2011). Chen et al. (2007) used the principle of summarization to specify a novel AND/OR graph representation for object parsing, which efficiently allows for all the different configurations of an articulated deformable object. Si and Zhu (2013) proposed a framework for unsupervised learning of a hierarchical AND/OR Template (AOT) for visual object recognition and detection. Zhao and Zhu (2011) designed a generative Stochastic Scene Grammar (SSG) by using AND, OR, and SET rules for scene parsing. Amer et al. (2012) introduced an AND/OR model to unify multi-scale action detection and recognition in a single framework.

The motivation of AND/OR structure in our motion phrase is different from these principled models. We aim to seek a flexible structure to allow both dealing with small temporal displacement and modeling long-scale temporal structure. Our idea is similar to the image representation of Grouplet (Yao and Li 2010). They designed a structure representation to capture the spatial relations among image patches for action recognition in still images. We extend this motivation into temporal domain to model the sequential structure of motion atoms for video-based action recognition.

3 Discovering Motion Atoms

In this section, our goal is to discover a set of motion atoms from multiple temporal scales by using unlabeled videos. These motion atoms aim to describe the motion patterns of different temporal durations (usually 20–100 frames) and they may server as effective mid-level "temporal parts" to represent the videos. However, this problem is very challenging due to the facts: (i) the number of possible short segments extracted from training videos is very huge. (ii) the videos for action recognition exhibit large intra-class variations caused by camera motions, viewpoint and scale changes. To handle these issues, we design a unsupervised discovery method by using discriminative clustering algorithms. We first describe the method of motion atom discovery from a single scale. Then, we extend this approach to find motion atoms from multiple scales.

Algorithm 1: Discovery of motion atoms.

Data: Training videos: $\mathcal{V} = \{V_n\}_{n=1}^{N}$.
Result: Motion atoms: $\mathcal{A} = \{A_m\}_{m=1}^{M}$.
// Initialization.
- $\mathcal{S} \leftarrow$ DenseSampling($\mathcal{V}$).
- $T \leftarrow$ CalculateSimilarity($\mathcal{S}$).
- $\mathcal{A} \leftarrow$ APCluster(T).
// Iteration Process.
while $t \leq Num$ **do**
  foreach *cluster* A_m *with* $size(A_m) > \tau$ **do**
    TrainSVM($A_m, \mathcal{V}$).
    FindTop($A_m, \mathcal{V}$).
  end
  CoverageCheck($\mathcal{A}, \mathcal{V}$).
  $t \leftarrow t + 1$.
end
- Return motion atoms: $\mathcal{A} = \{A_m\}_{m=1}^{M}$.

3.1 Discovery of Motion Atoms from a Single Scale

Inspired by a recent work on finding mid-level patches in images (Singh et al. 2012), we propose a discriminative clustering method for discovering motion atoms. However, the visual patterns contained in the action videos are more complex than those of natural images. We need take account of various aspects of video data such as static appearance, motion dynamics, and motion boundary. Thus, discovering effective motion atoms is more challenging than discovering mid-level patches from static images (Singh et al. 2012).

As shown in Algorithm 1, after initialization, our algorithm iterates between training discriminative classifiers and detecting top activations. Meanwhile, in order to ensure the representative capacity of discovered motion atoms, in each iteration, we propose a step to check the coverage of current discovered motion atoms, called *CoverageCheck*. We will give a detailed explanation of these steps in the remainder of this subsection.

Initialization The input to Algorithm 1 is a set of training videos $\mathcal{V} = \{V_n\}_{n=1}^{N}$, where V_n denotes the nth video clip and N is the total number of training videos. These video clips are with different action classes and have various temporal durations. As these clips are manually cropped from a continuous video stream, it is reasonable to assume that they are approximately aligned in temporal dimension. Based on this assumption, we design an average sampling strategy to obtain a set of short video segments and group them into clusters to initialize our iterative algorithm.

Specifically, we divide each training video into k short segments, where k is a parameter specified according to experimental explorations, and consecutive segments have half overlap with each other. For each short segment, we use Bag of Visual Words (BoVW) (Csurka et al. 2004; Sivic and Zisserman 2003) or Fisher vector (Sánchez et al. 2013) representation to describe the visual pattern. Based on the above analysis, unlike static images, videos need to be characterized from different visual aspects. So we extract multiple features for each short segment $\mathbf{F} = \{f^i\}_{i=1}^{I}$, where f^i is the feature histogram for ith view and I is the total number of views for video segment.

With these visual representations, a similarity measure between two segments $\mathbf{F}_m = \{f_m^i\}_{i=1}^{I}$ and $\mathbf{F}_n = \{f_n^i\}_{i=1}^{I}$ is defined as follows:

$$\mathcal{S}(\mathbf{F}_m, \mathbf{F}_n) = \sum_{i=1}^{I} \mathcal{K}(f_m^i, f_n^i), \tag{1}$$

where $\mathcal{K}$ is a kernel function defined over visual representations and its choice depends on the visual representation f. The details about the choice of low-level descriptors, encoding methods, and kernel function will be clarified in Sect. 5.

With this similarity measure, we use *Affinity Propagation* (AP) (Frey and Dueck 2007) to group similar segments into clusters. AP is an exemplar-based clustering algorithm whose input is a similarity matrix of input samples. This method is effective for discovering a group of similar segments and robust for input noise. It exchanges messages between data points until a good set of exemplars gradually emerges. The only parameter is the preference value. Due to the large variance of video segments, the preference value is set to be larger than the median of pairwise similarity to ensure that these segments are tightly clustered. For the initial clustering results, we set a threshold τ to eliminate the the clusters with small number of video segments (τ set to 4).

Iteration Process Given the initial clustering results, we iterate between SVM training and SVM detection. First, we train a SVM classifier for each cluster. The segments within the cluster are chosen as positive samples. For negative samples, we select the same number of segments with the lowest similarity to positive ones. With these positive and negative samples, we train a kernel SVM. The kernel function is chosen the same with Eq. (1). During SVM training, we set training parameter C as 0.01, which controls the balance between prediction loss and ℓ_2-regularization term.

After training the SVMs, we treat the classifier of each cluster as a detector and use it to scan over the training videos along temporal dimension. For each training video, we fix the length of scanning widow as $\frac{1}{k}d$ where k is equal to the number of divided segments in the initialization step, and d is the duration of video. We set the scanning step as 5 frames. Thanks to the additive property of histogram representation, we resort to the temporal integration histogram to speed up the scanning process. For each video clip, we construct its temporal integration histogram representation $\mathbf{H} = \{h_i\}_{i=1}^{d}$, where h_i denotes the integration histogram summarizing frames from 1 to i. Then, for a segment spec-

ified by starting frame s and ending frame e, its histogram representation f can be efficiently calculated as:

$$f(s,e) = \frac{1}{\|h_e - h_s\|}(h_e - h_s), \tag{2}$$

where $\|\cdot\|$ denotes a norm operation. This temporal integration histogram enables us to evaluate the detection score of sliding window efficiently. After scanning over all the training samples, we select top 10 video segments with the highest scores. Then based on these top detections, we retrain the SVM classifiers for all clusters.

Coverage Check To ensure the representative capacity of discovered motion atoms, we add a step of *CoverageCheck* in each iteration. We check the coverage percentage of current detection results and make sure that each training video has at least 3 segments detected as positive examples by the SVM classifiers. Otherwise, we will randomly extract segments from these clips and perform AP clustering in these newly-added segments. These clusters will be added into next iteration. This step is very crucial to gurantee the diversity of motion atoms and handle the complexity of action videos.

Finally, the whole iteration procedure is repeated until convergence, where we set the maximum of iteration number as 4. After the maximum iterations, these SVM classifiers $\mathcal{A} = \{A_m\}_{m=1}^{M}$ are the discovered motion atoms and we will introduce how to use them to represent the action videos in Sect. 5.

3.2 Extension to Multi-scale Motion Atoms

The above description about the discovery of motion atoms is based on a single temporal scale. However, as this discovery algorithm is an unsupervised learning method and there are no annotations for motion atoms in training samples, it is not easy to identify the temporal durations of motion atoms. To handle this issue, we propose a solution of discovering motion atoms from multiple temporal scales. We aim to discover motion atoms of different temporal durations (ranging from 20 to 100 frames) simultaneously.

Specifically, we first divide each video clips into $k_1, k_2, \ldots k_n$ segments. For each video clip, the resulting segments have various temporal durations and may correspond to motion atoms at different temporal scales. Then, we separately conduct initialization step for each temporal scale according to the description in previous section. Based on these initial clustering results, we iterate among SVM training, top detection and coverage checking for different temporal scales independently. Finally, we obtain a set of motion atoms from different temporal scales and these motion atoms may describe visual patterns of different temporal durations. We call these motion atoms discovered from different temporal scales as multi-scale motion atoms. Empirical results show that the proposed multi-scale extension is effective in improving the recognition performance.

4 Mining Motion Phrase

Motion atoms are based on clustering short video segments in an unsupervised manner. During the discovery process of motion atoms, the correlation between atoms and action categories is ignored. A motion atom may have strong activations in action videos with different categories. This fact may limit the discriminative capacity of motion atoms for the task of classifying actions. To circumvent this problem, we make use of these atoms as basic units to construct a structured representation, called *motion phrase*.

In this section, we will describe how to construct motion phrases based on motion atoms in a bottom-up manner. In order to achieve good performance in action recognition, motion phrases are expected to have following properties:

- *Descriptive property*: Each phrase should be a temporal composite of highly related motion atoms. It not only captures the appearance and motion information of each single motion atom, but also models the temporal and hierarchical structure between these motion atoms. Meanwhile, motion phrase should have a flexible structure, which allows for small temporal displacement caused by motion speed variations, but also encodes the co-occurrence of multiple motion atoms in a temporal and hierarchical manner.
- *Discriminative property*: To be effective in action classification, motion phrases should yield different levels of activations with respect to different action classes. It is desirable that a motion phrase is highly related to a certain class of action and it can distinguish this action class from others.
- *Representative property*: Due to large variations among complex action videos, each motion phrase can only activate with a small number of action videos. Thus, we need to take account of the correlations between different motion phrases and consider the complementarity between them. Ideally, a set of motion phrases are capable of conveying sufficient motion patterns to handle the variations of complex actions.

4.1 Motion Phrase Definition

Based on the analysis above, we resort to an AND/OR structure to define motion phrase as a temporal composite of multiple atom units as shown in Fig. 2. To begin with, we introduce some notations as follows.

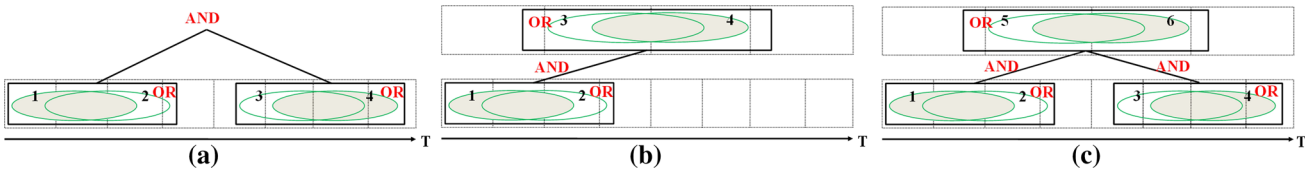

Fig. 2 Illustration of motion phrase. Motion phrase is an AND/OR structure over a set of atom units, which are indicated by ellipsoids. The motion atoms are discovered from multiple temporal scales. There are three types of motion phrases: **a** temporal motion phrase, **b** hierarchical motion phrase, **c** temporal and hierarchical motion phrase. Different types of motion phrases captures different kind of structure information contained in the action class

Each atom unit, denoted as $\Pi = (A, t, \sigma)$, refers to a motion atom A detected in the neighborhood of temporal anchor point t. The temporal extent of A in the neighborhood of t is modeled by a Gaussian distribution $\mathcal{N}(t'|t, \sigma)$. The response value v of an atom unit Π with respect to a given video V is defined as follows:

$$v(V, \Pi) = \max_{t' \in \Omega(t)} \text{Score}(f(V, t'), A) \cdot \mathcal{N}(t'|t, \sigma), \tag{3}$$

where $f(V, t')$ is the histogram representation extracted from video V at location t', $\text{Score}(f(V, t'), A)$ denotes the SVM output of motion atom A obtained in Sect. 3, and $\Omega(t)$ is the neighborhood extent over t. In current implementation, we fix the parameter σ as 0.5 for all atom units.

Based on these atom units, we construct motion phrases by AND/OR structure. We first apply *OR operation* over several atom units that have the same atom classifier and are located nearby (e.g. 1 and 2, 3 and 4 in Fig. 2). The atom unit that has the strongest response is selected by OR operation (e.g. 1 and 4 are selected in Fig. 2). Then, we perform *AND operation* over these selected atom units and choose the smallest response as motion phrase response. Therefore, the response value r of an motion phrase P with respect to a video V is given by:

$$r(V, P) = \min_{OR_i \in P} \max_{\Pi_j \in OR_i} v(V, \Pi_j), \tag{4}$$

where OR_i denotes the ith OR operation in motion phrase P. The size of motion phrase is defined as the number of OR operations it includes. For example, the sizes of atom phrase in Fig. 2a–c are 2, 2, and 3, respectively.

In essence, motion phrase representation is the temporal composite of multiple atomic motion units. The OR operation allows to search for the best location for the current motion atom, and makes it flexible to deal with the temporal displacement caused by speed variations. The AND operation captures the co-occurrence of several motion atoms from a long temporal scale. As motion atoms are discovered from multiple temporal scales, motion phrases may be defined over several scales. According to their modeling scales, the motion phrases may be generally classified into three types:

- *temporal motion phrase:* Motion phrase is composed of several motion atoms from the same temporal scale as shown in Fig. 2a. This kind of motion phrase is designed to capture the temporal relationship between the motion atoms. The temporal order among different motion atoms is an important cue for the understanding of complex actions.
- *hierarchical motion phrase:* Motion phrase contains multiple motion atoms from different temporal scales and there is only one OR operation in each temporal scale as shown in Fig. 2b. This kind of motion phrase models the hierarchical structure between the motion atoms from different temporal scales.
- *hierarchical and temporal motion phrase:* Motion phrase contains the motion atoms of different temporal scales and each temporal scale contains more than one OR operations as shown in Fig. 2c. This kind of motion phrase can be viewed as a combination of the other two types, which models both temporal and hierarchical structure.

In summary, motion phrase not only captures motion information of each motion atom, but also encodes temporal and hierarchical structure among them. This structured representation is able to enhance the descriptive capacity and make it more discriminative for complex action classification.

4.2 Evaluation of Discriminative Capacity

In order to mine a set of motion phrases for action recognition, we need to define their discriminative capacity. Intuitively, a motion phrase P is discriminative for the cth class of complex action if it has a strong activation with this class, but weakly activates with other action classes. Therefore, we define the discriminative capacity of motion phrase P with respect to class c as follows:

$$\text{Dis}(P, c) = \text{Rep}(P, c) - \max_{c_i \in C - c} \text{Rep}(P, c_i), \tag{5}$$

Algorithm 2: Mining motion phrases

Data: videos: $\mathcal{V} = \{V_n, y_n\}_{n=1}^{N}$, motion atoms: $\mathcal{A} = \{A_m\}_{m=1}^{M}$.
Result: Motion phrases: $\mathcal{P} = \{P_k\}_{k=1}^{K}$.
- Compute response value for each atom unit on all videos $v(V, \Pi)$ defined by Equation (3).
foreach *class c* **do**
  1. Select a subset of atom units (see Algorithm 3).
  2. Merge continuous atom units into 1-motion phrase $\mathcal{P}_1^c$.
  while $maxsize < MAX$ **do**
    a. Generate candidate s-motion phrase based on $(s-1)$-motion phrase.
    b. Select a subset of motion phrases $\mathcal{P}_s^c$ (see Algorithm 3).
  end
  3. Remove the motion phrase whose $\text{Dis}(P, c) < \tau$.
end
- Return motion phrases: $\mathcal{P} = \cup_{c,s}\mathcal{P}_s^c$.

where C represents all the classes and $\text{Rep}(P, c)$ denotes the representative capacity of P with respect to class c, for which the higher value indicates stronger correlation with the class c:

$$\text{Rep}(P, c) = \frac{\sum_{i \in S(P,c)} r(V_i, P)}{|S(P, c)|}, \tag{6}$$

where $r(V_i, P)$ denotes the response value of motion phrase P in video V_i calculated by Eq. (4), $S(P, c)$ is a set of videos defined as:

$$S(P, c) = \{i | Class(V_i) = c \wedge V_i \in top(P)\}, \tag{7}$$

where $Class(V_i)$ is the class label of video V_i and $top(P)$ represents the set of top videos that have the highest response values for motion phrase P. Due to the large variance among action videos, a single motion phrase P could have strong activations only with part of the videos of certain class. Thus, we evaluate its representative capacity by using a subset of videos of this class. In current implementation, we consider top 40 videos with strong activations for each motion phrase P.

4.3 Motion Phrase Mining

After the introduction to the definition of motion phrase and its corresponding evaluation of discriminative capacity, we are ready to propose the method to mine a set of effective motion phrases in this subsection.

As shown in Algorithm 2, the input is a set of videos with its corresponding labels $\mathcal{V} = \{V_n, y_n\}_{n=1}^{N}$, and a set of motion atoms $\mathcal{A} = \{A_m\}_{m=1}^{M}$, the output is a set of motion phrases $\mathcal{P} = \{P_k\}_{k=1}^{K}$. Given an action class c, our basic objective is to identify those motion phrases having high discriminative and representative capacity with current class. Furthermore, regarding a set of motion phrases $\mathcal{P} = \{P_k\}_{k=1}^{K}$, we also need to consider the correlation among them, and define the *set representative* capacity with respect to class c as follows:

$$\text{RepSet}(\mathcal{P}, c) = \frac{1}{T_c} | \cup_{P_k \in \mathcal{P}} S(P_k, c)|, \tag{8}$$

where T_c is the total number of training samples for class c, $S(P_i, c)$ is the video set defined in Eq. (7). Intuitively, considering the correlations of different motion phrases, it may help to eliminate the redundance and ensure the diversity of mining results. Besides, it can make sure that the mined motion phrases is able to handle the complexity of action videos.

The main challenge comes from the fact that the possible combinations of atom units that form motion phrases are huge. Assuming a video with k segments and the number of motion atoms is M, there are $M \times k$ possible atom units. Thus, the total number of possible motion phrase is approximately $\mathcal{O}(2^{M \times k})$. However, it is impossible to evaluate all possible configurations for mining motion phrase. We develop an efficient phrase mining method, inspired by Apriori algorithm (Agrawal and Srikant 1994). The main idea of Apriori mining is consistent with the AND operations in motion phrase: if a s-motion phrase has high representative capacity for action class c, where the size of motion phrase is defined as the number of OR operations, then any $(s-1)$-motion phrase should also have high representative capacity by removing any of its OR operation. Therefore, based on this observation, we are able to mine motion phrases efficiently in a bottom-up way as follows.

As shown in Algorithm 2, the OR operation is first used to obtain the 1-motion phrase. For each motion atom, a simple method is applied to merge nearby atom units that have strong representative capacity. Each merged result is then initialized as a 1-motion phrase. Then, during each iteration, we construct candidates of s-motion phrases based on $(s-1)$-motion phrases. To speed up the mining process, we need to identify a subset of motion phrases with high representative capacity. Ideally, both the individual and set representative capacity should be as high as possible. We design a greedy method to select effective phrases which is summarized in Algorithm 3. In each step, we select a motion phrase which not only has high representative capacity itself, but also increases the set representative capacity a lot.

For motion atoms from multiple temporal scales, we first construct temporal motion phrases for each single temporal scale independently as described above. Then, based on these mined temporal motion phrases, we further construct hierarchical motion phrases using the same method. In practice, we observe that the mined hierarchical motion phrases are complementary to temporal motion phrases and able to further boost recognition performance.

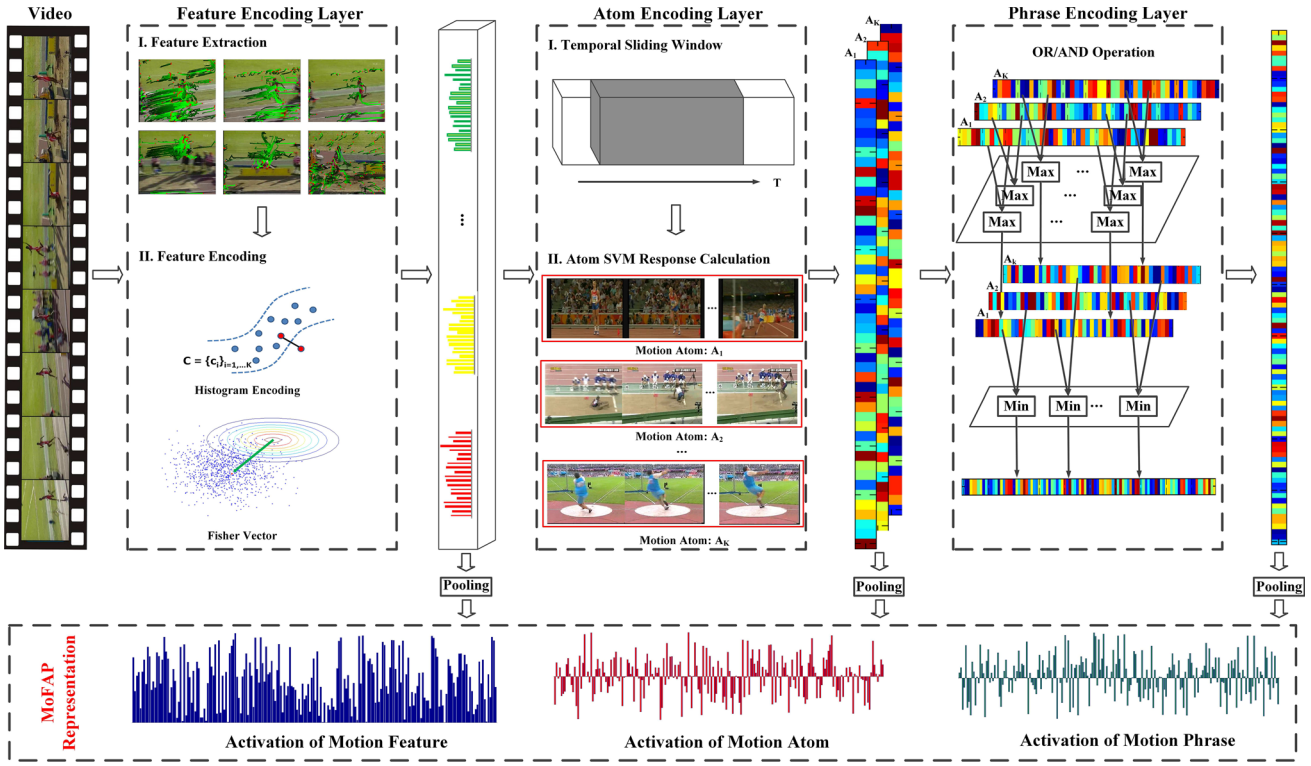

Fig. 3 Illustration of network architecture by stacking layers of motion feature encoding, motion atom encoding, and motion phrase encoding. In motion feature encoding layer, we extract low-level features and encode these descriptors using methods such as Histogram Encoding (HE) and Fisher Vector (FV). The resulting feature codes are simultaneously pooled as activation of motion feature and fed into next layer for further processing. In motion atom encoding layer, we conduct temporal sliding window scanning over the video and calculate the motion atom classifier responses. These response values are not only pooled as activation of motion atoms, but also passed into motion phrase layer. In the layer of motion phrase encoding, according to the AND/OR structure of mined motion phrases, we passed the response values of motion atoms through max and min layers to obtain the final activations of motion phrases. The resulting activations of motion feature, atoms, and atoms are concatenated as a multi-level representation, called activation of **MoFAP**

Algorithm 3: Selection of motion phrases.

Data: motion phrases candidates $\mathcal{P} = \{P_i\}_{i=1}^{L}$, class: c, number: K_c.
Result: selected motion phrases: $\mathcal{P}^* = \{P_i\}_{i=1}^{K_c}$.
- Compute the representative ability of each motion phrase $\text{Rep}(P, c)$ defined in Equation (6).
- Initialization: $n \leftarrow 0$, $\mathcal{P}^* \leftarrow \emptyset$.
while $n < K_c$ **do**
  1. For each remaining motion phrase P, compute: $\triangle\text{RepSet}(P, c) = \text{RepSet}(\mathcal{P} \cup P, c) - \text{RepSet}(\mathcal{P}, c)$, where $\text{RepSet}(\mathcal{P}, c)$ is defined in Equation (8).
  2. Choose the motion phrase: $P^* \leftarrow \arg\max_P [\text{Rep}(P, c) + \triangle\text{RepSet}(P, c)]$.
  3. Update: $n \leftarrow n + 1$, $\mathcal{P}^* \leftarrow \mathcal{P}^* \cup \{P^*\}$
end
- Return motion phrases: $\mathcal{P}^*$.

5 MoFAP: A Multi-level Representation

Now, we are ready to describe how to use the motion atoms and phrases as mid-level units to represent action video. Specifically, we first introduce the details about low-level visual features of a video segment. Then, we propose a multi-level representation MoFAP by stacking motion features, atoms, and phrases in a network manner, as shown in Fig. 3.

5.1 Segment Low-Level Representation

Local features such as Space Time Interest Points (Laptev 2005) and Dense Trajectories (Wang et al. 2013a) have turned out to be effective to capture the low-level visual information. The Dense Trajectories (Wang et al. 2013a) or its improved version (Wang and Schmid 2013a) with rich descriptors of HOG, HOF, MBHx, and MBHy have obtained the state-of-the-art performance on several challenging datasets. There are two typical choices about encoding methods and its corresponding similarity measure:

- Histogram encoding with an RBF-χ^2 kernel similarity measure (Wang et al. 2013a).
- Fisher vector encoding with a linear kernel similarity measure (Wang and Schmid 2013a).

These two choices are effective approaches that aggregate local descriptors into a holistic representation, which can be used to summarize the appearance and motion information for a short video segment. The detailed descriptions about them are out of the scope of this paper and can be found in their original papers (Wang and Schmid 2013a; Wang et al. 2013a). In experiments, we first conduct comparative study between these two kinds of low-level representations and figure out which are more effective for the construction of motion atoms and phrases. For feature fusion, we use kernel-level fusion to combine multiple descriptors, which means that we encode each descriptor independently, and the similarity measure is calculated as the summarization of multiple kernel matrices as defined in Eq. (1).

5.2 A Multi-level Representation

We have separately introduced motion feature encoding, motion atom discovery, and motion phrase mining in previous sections. These different feature units can be used to represent the video clip V for the task of action recognition. As shown in Fig. 3, we propose a multi-level representation, called MoFAP, to capture the visual information in a hierarchical manner.

In the layer of **motion feature encoding**, we extract low-level features, for instance, Improved Dense Trajectories (Wang and Schmid 2013a) with descriptors of HOG, HOF, MBHx, and MBHy. These different descriptors are independently encoded using Histogram Encoding or Fisher Vector as described in the previous subsection. Then these coding results are further pooled and normalized into a global representation R^f of the entire video clip, which is called *activation of motion features*. Meanwhile, these coding results are input into the layer of motion atom encoding for further processing.

In the layer of **motion atom encoding**, we calculate the response value $\text{Score}(f(V,t), A)$ of motion atom A for each location t defined in Eq. (3). Specifically, we conduct a temporal sliding window scanning and evaluate the classifier scores of the discovered motion atoms. In current implementation, the length of sliding window is determined according to the temporal scale of motion atoms and the scanning step is fixed as 5 frames. Finally, these motion atom response values are further processed with max pooling operation over the whole video clip:

$$r_A = \max_{t\in\{1,2,\ldots,T\}} \text{Score}(f(V,t), A). \tag{9}$$

Based on these pooled results of motion atoms, we obtain the global representation for the video clip by concatenating them together $R^a = [r_1, \ldots, r_M]$, which is called *activation of motion atom*. Besides, the response values of motion atoms are input into the next layer for computing the response values of motion phrases.

In the layer of **motion phrase encoding,** we compute the response value $r(V, P)$ for each mined motion phrase P according to the definition of Eq. (4). As shown in Fig. 3, these values can be efficiently calculated through a MIN and MAX layer sequentially. Finally, we obtain the global representation $R^p = [r(V, P_1), \ldots, r(V, P_C)]$ for video clip by concatenating the activation values of all the motion phrases, which is called *activation of motion phrase*.

Activations from different encoding layers may capture the visual patterns in different levels, and are complementary to each other. Based on this assumption, we propose a new representation, called *MoFAP*, by combing the activations of motion features, atoms, and phrases:

$$R = [R^f, R^a, R^p] \tag{10}$$

This representation turns out to be effective for improving action recognition performance in practice as shown in Sect. 6.

6 Experiments

In this section, we describe the detailed experimental settings and verify the performance of the proposed representation. In particular, we first introduce the datasets used for evaluation and their corresponding experimental setups. We then conduct experiments to study the effect of different low-level representations on the construction of mid-level motion atoms and phrases. After that, we investigate the performance of multi-scale motion atoms for different settings. Meanwhile, we study different configurations of motion phrase and figure out the best setting for the structure of motion phrase. In addition, we also conduct an experiment to evaluate the performance of motion atoms in a cross-dataset manner. Finally, we explore the effectiveness of MoFAP representation and compare with the state-of-the-art methods on four challenging action recognition datasets.

6.1 Datasets and Implementation Details

We conduct experiments on four datasets: the Olympic Sports dataset (Niebles et al. 2010), the UCF50 dataset (Reddy and Shah 2013), the HMDB51 dataset (Kuehne et al. 2011), and the UCF101 dataset (Soomro et al. 2012). These datasets contain video clips from YouTube and Movies, which are mainly captured in realistic scenarios and exhibit large intra-class variations.

The **Olympic Sports dataset** has 16 complex action classes such as high-jump, long-jump, and hammer-throw. This dataset includes 649 training videos and 134 testing

videos. We conduct experiments according to the setting specified on its project website.[1] The final recognition performance is evaluated by computing the average precision (AP) for each action class and reporting the mean of APs over all classes (mAP). The videos in this dataset belong to sport actions and exhibit complex temporal structure. Thus it is suitable for evaluating the effectiveness of motion atoms and phrases.

The **UCF50 dataset** has 50 action classes with a total of 6,618 videos. Videos in each action class are divided to 25 groups with at least 100 videos for each class. The video clips belonging to the same group are cropped from the same video and share similar background. We use the suggested evaluation protocol of Leave One Group Out cross validation (LOGO).[2] We report the average accuracy over the 25 cross validations.

The **HMDB51 dataset** has 51 action classes with 6,766 videos and each class has more than 100 videos.[3] All the videos are obtained from real world scenarios, such as Movies and YouTube. The intra-class variation is very high due to many factors, such as viewpoint, scale, background, illumination and so on. Thus, HMDB51 is a very difficult benchmark for action recognition. There are three training and testing splits released on the website of this dataset. We conduct experiments according to these splits and report average accuracy for evaluation.

The **UCF101 dataset** is an extension of the UCF50 dataset and has 101 action classes. It has 13,320 video clips, with resolution 320×240. The action classes can be divided into five types: human-object interaction, body motion only, human-human interaction, playing musical instruments, and sports. We perform experiments according to the three train/test splits posted on the website of THUMOS'13 Action Recognition Challenge (Jiang et al. 2013),[4] and report the mean average accuracy over these three splits.

In our evaluation experiment, we choose Support Vector Machine (SVM) as the classifier and employ the implementation of LIBSVM (Chang and Lin 2011). For multi-class classification, we adopt one-vs-all training scheme and choose the prediction with the highest score as our predicted label.

6.2 Exploration of Low-Level Representation

We begin our experiment by exploring different low-level representations. As described in Sect. 5.1, there are two popular choices in histogram representations and similarity measures, namely histogram encoding (HE) with a RBF-χ^2 kernel similarity measure (Wang et al. 2013a), and Fisher vector encoding (FV) with a linear kernel similarity measure (Wang and Schmid 2013a). Although it has been proved that Fisher vector encoding is more effective than histogram encoding for action recognition in previous research works, their performance on constructing mid-level temporal parts (motion atoms and phrases) remains unknown. In this exploration experiment, we choose the Olympic Sports dataset and the UCF50 dataset as they are representative in difficulty and size for the task of action recognition.

Specifically, for the first type of representation, we extract Dense Trajectories (DTs) with four descriptors: HOG, HOF, MBHx, and MBHy. For each type of descriptor, we train a codebook of size 1,000 using the randomly sampled 100,000 descriptors. Then we resort to histogram encoding (HE) to transform descriptor into the codeword space. For the second type representation, we extract Improved Dense Trajectories (iDTs) with the same four kinds of descriptors. Note that we do not employ human detector to help the estimation of camera motion as in (Wang and Schmid 2013a). We choose Fisher vector (FV) as encoding method, where the mixture number of GMMs is set as 256. To make the training of GMMs stable, we first conduct the pre-processing step of PCA and Whitening to de-correlate the each descriptor and reduce its dimension by a factor of two as suggested by (Sánchez et al. 2013). For pooling method, we choose sum pooling for both types of representations due to its good performance in previous research works. Following their original papers, we use the ℓ_1-normalization for histogram encoding and ℓ_2-power normalization for Fisher vector encoding.

In this exploration experiment, we divide each video clip into 5 segments for motion atom discovery and use single-scale atom representation. Therefore, for motion phrase mining, we only use temporal motion phrase for evaluation. The detailed exploration about different settings of motion atoms and phrases will be presented in the following subsections. The experimental results are shown in Table 1. As expected, the RBF-χ^2 kernel works better than the linear kernel for DTs+HE representation and non-linear kernel is able to bring around 10 % improvement on these datasets. It is also obvious that iDTs with Fisher vector is more effective than DTs with HE. For the MoFAP representation, iDTs+FV outperforms DTs+HE by 7–8 % on the two datasets. This performance improvement may be ascribed to the explicit camera motion estimation of iDTs and the rich representation of Fisher vector. To sum up, we will choose iDTs+FV as the low-level representation in the remaining experimental discussions.

6.3 Exploration of Multi-scale Motion Atoms

In this subsection, we study the effectiveness of motion atoms and explore the performance variations with different settings on the datasets of Olympic Sports and UCF50 .

[1] http://vision.stanford.edu/Datasets/OlympicSports/.

[2] http://crcv.ucf.edu/data/UCF50.php.

[3] http://serre-lab.clps.brown.edu/resources/HMDB/index.htm.

[4] http://crcv.ucf.edu/ICCV13-Action-Workshop/.

Table 1 Exploration of different low-level representations: DTs+HE and iDTs+FV, for the construction of motion atoms and phrases on the datasets of Olympic Sports and UCF50

Datasets	Olympic Sports dataset			UCF50 dataset		
Representation Kernel	DTs+HE linear kernel (%)	DTs+HE χ^2 kernel	iDTs+FV linear kernel (%)	DTs+HE linear kernel (%)	DTs+HE χ^2 kernel	iDTs+FV linear kernel (%)
Motion features	58.1	70.1	88.7	66.6	77.4	90.8
Motion atoms	69.1	76.1	83.4	73.1	82.5	90.7
Motion phrases	72.3	78.2	85.4	75.2	83.1	90.9
Motion atoms and phrases	74.1	79.5	86.9	76.4	84.0	91.1
MoFAP	77.3	84.9	91.9	77.2	85.7	93.1

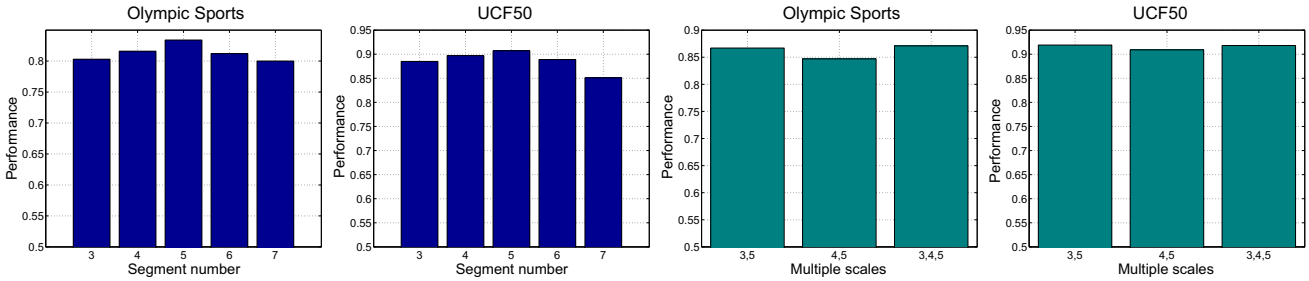

Fig. 4 Performance trend of motion atom activation with respect to the number of divided segments on the datasets of Olympic Sports and UCF50 (*first two figures*). We also study the effectiveness of multi-scale extension in motion atom discovery (*last two figures*)

For the discovery of motion atoms, an important parameter is the number of divided segments for each video clip. We vary the number from 3 to 7 and the experimental results are shown in Fig. 4. We observe that when segment number is set to 4, 5, 6, motion atoms obtain similar performance on both datasets, and number 5 is the best choice for motion atom discovery. When the number is set to 3 or 7, the performance is a bit lower. This is because that the extracted video segments may contain multiple motion primitives when the segment number is small such as 3. On the other hand, if the number is larger such as 7, it leads to over segmentation of video clip, and the divided video segment may be too short to contain any meaningful atomic motion. Thus, when choosing the number of divided segments, we need to keep a balance between under segmentation and over segmentation of video clip.

We also investigate the effectiveness of fusing the motion atoms discovered from multiple temporal scales and the experimental results are listed in Fig. 4. We first combine the motion atoms from two scales, such as 3 and 5 segments, 4 and 5 segments. Interestingly, although the setting of 4 segments outperforms that of 3 segments, its complementarity to 5 segments is smaller than that of 3 segments. This result may be ascribed to the fact when dividing video clips into 4 or 5 segments, it may be easy to discover very similar motion atoms, which may reduce the complementarity between them. We further explore the performance of motion atoms when combining more temporal scales such as 3,4,5, and we see that there is only a slight performance improvement over combination of two temporal scales. According to this observation, to keep a balance between efficiency and accuracy, we choose the motion atoms discovered from two temporal scales (segment numbers are 3 and 5) for representing videos in the rest of experiments.

6.4 Exploration of Motion Phrases

Having investigated different aspects of motion atoms, we now turn to analyze the properties of motion phrases. We mainly study the effect of phrase size on the recognition performance with the Olympic Sports dataset.

We first conduct experiments using the motion phrases mined from single-scale motion atoms (segment number is 5) and the results are shown in Fig. 5. As described in Sect. 4, the size of a motion phrase is defined as the number of OR operations. In order to compare the performance of motion phrase with motion atom, we regard the motion atoms as 0-motion phrases.[5] As shown in Fig. 5, motion phrase size varies from 0 to 4 and we observe that 1-motion phrases achieve the highest performance. Meanwhile, the 0-motion phrases and 2-motion phrases obtain similar accuracy. However, when increasing the phrase size to 3 or 4, there is a 6–9 % decrease in the performance of action recognition. This result

[5] Here we use the notation of #-motion phrase to represent motion phrase of size #.

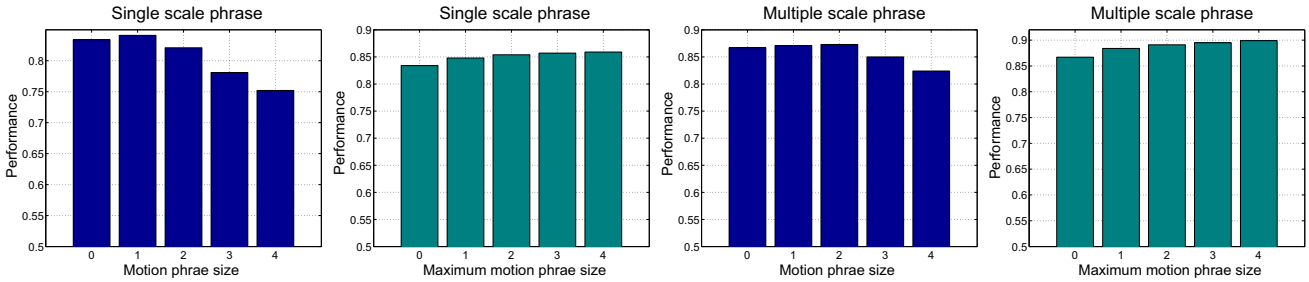

Fig. 5 Exploration of the effect of motion phrase size on the Olympic Sports dataset. We first conduct experiments using the motion phrases from a single scale (*first two figures*). Then, we investigate the motion phrases mined from the multiple scales and verify the effectiveness hierarchical motion phrases (*last two figures*)

may be explained by large intra-class variations contained by action videos. Although motion phrases of larger size are more discriminative, they can only handle a small portion of the complex videos. Thus, their representative capacity may be relatively low compared with motion phrase of smaller sizes.

In the next, we explore the performance of combining motion phrases with different sizes and the results are summarized in Fig. 5. We see that the performance increases apparently when using motion phrases from size 0–2. But the accuracy stabilizes when including motion phrases of larger sizes such as 3 and 4. Two reasons may explain this result: (i) As stated in previous analysis, motion phrases of larger sizes may lack representative capacity to deal with the complexity of video data. (ii) Typically the useful information delivered by the motion phrases of larger sizes may also be contained in the motion phrases of smaller sizes. Thus, increasing the size of motion phrase may not add much complementary visual cues for action understanding. Therefore, to make a trade-off between efficiency and accuracy, we will fix the maximum size for the single-scale motion phrase as 2.

After finishing the exploration of motion phrases mined from a single scale, we finally investigate the effectiveness of mining motion phrases from multiple temporal scales. We choose the motion atoms from two scales (segment numbers are 3 and 5). We first independently mine motion phrases for each temporal scale and the maximum size is set as 2. Then, based on these temporal motion phrases from two different scales, we construct hierarchical motion phrases. The results are shown in Fig. 5 with different maximum sizes of hierarchical and temporal motion phrases. We see that mining motion phrases from two temporal scales is able to further boost the recognition performance by around 4%. This improvement implies the importance of modeling hierarchical structure for action recognition, which agrees the conclusions of Wang et al. (2014a) and Pirsiavash and Ramanan (2014). To sum up, in the following explorations, we will mine motion phrases from two temporal scales with maximum size as 2 for temporal motion phrases and maximum size as 4 for hierarchical and temporal motion phrases.

6.5 Effectiveness of MoFAP Representation

We have separately studied different aspects of motion features, atoms, and phrases in previous subsections. Here we will evaluate the effectiveness of the proposed multi-level MoFAP representation in this section. For MoFAP representation, we will combine the activation values of motion features, atoms, and phrases. We use a simple fusion method by averaging their SVM scores. We conducted experiments on four challenging datasets: (i) the Olympic Sports dataset, (ii) the UCF50 dataset, (iii) the HMDB51 dataset, and (iv) the UCF101 dataset.

The results are summarized in Table 2. From these results, we see that activations of motion atoms outperform that of

Table 2 Performance evaluation and comparison on the four datasets

Representation	Olympic Sports (%)	UCF50 (%)	HMDB51 (%)	UCF101 (%)
Activation of motion features	88.7	90.8	57.2	84.4
Activation of motion atoms	86.7	91.9	58.9	85.5
Activation of phrases	89.9	92.2	59.5	85.7
Activation of atoms and phrases	90.2	92.7	60.5	86.5
Activation of MoFAP	92.6	93.8	61.7	88.3

We first compare the recognition performance of different layers: motion features, motion atoms, and motion phrases. We then explore the complementarity of motion atoms and phrases. Finally, we verify the effectiveness of the proposed MoFAP representation

Table 3 Exploration of the effectiveness of motion atoms in the cross-dataset setting

Olympic Sports dataset			UCF101 Sports dataset		
Original dataset (%)	Cross dataset (%)	Combination (%)	Original dataset (%)	Cross dataset (%)	Combination (%)
86.7	88.2	89.3	90.7	64.6	91.2

Original dataset means using the motion atoms discovered from its original dataset. Cross dataset means using the motion atoms discovered from another dataset. Combination indicates using motion atoms discovered from both datasets

motion features on these datasets except the Olympic Sports dataset. One possible explanation is that the Olympic Sports dataset is relatively small compared with the other three datasets. We discover about 8,000–15,000 motion atoms from these large datasets. However, for the Olympic Sports dataset, we only discover around 1,000 motion atoms. It should be noted that for Fisher vector, the dimension is $2 \times 256 \times 198 \approx 100,000$, which is much higher than that of atom activations.

Motion phrases obtain higher performance than motion atoms on the four datasets, due to their rich descriptive power and high discriminative capacity. We also see that the improvement is around 3 % on the dataset of Olympic Sports, which is more evident than on the other datasets. Olympic Sports dataset is composed of complex action classes, such as high-jump and long-jump, which can be temporally decomposed into several primitive actions, such as waiting, running, and jumping. Motion phrases are more suitable to describe these composite actions and model the temporal and hierarchical structure contained by them.

The combination of motion atoms and phrases further improves the recognition performance of mid-level representations by around 1 %. For MoFAP representation, it obtains the highest recognition accuracy on the four datasets. It improves over motion atoms and phrases by about 1–2 %, and over motion features by about 3–4 %. The best performance of MoFAP may be ascribed to the complementarity of activation values from different layers. Although the construction of motion atoms and phrases makes use of motion features, they may extract new visual information and are able to capture more complex patterns than low-level features.

Computational Cost of MoFAP Representation The training process for discovering motion atoms takes time due to its iterations over large datasets. For the UCF101 dataset, it requires about 2 days on a workstation with 8 cores CPU and 48G RAM. The mining process of motion phrases is more efficient due to Apriori-alike algorithm and it takes about 5 h for the UCF101 dataset. For testing, the extraction of MoFAP representation is very efficient. Once we have extracted iDT features and constructed temporal integration histogram, the computation of activations of motion atoms and phrases only involves a matrix multiplication and MIN/MAX operations. In addition to motion feature extraction, it usually takes about extra 5 s to calculate MoFAP representation for one video clip.

6.6 Cross-Dataset Evaluation

After verifying the effectiveness of MoFAP representations, we now evaluate the performance of motion atoms in a cross-dataset manner. Cross-dataset evaluation is much more challenging than testing on the original dataset, and it is very helpful to investigate the generalization ability of motion atom. Specifically, we choose two related datasets: the Olympic Sports dataset and the Sports dataset of UCF101. Although these two datasets are both about sports, the diversity of the UCF101 Sports dataset is much higher than that of the Olympic Sports dataset. First, the number of action class in the Olympic Sports dataset is much less than that of the Sports dataset (16 vs. 50). The action classes in the Olympic Sports dataset are all related to the Olympic Games. However, the range of action classes in the UCF101 Sports dataset is much broader. It contains several daily sport classes, such as biking, horse-riding, and boxing. Meanwhile, the number of videos in the UCF101 Sports dataset is 6,673, while the Olympic Sports dataset only contains 783 videos. More videos also add the complexity of the dataset of UCF101 Sports.

The experimental results are shown in Table 3. We list the recognition results of using motion atoms discovered from the original dataset and cross dataset. On the dataset of Olympic Sports, the motion atoms discovered from UCF101 sports dataset outperform the original motion atoms by around 1.5 %. However, on the UCF101 Sports dataset, the motion atoms of cross dataset perform much worse and bring about 25 % decrease in recognition accuracy. This result can be ascribed to the fact that the complexity of UCF101 Sports dataset is much higher than that of the Olympic Sports dataset. Thus, the representative capacity of motion atoms discovered from the Olympic Sports dataset is insufficient for handling the diversity of UCF101 Sports dataset. Finally, we also conduct experiments by combining the motion atoms from both datasets and we see that this combination brings around 1 % improvement on both datasets. This result indicates the motion atoms from two different datasets contain complementary information to each other.

6.7 Comparison to the state of the art

In this subsection, we compare the recognition performance of MoFAP representation to that of the state-of-the-art meth-

Table 4 We compare the performance of MoFAP representation with the state-of-the-art methods on the four challenging dataset: Olympic Sports, UCF50, HMDB51, and UCF101

Olympic Sports	mAP (%)	UCF50	Accuracy (%)
Tang et al. (2012)	66.8	Kliper-Gross et al. (2012)	72.7
Wang et al. (2013a)	77.2	Sadanand and Corso (2012)	57.9
Wang et al. (2014a)	85.2	Wang et al. (2013c)	78.4
Gaidon et al. (2014)	85.5	Wang et al. (2013a)	85.6
Wang and Schmid (2013a)	91.1	Wang and Schmid (2013a)	91.2
Activation of MoFAP	92.6	Activation of MoFAP	93.8
HMDB51	Accuracy (%)	UCF101	Accuracy (%)
Jiang et al. (2012)	40.7	Karpathy et al. (2014)	63.3
Wang et al. (2013c)	42.1	Simonyan and Zisserman (2014)	88.0
Wang et al. (2013a)	46.6	Cai et al. (2014)	83.5
Jain et al. (2013b)	52.1	Wu et al. (2014)	84.2
Wang and Schmid (2013a)	57.2	Wang and Schmid (2013b)	85.9
Activation of MoFAP	61.7	Activation of MoFAP	88.3

The effectiveness of MoFAP representation is demonstrated by its superior performance on these dataset

ods on the four challenging datasets. The detailed comparison results are shown in Table 4.

For the Olympic Sports dataset, we first compare our representation with two temporal models. The methods in Tang et al. (2012) and Wang et al. (2014a) both resort to use latent variables to decompose the complex action clips into segments in a sequential and hierarchical manner. They learn a single decomposition model for each complex action using iterative algorithms. The superior performance of our representation indicates that the mid-level representation is able to better handle the large complexity of videos than a single action model. We also compare our representation with other mid-level representations such as Tracklets (Gaidon et al. 2014). The superior performance to these methods may be due to fact that MoFAP is a rich representation, which utilizes multi-level and multi-scale information. Finally, we compare our performance with the best results (Wang and Schmid 2013a) and MoFAP outperforms it by 1.5%. It should be noted that they used the sophisticated human detection and tracking techniques to extract improved dense trajectories, while our low-level features extraction does not require these complex operations. Our multi-level representation achieves the best result on the Olympic Sports dataset and shows its effectiveness for the recognition of complex sports action.

For the UCF50 dataset, our representation is first compared with several low-level features, such as Motion Interchange Patterns (Kliper-Gross et al. 2012) and Dense Trajectories (Wang et al. 2013a). We see that our multi-level representation clearly outperforms these low-level representations on the UCF50 dataset. These low-level features only describe a small spatio-temporal region while our representation captures the visual information from multiple scales with longer context. Action Bank (Sadanand and Corso 2012) is a global template to represent the action videos and is not effective for dealing with the large intra-class variations. Unlike action bank, our motion atoms and phrases correspond to mid-level "parts" of the action, similar to the mid-level Motionlets (Wang et al. 2013c). They make a good tradeoff between low-level features and global templates. However, Motionlets are limited in temporal domain and lack descriptive power for longer temporal structure. Finally, we compare our performance with the best result (Wang and Schmid 2013a), and our representation outperforms it by 2.6 %.

For the HMDB51 dataset, Jiang et al. (2012) achieved 40.7 % recognition accuracy by modeling the relationship between dense trajectory clusters. Jain et al. (2013b) better exploited motion features by removing camera motion, and used VLAD as encoding method, where they obtained 46.6 % performance. Our multi-level representation significantly outperforms these methods by 10 to 15 %, which indicates the effectiveness of using representations from different layers. We also compare our MoFAP representation with Motionlet features (Wang et al. 2013c) and our method obtains much higher recognition accuracy. Finally, we compare our results with the state-of-the-art method Wang and Schmid (2013a) and our representation outperforms it by 4.5 %.

UCF101 is probably the newest action dataset till now. So few published papers evaluate the performance of their methods on this dataset. We mainly compare our representation with the winner of the THUMOS13 Action Recognition Challenge (Jiang et al. 2013) and the latest methods in CVPR 2014. Karpathy et al. (2014) explored the deep Convolutional Networks (ConvNets) for video classification. They trained their network with an extra 1M training dataset and then adapted the trained model to the UCF101 dataset. Our proposed multi-level representation significantly outperforms

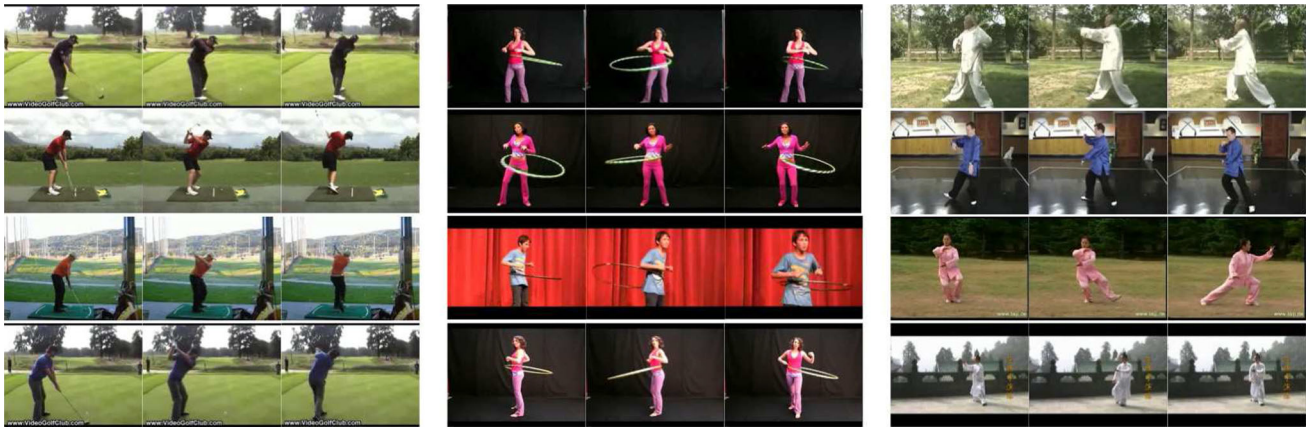

Fig. 6 Three examples of motion atoms, **Left**: motion atom corresponds primitive action of Golf-Swing; *Center* motion atom corresponds to primitive action of Hula-Hoop; *right* motion atom corresponds to primitive action in of Tai-Chi

this deep learning based method. This result may be ascribed to the fact that the diversity and complexity of video data is much higher than that of image data. Therefore, it requires much more data to train ConvNets for handling the complexity of video data. However, in practice, the size of action dataset is much smaller than the image dataset, and training the ConvNets in video domain is much more time consuming, all of which leads to the poor performance of their method. Recently, Simonyan and Zisserman (2014) propose a two stream ConvNet for action recognition, by utilizing the pre-trained of ImageNet model for spatial ConvNet initialization and hand-crafted optical flow fields as temporal ConvNet input. They obtain a similar performance to our multi-level representation on the dataset of UCF101. The methods in Cai et al. (2014) and Wu et al. (2014) aim to design more effective and powerful encoding scheme to boost the recognition performance. However, their improvement over Fisher Vector is smaller than that of our multi-level representation. Finally, our result outperforms the winner (Wang and Schmid 2013b) in THUMOS13 by 2.4 %, where they use the spatio-temporal pyramid to incorporate structure information.

6.8 Visualization of Motion Atoms and Phrases

In this subsection, we provide the examples of learned motion atoms and phrases. We first show several examples of discovered motion atoms in Fig. 6. From these results, we observe that the designed discriminative clustering method is able to learn motion atom classifiers that detect segments with similar appearance and motion. Each motion atom may correspond to the primitive actions such as golf-swing, hula-hoop and tai-chi. Thus, the motion atoms can be viewed as mid-level units and used to bridge the semantic gap between low-level features and high-level action concepts.

Figure 7 shows several detected motion phrases on the Olympic Sports dataset. From these examples, we notice that the mined motion phrases can automatically detect temporal composite of primitive motions that is of great importance for complex action recognition. We first show two examples of motion phrases mined from a single scale. For the action of hammer-throw, motion phrase decomposes the video into short segments corresponding to rolling and throwing respectively. The action of basketball is divided into running and lay-up. Then, we also give two examples of motion phrases mined from multiple scales for action classes of triple-jump and diving-platform. We see that these hierarchical motion phrases are able to localize the motion atoms of longer duration in the coarser temporal scale and each motion atom is further divided into two motion atoms of short duration in the finer temporal scale. For example of diving-platform, the rolling motion atom in the coarser scale is divided into jumping-up and falling-down in the finer scale. To sum up, these results demonstrate that our proposed motion phrase mining algorithm is effective in modeling the temporal structure of complex action.

7 Conclusions and Discussions

This paper has proposed a new multi-level representation of video, called *MoFAP*, by utilizing the activations from motion features, motion atoms, and motion phrases. This multi-level representation is able to capture the visual information from different scales, and provides complementary visual cues for action recognition. The effectiveness of MoFAP has been demonstrated on four challenging datasets: Olympic Sports, UCF50, HMDB51, and UCF101, and our approach obtains the state-of-the-art performance on these datasets. In addition to this superior performance, several additional interesting insights are concluded as follows.

Motion atoms are discovered through a discriminative clustering algorithm and act as mid-level units to represent the video data. The activations of motion atoms is better than activations of motion features on three challenging datasets.

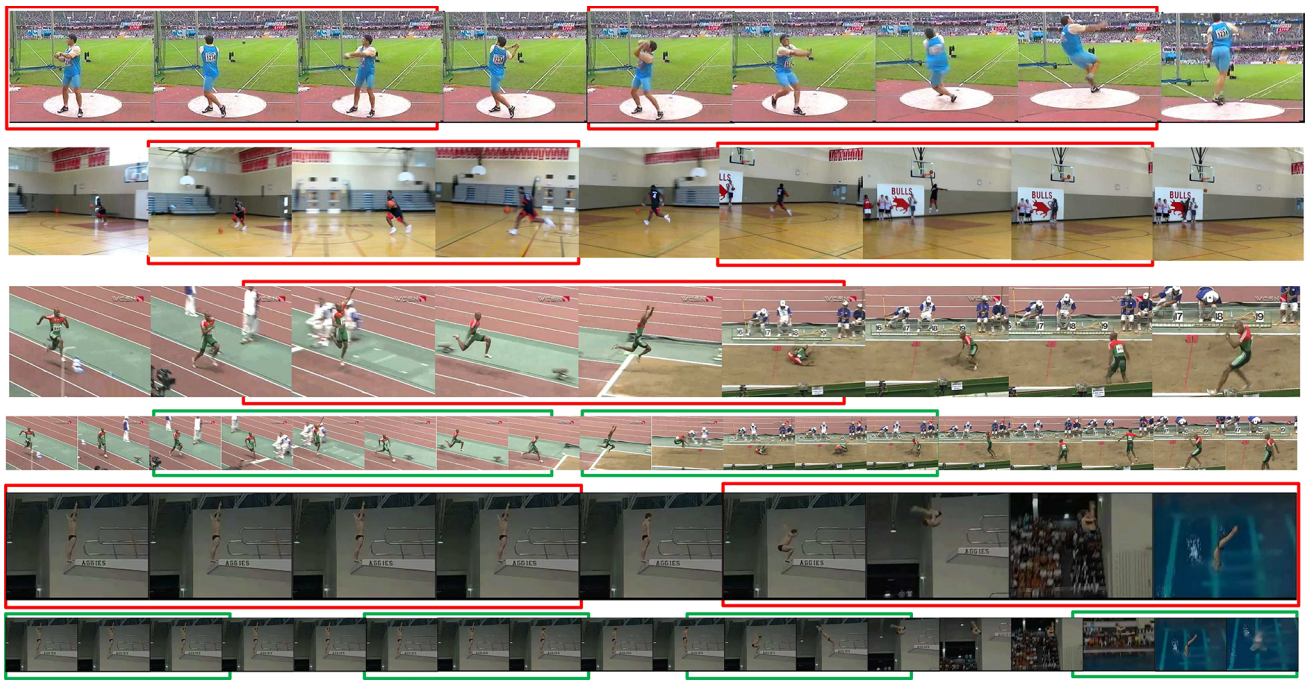

Fig. 7 Two examples of 2-motion phrase from a single temporal scale for complex actions: hammer-throw and basketball-layup. An example of 3-motion phrase and 4-motion phrase from multiple temporal scales for complex actions: triple-jump and diving platform. For multiple scales, we use *red boxes* to denote localized motion atoms in coarser temporal scale and *green boxes* to denote localized motion atoms in finer temporal scale (Color figure online)

Furthermore, the activation of motion atoms is much more compact than Fisher vector (around 15,000 vs. 100,000 on UCF101). In a word, the activations of motion atoms not only perform well but also keep low dimensionality.

Motion phrases are defined over multiple motion atoms using an AND/OR structure. They have the capacity of both dealing with local temporal displacement and capturing temporal and hierarchical structure. This structured representation is complementary to motion atom, and pretty effective for complex action with longer temporal duration, such as the Olympic Sports dataset.

The proposed multi-level representation is very effective for boosting final recognition performance. Although the construction of motion atoms and phrases is based on low-level features, they are able to describe the video data from different levels. The representations from these different levels can provide complementary information to low-level features. Thus, in practice, stacking multi-layer representations works pretty well for action recognition.

We also conduct a primary test about the generalization ability of motion atoms in the cross-dataset setting. Although the results of transferred representation are significantly lower than that of the original representation, fusing them may lead to higher recognition performance. In the future, we will focus on making the discovered mid-level motion atoms be able to generalize well on the large datasets and other video tasks.

References

Aggarwal, J. K., & Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys*, *43*(3), 16.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *VLDB* (pp. 487–499).

Amer, M. R., Xie, D., Zhao, M., Todorovic, S., & Zhu, S. C. (2012). Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *ECCV* (pp. 187–200).

Berg, T. L., Berg, A. C., & Shih, J. (2010). Automatic attribute discovery and characterization from noisy web data. In *ECCV* (pp. 663–676).

Bishop, C. (2006). *Pattern recognition and machine learning* (Vol. 4). Berlin: Springer.

Bourdev, L. D., & Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV* (pp. 1365–1372).

Cai, Z., Wang, L., Peng, X., & Qiao, Y. (2014). Multi-view super vector for action recognition. In *CVPR* (pp. 596–603).

Chang, C. C., & Lin, C. J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27.

Chen, Y., Zhu, L., Lin, C., Yuille, A. L., & Zhang, H. (2007). Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In *NIPS*.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV, Prague* (Vol. 1, pp. 1–2).

Doersch, C., Gupta, A., & Efros, A. A. (2013). Mid-level visual element discovery as discriminative mode seeking. In *NIPS* pp. 494–502.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Forsyth, D. A., Arikan, O., Ikemoto, L., O'Brien, J. F., & Ramanan, D. (2005). Computational studies of human motion: Part 1, tracking and motion synthesis. Foundations and Trends in Computer Graphics and Vision 1(2/3).

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*, 972–976.

Gaidon, A., Harchaoui, Z., & Schmid, C. (2013). Temporal localization of actions with actoms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2782–2795.

Gaidon, A., Harchaoui, Z., & Schmid, C. (2014). Activity representation with motion hierarchies. *International Journal of Computer Vision*, *107*(3), 219–238.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253.

Jain, A., Gupta, A., Rodriguez, M., & Davis, L. S. (2013a). Representing videos using mid-level discriminative patches. In *CVPR* (pp. 2571–2578).

Jain, M., Jegou, H., & Bouthemy, P. (2013b). Better exploiting motion for better action recognition. In *CVPR* (pp. 2555–2562).

Jiang, Y., Dai, Q., Xue, X., Liu, W., & Ngo, C. (2012). Trajectory-based modeling of human actions with motion reference points. In *ECCV* (pp. 425–438).

Jiang, Y. G., Liu, J., Roshan Zamir, A., Laptev, I., Piccardi, M., Shah, M., & Sukthankar, R. (2013). THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/ICCV13-Action-Workshop/.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *CVPR* (pp. 1725–1732).

Kliper-Gross, O., Gurovich, Y., Hassner, T., & Wolf, L. (2012). Motion interchange patterns for action recognition in unconstrained videos. In *ECCV* (pp. 256–269).

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: A large video database for human motion recognition. In *ICCV* (pp. 2556–2563).

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2–3), 107–123.

Laxton, B., Lim, J., & Kriegman, D. J. (2007). Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *CVPR* (pp. 1–8).

Liu, J., Kuipers, B., & Savarese, S. (2011). Recognizing human actions by attributes. In *CVPR* (pp. 3337–3344).

Niebles, J. C., Chen, C. W., & Li, F. F. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV* (pp. 392–405).

Oliver, N., Rosario, B., & Pentland, A. (2000). A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 831–843.

Parikh, D., & Grauman, K. (2011). Relative attributes. In *ICCV* (pp. 503–510).

Pirsiavash, H., & Ramanan, D. (2014). Parsing videos of actions with segmental grammars. In *CVPR* (pp. 612–619).

Raptis, M., Kokkinos, I., & Soatto, S. (2012). Discovering discriminative action parts from mid-level video representations. In *CVPR* (pp. 1242–1249).

Reddy, K. K., & Shah, M. (2013). Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, *24*(5), 971–981.

Rohrbach, M., Regneri, M., Andriluka, M., Amin, S., Pinkal, M., & Schiele, B. (2012). Script data for attribute-based recognition of composite activities. In *ECCV*.

Sadanand, S., & Corso, J. J. (2012). Action bank: A high-level representation of activity in video. In *CVPR* (pp. 1234–1241).

Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. J. (2013). Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, *105*(3), 222–245.

Sapienza, M., Cuzzolin, F., & Torr, P. H. S. (2012). Learning discriminative space-time actions from weakly labelled videos. In *BMVC* (pp. 1–12).

Schüldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local svm approach. In *ICPR*.

Si, Z., & Zhu, S. C. (2013). Learning AND-OR templates for object recognition and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(9), 2189–2205.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *NIPS* (pp. 568–576).

Singh, S., Gupta, A., & Efros, A. A. (2012). Unsupervised discovery of mid-level discriminative patches. In *ECCV* (pp. 73–86).

Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *ICCV* (pp. 1470–1477).

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR abs/1212.0402.

Tang, K. D., Li, F. F., & Koller, D. (2012). Learning latent temporal structure for complex event detection. In *CVPR* (pp. 1250–1257).

Turaga, P. K., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), 1473–1488.

Wang, H., & Schmid, C. (2013a). Action recognition with improved trajectories. In *ICCV* (pp. 3551–3558).

Wang, H., & Schmid, C. (2013b). Lear-inria submission for the thumos workshop. In: ICCV Workshop on Action Recognition with a Large Number of Classes.

Wang, H., Kläser, A., Schmid, C., & Liu, C. L. (2013a). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, *103*(1), 60–79.

Wang, L., Qiao, Y., & Tang, X. (2013b). Mining motion atoms and phrases for complex action recognition. In *ICCV* (pp. 2680–2687).

Wang, L., Qiao, Y., & Tang, X. (2013c). Motionlets: Mid-level 3D parts for human motion recognition. In *CVPR* (pp. 2674–2681).

Wang, L., Qiao, Y., & Tang, X. (2014a). Latent hierarchical model of temporal structure for complex activity classification. *IEEE Transactions on Image Processing*, *23*(2), 810–822.

Wang, L., Qiao, Y., & Tang, X. (2014b). Video action detection with relational dynamic-poselets. In *ECCV* (pp. 565–580).

Wang, L., Qiao, Y., & Tang, X. (2015). Action recognition with trjectory-pooled deep-convolutional descriptors. In *CVPR* (pp. 4305–4314).

Wang, S. B., Quattoni, A., Morency, L. P., Demirdjian, D., & Darrell, T. (2006). Hidden conditional random fields for gesture recognition. In *CVPR* (pp. 1521–1527).

Wang, X., Wang, L., & Qiao, Y. (2012). A comparative study of encoding, pooling and normalization methods for action recognition. In *ACCV* (pp. 572–585).

Wu, J., Zhang, Y., & Lin, W. (2014). Towards good practices for action video encoding. In *CVPR* (pp. 2577–2584).

Yao, B., & Li, F. F. (2010). Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*.

Zhang, W., Zhu, M., & Derpanis, K. G. (2013). From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV* (pp. 2248–2255).

Zhao, Y., & Zhu S. C. (2011). Image parsing with stochastic scene grammar. In *NIPS* (pp. 73–81).

Zhu ,J., Wang, B., Yang, X., Zhang, W., & Tu, Z. (2013). Action recognition with actons. In *ICCV* (pp. 3559–3566).