Learning Composite Latent Structures for 3D Human Action Representation and Recognition

Ping Wei, *Member, IEEE*, Hongbin Sun, *Member, IEEE*, and Nanning Zheng, *Fellow, IEEE*

***Abstract*—3D human action representation and recognition are important issues in many multimedia applications. While latent state approaches have been widely used for action modeling, previous works assume the latent states of actions are single attribute. This assumption is inaccurate for representing structures of complex actions. In this paper, we propose that latent states have composite attributes and introduce a novel composite latent structure (CLS) model to represent and recognize 3D human actions with skeleton sequences. A human action is modeled with a hierarchical graph, which represents the action sequence as sequential atomic actions. An atomic action is represented as a composite latent state, which is composed of a latent semantic attribute and a latent geometric attribute. A discriminative EM-like algorithm is proposed to learn the model parameters and the composite latent structures of human actions. Given a 3D skeleton sequence, a composite attribute iterative programming algorithm is proposed to recognize the action and infer the action's latent temporal structure. We evaluate the proposed method on three challenging 3D action datasets—MSR 3D Action Dataset, Multiview 3D Event Dataset, and UTKinect-Action 3D Dataset. Extensive experimental results on these datasets demonstrate the effectiveness and advantage of the proposed method.**

***Index Terms*—3D human action, action representation, action recognition, composite latent structure.**

I. Introduction

Modeling 3D human actions is of great importance in many multimedia applications, such as content-based information retrieval, 3D human animation, educational entertainment, and multimedia learning. In addition to recognizing the classes of actions, learning and establishing structure representations for actions are also important but challenging issues. For example, in an edutainment system where a human is interacting with a computer, the computer needs not only to recognize the human action but also to understand the action's temporal structure so that its response can precisely coordinate with the human action.

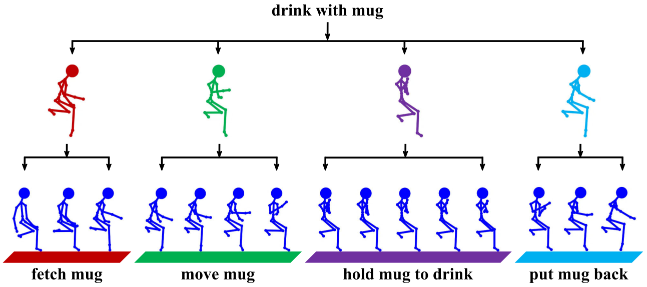


Fig. 1. An example of 3D human action division in time axis. The colorful segments correspond to different atomic actions, which are represented as latent states in our model. We label them with semantic categories for illustration.

Modeling latent temporal structures of actions is one of the most widely-used techniques for action representation and recognition [1]–[7]. The general idea of the latent structure methods is that an action is composed of multiple latent substructures or states in the temporal domain [1], [3], [5]. However, in most previous works, the latent states are assumed to be single-attribute, i.e. a latent state is described by only one type of attribute. A limitation of single-attribute latent models is that they only use one type of latent information for action modeling, which is inaccurate for representing structures of complex actions. For example, in Hidden Markov Models (HMM) [1], [7] and Hidden Conditional Random Fields (HCRF) [2], an action is represented as a sequence of hidden states. Each state is interpreted by a hidden category, and the temporal window size of each state is defined as the same constant. However, the temporal window size of each state might be different. As the action *drink with mug* shown in Fig. 1, the sub-actions *fetch mug* and *move mug* have different durations. The different temporal window sizes also contain useful information for action modeling and therefore should not be manually defined as constants but rather mined from data as variables.

Indeed, latent states have composite attributes, which means a latent state is interpreted by multiple types of attributes and these attributes closely interact with each other. For example, in Fig. 1, the action sequence *drink with mug* can be divided into several temporal clips which correspond to atomic actions with the semantic categories: *fetch mug*, *move mug*, *hold mug to drink*, and *put mug back*. In addition to the semantic category, each atomic action has a geometric attribute defined by the temporal

Manuscript received July 16, 2018; revised November 4, 2018 and January 6, 2019; accepted January 7, 2019. Date of publication February 6, 2019; date of current version August 23, 2019. This work was supported in part by the National Natural Science Foundation of China under Grants 61876149, 61503297, 61722406, in part by China Postdoctoral Science Foundation 2018M643657, and in part by the National Key Research and Development Program of China 2016YFB1000903. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Manoranjan Paul. *(Corresponding author: Hongbin Sun.)*

The authors are with Xi'an Jiaotong University, Shaanxi 710049, China (e-mail: pingwei@xjtu.edu.cn; hsun@xjtu.edu.cn; nnzheng@xjtu.edu.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/TMM.2019.2897902

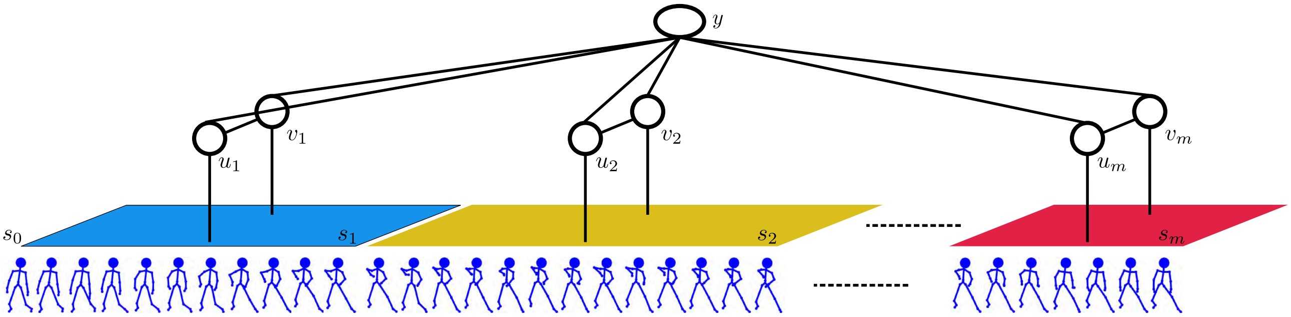


Fig. 2. The composite latent structure model. The colorful segments correspond to the atomic actions which are described by the semantic category attributes and the geometric interval attributes.

interval, as the colorful segments in Fig. 1. The human poses and motions have similar characteristics in the same atomic action interval, but present different characteristics in different atomic action intervals. The semantic attribute indicates which category an atomic action belongs to and the geometric attribute defines where the atomic action is and how long it lasts in the whole sequence.

However, it is difficult to manually define the atomic action categories and their interval boundaries in an action sequence. Automatically mining these attributes from data would be more reasonable and effective since these attributes may hide significant information of interpreting action sequences. Thus, both the semantic and geometric attributes of atomic actions are treated as latent variables. In this manner, an atomic action is a composite latent state which includes a latent semantic attribute and a latent geometric attribute. An action sequence is composed of several composite latent states.

Moreover, different types of latent attributes are not independent but rather closely interact with each other. For example, the semantic attribute of *fetch mug* is constrained by its geometric attribute since *fetch mug* is usually located in the initial phase of the whole action and often lasts for a period with a relatively fixed length. We believe that mining and utilizing the composite latent structures can benefit action modeling and recognition.

In this paper, we propose a composite latent structure (CLS) model to represent and recognize 3D human actions, as shown in Fig. 2. The inputs of our method are action sequences of 3D human skeletons which are estimated by the motion capture technology, such as Kinect camera [8]. A human action sequence is divided into several sequential temporal intervals, each of which corresponds to an atomic action. An atomic action is represented as a composite latent state including the latent semantic attribute and the latent geometric attribute. A hierarchical graph is employed to formulate the hierarchical structure of the action, the atomic actions, and the input data.

To learn the model parameters and mine the action structures, we propose a discriminative EM-like algorithm which carries out the learning process under the conventional EM framework but with discriminative optimization. Given a 3D human skeleton sequence, a composite attribute iterative programming algorithm is proposed to infer the composite latent structure and recognize the human action.

We evaluate the proposed method on three representative and challenging 3D human action datasets: MSR 3D Action Dataset [9], Multiview 3D Event Dataset [10], and UTKinect-Action3D Dataset [11]. The extensive experimental results show that the proposed method improves the action recognition accuracy. We also analyze the effects of the composite attributes and the latent state numbers on action recognition. Furthermore, we compare the effects of pose features and motion features. Finally, we visualize the composite latent structures in action sequences.

In comparison with previous studies, the major contribution of our work is that it models action structures with composite latent states. This new perspective develops the concept of latent states from single attributes to composite attributes, which provides new insights into video representation and modeling.

Another contribution of our method is that it can learn and construct composite latent structure representations of human actions which can be potentially applied to many multimedia applications, such as content-based information retrieval and educational entertainment.

It should be noted that our work does not use neural network or deep learning techniques and its action recognition performance is lower than some deep learning methods. However, our method not only aims at recognizing actions but also constructing composite latent structure representations of human actions, which was not well addressed in other methods.

This paper is organized as follows. In Section II, we briefly review the related work. Section III introduces the composite latent structure model and the skeleton feature representation. Section IV presents the model inference algorithm and Section V describes how to mine the latent structures and learn the model parameters. The experiments and evaluations are presented in Section VI.

II. Related Work

We will review the previous works from three main related streams of research.

A. 3D Action Recognition

With the rapid advance of motion capture technology, 3D action modeling and recognition have recently received growing attention for its significance in many applications [5], [12]–[25]. The 3D human action data often presents as RGB-D or depth values of human body [26]–[29], or 3D human skeletons representing the 3D locations of human body joints [30]–[33].

Compared to 2D human actions in images or videos, 3D human skeleton data is sparser and more sensitive to spatial noise and temporal warping. Many approaches have been proposed to model 3D actions with skeleton data. Ben Tanfous *et al.* [12] encoded skeleton shape trajectories with a sparse coding method on the Kendalls shape space, which achieved impressive results on 3D skeleton action recognition. Ke *et al.* [13] transformed a skeleton sequence to clips and learned long-term temporal information with deep convolutional neural networks. Wang *et al.* [14] represented 3D skeletons with joint location differences and the sequences with hierarchical Fourier features. These features are then fed to a discriminative actionlet ensemble model to recognize actions. Wei *et al.* [15] applied wavelet to trajectories of the joint location differences to describe a sequence clip. Hu *et al.* [27] further introduced the Fourier analysis to the temporal gradient of human skeletons, and proposed a joint heterogeneous feature learning method for action recognition. Wei *et al.* [5] applied PCA analysis to human skeleton joints and their motion vectors, and used the PCA parameters to characterize 3D human action features. Yang *et al.* [17] generated depth motion maps by extracting motion information between successive depth frames for recognizing actions in depth sequences. Jia and Fu [20] represented the RGB-D sequence data with third-order tensors and learned the tensor subspace dimension by low-rank learning. Inspired by these works, we utilize the information in both spatial and temporal domains to characterize the 3D human skeleton features.

Compared to the above methods which recognize actions from sequences, our method can also learn and mine the latent substructures from sequence samples. These substructures show promising potential and play important roles in many multimedia applications, such as human-computer interaction [34] and human animation [35].

In recent years, neural network and deep learning methods have been extensively used in action recognition and achieved impressive results [36]–[46]. However, most deep learning based methods were inapplicable to establish temporal structure representations of actions. And deep learning based methods need large-scale data for training and powerful hardware for computation, which is inapplicable to many multimedia applications such as human-computer interaction and educational entertainment.

B. Human Action Attributes

Many previous studies analyze human actions through modeling attributes and parts [47]–[52]. Yao *et al.* [47] used attributes to represent human action properties and jointly modeled attributes and parts to recognize actions in still images. Liu *et al.* [51] used a group of semantic attributes to represent human actions and learned the attribute importance to the actions with a latent SVM. Pei *et al.* [50] proposed a framework integrating heterogeneous attributes to detect actions in videos. Su *et al.* [49] represented part-wise and body-wise attributes as latent states to model actions.

While these studies introduce explicit or latent attributes into action modeling, those attributes are defined to characterize features in the spatial domain, such as human bodies, body parts, and scenes. They did not model nor represent the temporal attributes of human actions, such as the atomic actions and their intervals in action sequences. Different from them, we propose new methods to represent and mine the latent attributes of human actions in the temporal domain.

C. Action Structure Modeling

It has been demonstrated that modeling and mining substructures of human actions can improve the performance of action recognition, detection, and segmentation [3], [4], [14], [20], [27], [33], [53]–[62]. Lv and Nevatia [1] defined hidden states and their transitions in videos for joint recognition and segmentation of 3D human actions. HCRF [2] defines a conditional random field over the action class and hidden states in fixed-size intervals for action recognition. Wang and Mori [4] proposed hidden part models to describe human pose deformations in each frame. Pei *et al.* [63] and Wei *et al.* [5] decomposed an action sequence into atomic units and used stochastic graphs to represent human actions. Zheng *et al.* [64] learned a local descriptor for action recognition in RGB videos.

Those methods inspire us to mine temporal structures to analyze actions. However, they do not characterize the composite attributes of actions' latent states but model the latent states with single type of attributes. In contrast, our model represents human actions with composite latent attributes.

III. Composite Latent Structure Model

Action recognition is to predict a class label for an input video sequence. An action is divided into several atomic actions which represent the action's sub-processes. For example, as shown in Fig. 2, the action *pour water from kettle* can be divided into several atomic actions, such as *fetch kettle*, *hold kettle to pour water to mug*, and *put kettle back*. In the same atomic action, the human poses and motions have similar characteristics; in different atomic actions, the human poses and motions present different patterns.

Let $\mathbf{x} = (x_1, x_2, \dots, x_T)$ be an input video sequence, where x_t $(t = 1, \dots, T)$ is the feature of the frame at time t and T is the sequence length. $y \in \mathcal{Y}$ is the action class label of the input sequence $\mathbf{x}$, where $\mathcal{Y}$ is the set of all action class labels, such as *drink with mug*, *use computer*, *eat food*, and *walk*.

We assume that the time boundaries $s_0, s_1, \dots, s_{m-1}$, and s_m segment the sequence $\mathbf{x}$ into m sub-sequences and each sub-sequence corresponds to an atomic action, as shown in Fig. 2. The time boundaries satisfy the following conditions:

$$\begin{cases} s_0 = 0, & s_m = T, \\ s_i \in \{1, \dots, T-1\}, & \forall i = 1, \dots, m-1, \\ s_{i-1} < s_i, & \forall i = 1, \dots, m. \end{cases} \quad (1)$$

The ith sub-sequence starts at time $s_{i-1} + 1$ and ends at s_i. We define $v_i = [s_{i-1} + 1, s_i]$ as the time interval of the ith atomic action in the the sequence $\mathbf{x}$.

Let $u_i \in \mathcal{U} = \{1, \ldots, K\}$ be the category label of the ith atomic action in the sequence $\mathbf{x}$. $\mathcal{U}$ is the set of all atomic action categories such as *fetch mug* and *move mug* in the action *drink with mug*. K is the number of all atomic actions. We assume the relations between an action class and its atomic action categories are hard constraints. For example, the sub-action *hold kettle to pour water to mug* indicates the action is *pour water from kettle*.

The category labels and time intervals of the atomic actions in a sequence are not manually annotated nor observed in data. They are both latent variables. Each subsequence is described by two latent variables (u_i, v_i). The category label u_i describes the semantic attribute of the subsequence and the interval variable v_i characterizes the geometric attribute. We define (u_i, v_i) as the composite latent variable.

Let $\mathbf{u} = \{u_1, u_2, \ldots, u_m\}$ and $\mathbf{v} = \{v_1, v_2, \ldots, v_m\}$ be the composite latent variables of the sequence $\mathbf{x}$. The score that the sequence $\mathbf{x}$ is interpreted by $\mathbf{u}$, $\mathbf{v}$, and y is formulated as

$$S(\mathbf{x}, \mathbf{u}, \mathbf{v}, y) = \sum_{i=1}^{m} \sum_{t \in v_i} \phi(x_t, u_i, \boldsymbol{\omega}), \tag{2}$$

where $\phi(x_t, u_i, \boldsymbol{\omega})$ is the frame score which is defined as

$$\phi(x_t, u_i, \boldsymbol{\omega}) = \log \frac{1/(1 + e^{-\omega_{u_i}^{\mathrm{T}} \cdot x_t})}{\sum_{k=1}^{K} 1/(1 + e^{-\omega_k^{\mathrm{T}} \cdot x_t})}. \tag{3}$$

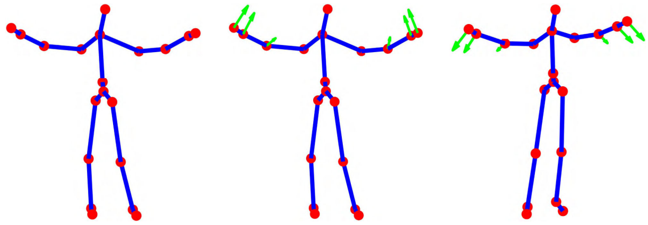
Equation (3) describes the score of an atomic action in a frame. $\boldsymbol{\omega} = \{\omega_1, \ldots, \omega_K\}$ is the model parameter where ω_k is the template parameter of the atomic action k. ω_{u_i} describes the compatibility between the atomic action category u_i and the frame feature x_t. Since ω_{u_i} is related to the latent variables u_i and v_i, it should be jointly learned with u_i and v_i from training samples. Since the relations between an action and its atomic actions are hard constraints, for clarity, the variable y is omitted in the right side of Eq. (2).

In Eq. (2), $\mathbf{u}$ describes the semantic structure of the action sequence and $\mathbf{v}$ characterizes the geometric structure of the action sequence. They are both latent variables and should be mined from data. On the other hand, by mining $\mathbf{u}$ and $\mathbf{v}$, we can obtain the semantic and geometric structure representation of an action.

A. Feature Representation

The inputs in our work are sequences of 3D human skeletons. A 3D human skeleton is composed of 3D positions of human body joints, as shown in Fig. 3(a). These 3D human skeletons are estimated by motion capture technologies, such as the Kinect camera [8].

It has been shown that the relative difference vectors among joints have strong discriminative ability for action recognition [14]. Additionally, the motion information of human bodies is also important to action recognition. As shown in Fig. 3(b) and (c), the two similar poses have different motions, which make them have different atomic action labels. We take advantages of both the pose feature and the motion feature to describe human skeletons.



(a) (b) (c)

Fig. 3. The illustration of 3D skeleton features. (a) 3D skeleton joints of a frame in the action *wave hands*. (b) Motion vectors in the action *wave hands (up)*. (c) Motion vectors in the action *wave hands (down)*. For clarity, we only visualize the motion vectors on the arms.

x_t is the feature extracted from the 3D human skeletons. It is a concatenation of the features of the joints on the human skeleton, i.e. $x_t = \{x_t^i | i = 1, \ldots, J\}$, where x_t^i is the feature of the ith joint and J is the number of skeleton joints. The feature x_t^i is composed of the motion feature $x_{t,mot}^i$ and the pose feature $x_{t,pos}^i$,

$$x_t^i = (x_{t,mot}^i, x_{t,pos}^i). \tag{4}$$

Suppose z_t^i $(i = 1, \ldots, J)$ is the 3D coordinate of the ith joint on the human skeleton at time t. For the motion feature of the joint i, we compute the difference vectors between the joint i at time t and all the joints at time $t-1$, i.e.

$$x_{t,mot}^{i,j} = z_t^i - z_{t-1}^j, j = 1, \ldots, J. \tag{5}$$

The motion feature $x_{t,mot}^i$ of the joint i is the concatenation of all the difference vectors, i.e.

$$x_{t,mot}^i = \{x_{t,mot}^{i,j} | j = 1, \ldots, J\}. \tag{6}$$

We use the same method defined in the work [14] to compute the difference vectors. The pose feature $x_{t,pos}^i$ of the joint i is defined as the concatenation of all the difference vectors between this joint and all other joints at time t.

The pose features describe the still appearance information and the motion features characterize the changing tendency information. The combination of the pose features and motion features can capture more discriminative information in human skeletons for action recognition.

IV. Inference

Given a sequence $\mathbf{x}$, action recognition is to predict an action class label for the sequence. The score of labeling $\mathbf{x}$ with an action label $y \in \mathcal{Y}$ is

$$f(\mathbf{x}, y) = S(\mathbf{x}, \mathbf{u}^*, \mathbf{v}^*, y), \tag{7}$$

where $\mathbf{u}^*$ and $\mathbf{v}^*$ are the optimal latent variables for the pair $(\mathbf{x}, y)$,

$$(\mathbf{u}^*, \mathbf{v}^*) = \arg\max_{\mathbf{u}, \mathbf{v}} S(\mathbf{x}, \mathbf{u}, \mathbf{v}, y). \tag{8}$$

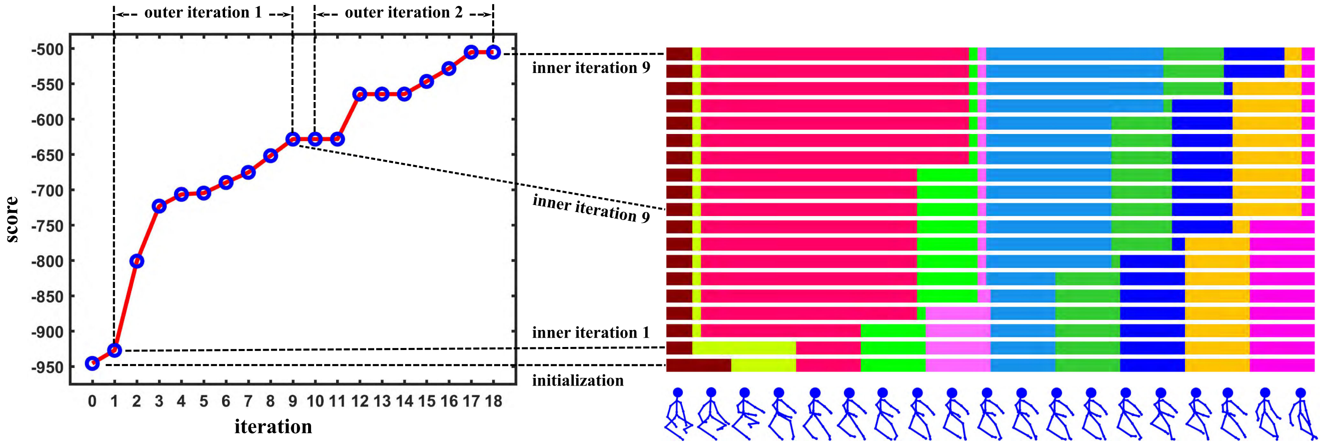


Fig. 4. An illustration of the composite attribute iterative programming algorithm. Each colorful bar in the right figure corresponds to an inner iteration step.

The optimal action label y^* for labeling $\mathbf{x}$ is

$$y^* = \arg\max_{y \in \mathcal{Y}} f(\mathbf{x}, y). \tag{9}$$

Computation of the optimal action class label y^* involves the optimization of the latent variables $\mathbf{u}$ and $\mathbf{v}$ as defined in Eq. (8). However, the complexity of solving Eq. (8) is exponentially related to the space size of $\mathbf{u}$ and $\mathbf{v}$. A sequence may have more than a thousand frames, which means there are a huge number of possible interval segmentation defined by $\mathbf{v} = \{v_1, v_2, \ldots, v_m\}$. We propose a composite attribute iterative programming (CAIP) algorithm to solve this problem and optimize Eq. (8).

A. *Composite Attribute Iterative Programming*

The composite attribute iterative programming (CAIP) algorithm aims to compute the optimal composite latent variables for a given sequence $\bar{\mathbf{x}}$ with a hypothesized label $\bar{y}$. The general idea of CAIP algorithm is to separate the optimization of the composite attribute vector into the iterative optimization of each vector item step by step. As shown in Fig. 4, the iterative optimization consists of two iteration processes: outer iterations and inner iterations. The outer iteration process corresponds to the optimization of the entire composite attribute vector and the inner iteration process aims to optimize each vector item step by step. At each step of the inner iteration, one item of the composite attribute vector is optimized by fixing other items of the vector. The process of inner-out iterative optimization is iteratively performed until the optimization condition is satisfied. This inner-out iteration process is illustrated in Fig. 4.

For the given sequence $\bar{\mathbf{x}}$ with the label $\bar{y}$, $\mathbf{v} = \{v_1, v_2, \ldots, v_m\}$ is uniquely determined by $s_1, \ldots, s_{m-1}$. For convenience, we introduce $\mathbf{s} = \{s_1, \ldots, s_{m-1}\}$ and define

$$h(\mathbf{u}, \mathbf{s}) = S(\bar{\mathbf{x}}, \mathbf{u}, \mathbf{v}, \bar{y}). \tag{10}$$

The problem (8) is equivalent to solving the following problem:

$$(\mathbf{u}^*, \mathbf{s}^*) = \arg\max_{\mathbf{u}, \mathbf{s}} h(\mathbf{u}, \mathbf{s}). \tag{11}$$

Algorithm 1: CAIP: Composite Attribute Iterative Programming Algorithm

```
1: **Initialization:** $\mathbf{u}^0$, $\mathbf{s}^0$.
2: **Outer iteration:** set $i = i + 1$.
     **Inner iteration:** sequentially perform
     $(u_1^i, s_1^i) = \arg\max_{u_1, s_1} h(u_1, s_1, \mathbf{u}^{i-1} \backslash u_1^{i-1}, \mathbf{s}^{i-1} \backslash s_1^{i-1})$,
        $u_1^{i-1} = u_1^i, \ \ s_1^{i-1} = s_1^i$,
     $(u_2^i, s_2^i) = \arg\max_{u_2, s_2} h(u_2, s_2, \mathbf{u}^{i-1} \backslash u_2^{i-1}, \mathbf{s}^{i-1} \backslash s_2^{i-1})$,
        $u_2^{i-1} = u_2^i, \ \ s_2^{i-1} = s_2^i$,
                  $\vdots$
        $u_m^i = \arg\max_{u_m} h(u_m, \mathbf{u}^{i-1} \backslash u_m^{i-1}, \mathbf{s}^{i-1})$.
3: **Convergence check**. If the convergence condition is
     satisfied, stop and output $(\mathbf{u}^i, \mathbf{s}^i)$; else, return to step 2.
```

Let $\mathbf{u}^i = (u_1^i, \ldots, u_m^i)$ and $\mathbf{s}^i = (s_1^i, \ldots, s_{m-1}^i)$ be the values in the ith outer iteration. We denote $\mathbf{u}^i \backslash u_j^i$ as all the items of $\mathbf{u}^i$ except u_j^i, i.e. $\mathbf{u}^i \backslash u_j^i = (u_1^i, \ldots, u_{j-1}^i, u_{j+1}^i, \ldots, u_m^i)$. $\mathbf{s}^i \backslash s_j^i$ is defined in the same way.

The CAIP algorithm is summarized in Algorithm 1. It is composed of three blocks: initialization, iteration, and termination. In the initialization step, the boundary variables $\mathbf{s}^0$ are initialized by setting each interval to be with the equal length, and the semantic variables $\mathbf{u}^0$ are initialized by setting each action class to be with distinct atomic action categories. The initialization is shown as the bottom bar in Fig. 4.

The iteration block of CAIP includes the outer iteration and the inner iteration. Each outer iteration is composed of several inner iteration steps at which each time boundary variable is sequentially optimized, as colorful bars shown in Fig. 4. After an outer iteration, all the items of the time boundary vector are optimized once. It should be noted that after each inner iteration, the corresponding items will be updated with the optimized values before the next inner iteration is performed. In the last inner iteration, only u_m is optimized rather than (u_m, s_m), since for a given video sequence s_m equals to the sequence length.

In the termination block, the convergence condition is checked. In our work, the convergence condition is defined by the score defined in Eq. (10). If this score remains unchanged in more than two successive outer iterations, the algorithm terminates and outputs the variable values as the final results.

B. Convergence Analysis of CAIP Algorithm

We propose and prove a proposition which indicates the convergence of the CAIP algorithm.

Proposition: The CAIP algorithm converges.

Proof: For a given action sequence $\bar{\mathbf{x}}$ and an action label $\bar{y}$, we define $h(\mathbf{u}^i, \mathbf{s}^i) = S(\bar{\mathbf{x}}, \mathbf{u}^i, \mathbf{v}^i, \bar{y})$ as the output score after the ith outer iteration in the CAIP algorithm. Our proof is achieved by proving two sub-propositions: 1) $h(\mathbf{u}^i, \mathbf{s}^i)$ is an upper bounded function; 2) $h(\mathbf{u}^i, \mathbf{s}^i) \geq h(\mathbf{u}^{i-1}, \mathbf{s}^{i-1})$. The first sub-proposition guarantees that the possible maximum score is a finite value. The second one indicates that each iteration of our algorithm will not decrease the score but rather possibly move the score towards a bigger value.

For valid action sequences and model parameters, Eq. (3) defines the sequence frame score given the latent variable, which shows that the frame score is upper-bounded. Since $h(\mathbf{u}^i, \mathbf{s}^i)$ is the summation of the score of each frame in a sequence with a finite length, it must be an upper bounded function.

We define

$$h\left(\mathbf{u}_j^{i-1}, \mathbf{s}_j^{i-1}\right) = \max_{u_j, s_j} h\left(u_j, s_j, \mathbf{u}^{i-1}\backslash u_j^{i-1}, \mathbf{s}^{i-1}\backslash s_j^{i-1}\right) \quad (12)$$

as the output score after the jth inner iteration (i.e. after the jth variable is optimized) in the CAIP algorithm.

According to Eq. (12) and the inner iterations in Algorithm 1, we can obtain the relation:

$$\begin{aligned} h\left(\mathbf{u}^{i-1}, \mathbf{s}^{i-1}\right) &\leq h\left(\mathbf{u}_1^{i-1}, \mathbf{s}_1^{i-1}\right) \\ &\vdots \\ &\leq h\left(\mathbf{u}_m^{i-1}, \mathbf{s}_m^{i-1}\right) \\ &= h\left(\mathbf{u}^i, \mathbf{s}^i\right). \end{aligned} \quad (13)$$

The formula (13) shows that $h(\mathbf{u}^i, \mathbf{s}^i) \geq h(\mathbf{u}^{i-1}, \mathbf{s}^{i-1})$. ■

C. Implementation of Action Recognition

To label a sequence $\mathbf{x}$ with an optimal action class, we should compute the composite latent variables for each hypothesized action label $\bar{y}$, and then compute the labeling score with Eq. (7). The action label is the one with the maximal score. This action recognition algorithm is shown in Algorithm 2.

V. Learning

In our composite latent structure model, each atomic action category $u_i \in \mathcal{U}$ has a template parameter ω_{u_i}. $\boldsymbol{\omega} = \{\omega_1, \ldots, \omega_K\}$ is the set of all the template parameters. Given the training data of n sequence-label samples $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_n, y_n)\}$, the goal is to learn $\boldsymbol{\omega}$ from these samples.

Algorithm 2: Action Recognition Algorithm

```
Input: A 3D human skeleton sequence $\mathbf{x}$;
Output: The optimal action label $y^*$ of $\mathbf{x}$;
for each $\bar{y} \in \mathcal{Y}$ do
    compute $(\mathbf{u}^*, \mathbf{v}^*) = \arg\max_{\mathbf{u},\mathbf{v}} S(\mathbf{x}, \mathbf{u}, \mathbf{v}, \bar{y})$ with CAIP;
    compute $f(\mathbf{x}, \bar{y}) = S(\mathbf{x}, \mathbf{u}^*, \mathbf{v}^*, \bar{y})$;
end for
return $y^* = \arg\max_{\bar{y} \in \mathcal{Y}} f(\mathbf{x}, \bar{y})$.
```

Suppose $(\mathbf{u}_i, \mathbf{v}_i)$ are the composite latent variables of the training sample $(\mathbf{x}_i, y_i)$. The total score $\Theta(\boldsymbol{\omega}, \mathbf{D})$ of all the training samples is defined as

$$\Theta(\boldsymbol{\omega}, \mathbf{D}) = \sum_{i=1}^{n} S(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i, y_i), \quad (14)$$

where $S(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i, y_i)$ is the score of the ith sample, as defined in Eq. (2). The learning problem is formulated as

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}} \Theta(\boldsymbol{\omega}, \mathbf{D}). \quad (15)$$

One challenge of solving the learning problem (15) is that it contains latent variables which define the division of the samples for each atomic action category. Thus, solving Eq. (15) is related to the optimization of these latent variables. Another challenge is that these latent variables are composite variables rather than single-attribute variables.

Expectation maximization (EM) methods are effective ways to solve latent-variable involved problems [65]. However, the above two challenges in the problem (15) make it difficult to learn the model parameters with the conventional EM methods.

Inspired by the recent studies on the latent model learning [5], [6], we propose a discriminative EM-like (DEML) learning method. The DEML algorithm is performed under the general EM framework [65] but with different techniques. In the E-like step, DEML computes the optimal composite latent variables for each sequence instead of computing the responsibilities as in the E step of conventional EM. In the M-like step, DEML adopts a discriminative learning method to update the model parameters rather than the maximum likelihood estimation in the M step of EM. The E-like step computes the assignments of samples for different atomic action categories and the discriminative learning in M-like step enhances the model's discrimination ability.

The DEML algorithm is presented as follows:

1) **Initialization.** Initialize $\mathbf{v}_i$ by cutting each sequence $\mathbf{x}_i$ into m segments with equal length; initialize $\mathbf{u}_i$ by assigning each segment with different atomic action categories of the action class y_i. With the initial $(\mathbf{u}_i, \mathbf{v}_i)$ for each sequence, compute initial $\boldsymbol{\omega}$ with Eq. (17).

2) **E-like step**. With current $\boldsymbol{\omega}$, compute the optimal latent variables $(\mathbf{u}_i^*, \mathbf{v}_i^*)$ for each sequence,

$$(\mathbf{u}_i^*, \mathbf{v}_i^*) = \arg\max_{\mathbf{u}_i, \mathbf{v}_i} S(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i, y_i). \quad (16)$$

3) **M-like step**. With current $(\mathbf{u}_i, \mathbf{v}_i)$ for each sequence, compute $\boldsymbol{\omega}$ by solving a series of L2-regularized logistic regression problems [66],

$$\omega_k^* = \arg\min_{\omega_k} \frac{1}{2}\omega_k^{\mathrm{T}} \cdot \omega_k + C\sum_{i=1}^{n}\sum_{t=1}^{T_i} \ln(1 + e^{-l_{i,t}\omega_k^{\mathrm{T}} \cdot x_{i,t}}), \tag{17}$$

where C is a weight constant. T_i is the length of the sequence $\mathbf{x}_i$. $x_{i,t}$ is the feature of frame at time t in the sequence $\mathbf{x}_i$. $l_{i,t}$ is a indicator variable whose value is 1 when the latent atomic action category of $x_{i,t}$ is k; else it equals -1. We solve Eq. (17) with the public implementation package released in the work [66].

4) **Evaluate** the score $\Theta(\boldsymbol{\omega}, \mathbf{D})$ with the new parameter $\boldsymbol{\omega}$ and latent variables $(\mathbf{u}_i, \mathbf{v}_i)$ of each sequence. If the convergence criterion is satisfied, stop and output the optimization results; else, return to 2) E-like step.

With the DEML algorithm, we can learn the model parameter $\boldsymbol{\omega}$ and mine the latent semantic and geometric structures of each video sequence.

VI. Experiment

We evaluate our method on three representative and challenging 3D action datasets - MSR 3D Action Dataset [9], Multiview 3D Event Dataset [10], and UTKinect-Action3D Dataset [11]. MSR 3D Action Dataset has a large number of action classes. Multiview 3D Event Dataset has large number of action sequences and is a multiview dataset with abundant human-object interactions. UTKinect-Action3D Dataset contains pairs of inverse actions and many actions have repeating sub-actions which bring challenges to action structure learning.

In the experiments, we evaluate action recognition accuracy of our method on the three datasets and compare the results with other methods. We demonstrate the advantage of composite attribute latent structures compared to the single attribute latent structures. Also, we compare the effects of pose information and motion information on action recognition. Furthermore, we analyze the effects of different composite state numbers. Finally, we visualize the latent structures of action sequences. The extensive experimental results demonstrate the strength of our method.

A. Action Recognition on MSR 3D Action Dataset

MSR 3D Action Dataset [9] is one of the most widely-used datasets for 3D action analysis and recognition. It contains 20 action classes and 567 action sequences, which were performed by 10 people and each people performed the same action two or three times. The 20 actions classes are: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, sideboxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup and throw*. Fig. 5(a) shows some skeleton examples of the 20 action classes. This dataset contains the data of depth image sequences and 3D skeleton sequences. We use the 3D skeleton sequences in our experiments.

TABLE I
Comparison of Action Recognition Accuracy on MSR 3D Action Dataset

Methods	Accuracy
HMM with AdaBoost [67]	0.630
Dynamic Temporal Warping [68]	0.540
Recurrent Neural Network [69]	0.425
ROP (Sparse Coding) [70]	0.862
Actionlet with Absolute Joints [14]	0.685
HDG with FAV Features[16]	0.815
Holistic HOPC [71]	0.865
Local HOPC+STK-D [71]	0.829
4DHOI with 3D Skeleton Joints [5]	0.83
Kendall's Shape Bi-LSTM [12]	0.862
Kendall's Shape FTP-SVM [12]	0.900
Our Component Latent Structure	0.872

MSR 3D Action Dataset is a challenging dataset for action recognition. The skeleton data in this dataset is noisy. The human motions in many actions are very subtle, such as *jogging* and *forward punch*. Some different actions are very similar, such as *forward kick* and *side kick*, *hand catch*, *high arm wave*, and *high throw*.

Following the cross-subject setting [14], we use the samples of half of the subjects as training data and the rest as testing data. Table I shows the comparison of action recognition accuracy on MSR 3D Action Dataset. The results of HMM with AdaBoost [67], Dynamic Temporal Warping [68], and Recurrent Neural Network [69] were reported in the work [14].

Our method achieves an accuracy of 0.872, which is better than most of the recently reported results. The HMM with AdaBoost method [67] uses Hidden Markov Model and the AdaBoost strategy to recognize 3D actions, which is a representative HMM method with single-attribute hidden states. Compared to the HMM with AdaBoost method, our method with composite latent states outperforms it by a considerable margin, which proves the advantage of the composite latent structures. The 4DHOI with 3D Skeleton Joints method [5] uses an ordered expectation maximization method to learn the substructures of actions but it updates the parameters of a probability model with maximum likelihood estimation. The performance of our method is better than the 4DHOI with 3D Skeleton Joints method, which proves the strength of our discriminative EM-like learning method.

Our method only uses human skeleton features to recognize 3D actions. In Table I, the methods HMM with AdaBoost [67], Dynamic Temporal Warping [68], Recurrent Neural Network [69], Actionlet with Absolute Joints [14], and 4DHOI with 3D Skeleton Joints [5] also use human skeleton features. Compared these skeleton-based methods, our CLS model impressively improves the action recognition accuracy. The methods ROP (Sparse Coding) [70], HDG with FAV Features[16], Holistic HOPC [71], and Local HOPC+STK-D [71] use depth or point cloud features. Compared with these methods, our method uses simpler features but achieves better performance, which proves the advantage of our method.

Kendall's Shape Bi-LSTM [12] and Kendall's Shape FTP-SVM [12] encode Kendall's shape trajectories for 3D action

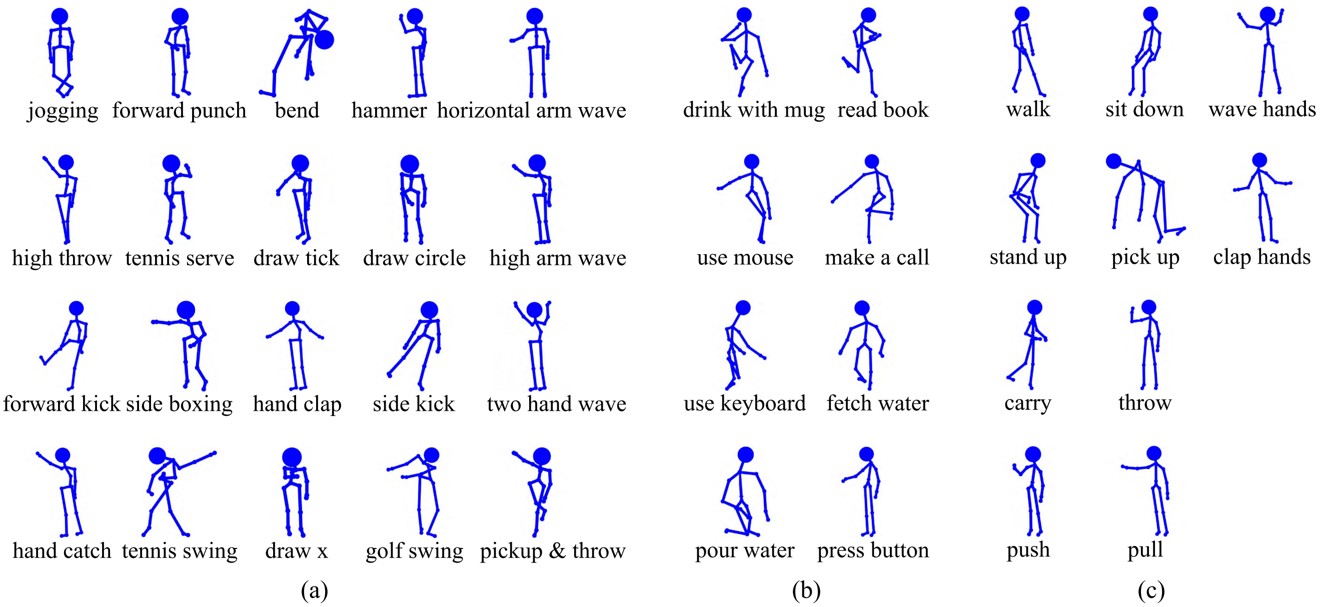


Fig. 5. Some skeleton examples in the 3D action datasets. (a) MSR 3D Action Dataset. (b) Multiview 3D Event Dataset. (c) UTKinect-Action3D Dataset.

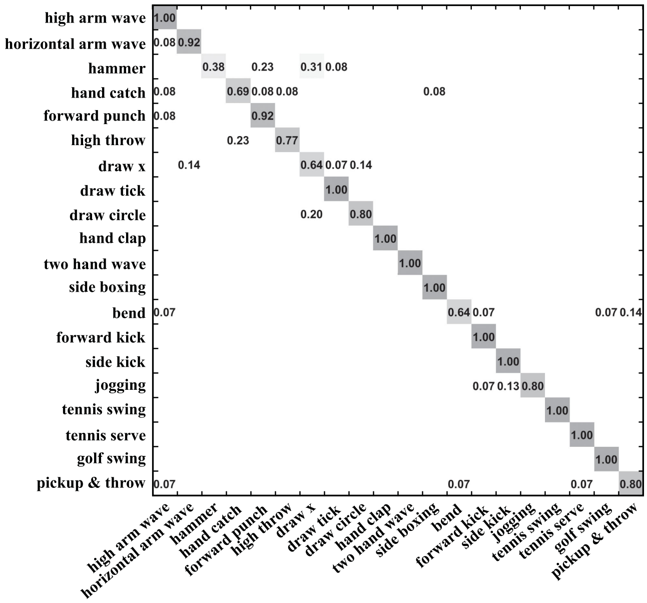


Fig. 6. Confusion matrix on MSR 3D Action Dataset.

recognition. The action recognition accuracy of our CLS is lower than that of Kendall's Shape FTP-SVM [12]. However, in addition to recognizing actions, CLS can construct the composite structure representations of actions in the temporal domain, which plays vital roles in many applications such as human-computer interaction and human animation.

Fig. 6 shows the confusion matrix of action recognition on MSR 3D Action Dataset. On many action classes, our method achieves 100% accuracy. The lowest accuracy is related to the action *hammer* and its most false positives lie in the action *draw x*. This is because these two actions are very similar, especially in the initial stages of the actions.

The major reasons that our CLS method outperforms other approaches lie in two aspects. First, our CLS method models actions with composite latent states. It jointly mines and utilizes latent semantic attributes and latent geometric attributes to represent and recognize actions. Thus it performs better than those single-attribute based methods, such as HMM with AdaBoost [67]. Second, it adopts a discriminative EM-like method to learn the composite latent structures. It can mine discriminative information and features for action recognition.

B. Action Recognition on Multiview 3D Event Dataset

Multiview 3D Event Dataset [10] is a multiview dataset with 3815 RGB-D video sequences and approximately 383,000 video frames. It was captured by three Kinect cameras at different viewpoints simultaneously around the events. The events were performed by volunteers in indoor scenes with different objects. It contains 8 event classes: *drink with mug, call with cellphone, read book, use mouse, type on keyboard, fetch water from dispenser, pour water from kettle*, and *press button*, and these events are related to 11 object classes: *mug, cellphone, book, mouse, keyboard, dispenser, kettle, button, monitor, chair*, and *desk*. This dataset contains RGB video sequences, depth video sequences, and 3D skeleton sequences. In our experiment, we only use the 3D skeleton data. Fig. 5(b) shows some skeleton examples of the eight event classes.

One major characteristic of this dataset is that it contains abundant functional object interactions. The functional object interactions bring two major challenges. First, the human bodies are always occluded by the objects, which makes the skeleton data very noisy and unstable. Second, most events are mainly defined by the functional object information rather than the human pose and motion information. For example, the events *drink with mug* and *call with cellphone* are differentiated relying on the objects *mug* and *cellphone*, respectively, since

TABLE II
ACTION RECOGNITION ON MULTIVIEW 3D EVENT DATASET

Methods	Accuracy
Motion Templates [68]	0.69
Hidden Markov Model [72]	0.74
4DHOI with 3D Joints [5]	0.87
Our Component Latent Structure	0.914

the human poses and motions in these two events are highly similar. These challenges make it very difficult to recognize the events with skeleton information. Other characteristics of this dataset are that it is multiview and large-scale with 3815 videos and approximately 383,000 frames. These characteristics make Multiview 3D Event Dataset very challenging for event recognition.

In experiments, we extract features from each view independently and do not use the multi-view information. Table II shows the comparison of action recognition accuracy on the Multiview 3D Event Dataset. We compare our CLS method with the Motion Templates method [68], original Hidden Markov Model [72], and 4DHOI method [5]. The results of Motion Templates [68] and Hidden Markov Model [72] are cited from the work [5]. Motion Templates method [68] trains templates for events and matches the testing sequences with dynamic temporal warping. Hidden Markov Model [72] uses human skeleton joints as inputs and trains an HMM for each event category. 4DHOI with 3D Joints method [5] uses the 3D human skeleton joint features to recognize events.

As shown in Table II, our method achieves much higher recognition accuracy than the Motion Templates and Hidden Markov Model methods. Compared to the 4DHOI method, when only using 3D human skeleton joint features, our CLS model outperforms 4DHOI by a considerable margin in accuracy. This is a decent result for our CLS method considering that most of the events in the Multiview 3D Event Dataset are mainly defined by functional object information.

The Motion Templates method [68] utilizes simple templates to recognize actions and does not mine the temporal structures of actions. Though the Hidden Markov Model [72] mines the temporal structures of actions, it models the temporal structures with single attributes. Thus, our CLS method which models temporal structures of actions with composite attributes outperforms the two comparison methods. Our CLS explicitly describes the motion features of skeleton sequences and adopts a discriminative EM-like learning algorithm, which make CLS outperform the 4DHOI with 3D Joints [5].

Fig. 7 shows the confusion matrix of action recognition on Multiview 3D Event Dataset. Most of the action classes achieve an accuracy above 90% except for one action class *drink with mug*. The most false positives of *drink with mug* lie in the action class *make a call* since the poses and motions in the action classes *drink with mug* and *make a call* are very similar. On the other hand, the acceptable accuracy of the action class *make a call* shows most samples are differentiated from the action *drink with mug*. This proves the effectiveness of our recognition method and skeleton features.

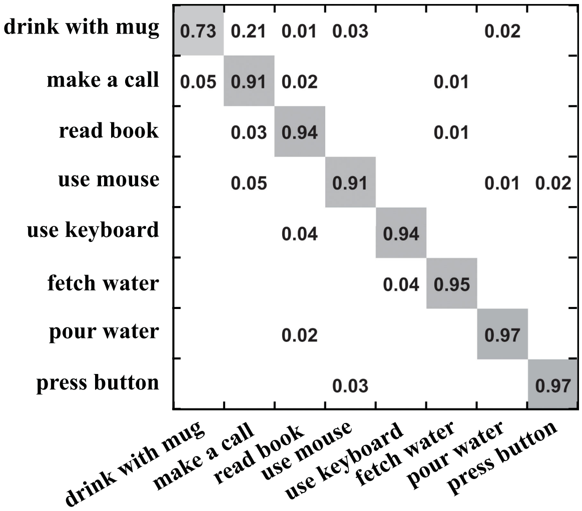


Fig. 7. Confusion matrix on Multiview 3D Event Dataset.

TABLE III
ACTION RECOGNITION ON UTKINECT-ACTION3D DATASET

Methods	Accuracy
Hanklet-Based HMM [73]	0.868
MSD-HMM with 3D Skeletons [32]	0.895
Histograms of 3D Joints [11]	0.909
HIF 3D skeleton [74]	0.940
Spatio-Temporal Feature Chain [75]	0.915
Grassmann Manifold [76]	0.885
3D Key Pose Motifs [33]	0.935
Motion Trajectories on Manifold [77]	0.915
Kendall's Shape Bi-LSTM [12]	0.985
Kendall's Shape FTP-SVM [12]	0.975
Our Component Latent Structure	0.955

C. *Action Recognition on UTKinect-Action3D Dataset*

UTKinect-Action3D Dataset [11] contains 10 action classes: *walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands*. These actions were performed by 10 subjects and each subject performed each action two times. It contains RGB, depth, and 3D skeleton joint sequences at 15 fps. Some frames are discontinuous due to the missing human skeletons. Fig. 5(c) shows some skeleton examples of the 10 action classes.

Many action classes in UTKinect-Action3D Dataset contain repeating body motion structures, such as *walk, clap hands*, and *carry*. Such repeating motion structures make the action structure modeling and learning very difficult. This dataset also contains inverse action pairs, in which the actions have similar poses but inverse motion processes, such as *sit down* and *stand up*, *push* and *pull*. Such inverse action pairs bring challenges to both action structure learning and action recognition.

Following the experiment setting in the work [11], we adopted a leave one sequence out cross validation (LOOCV) to compute the action recognition accuracy. Table III shows the comparison of action recognition accuracy on this dataset.

On this dataset, our method achieves an accuracy of 0.955, which outperforms most of the comparison methods in Table III.

TABLE IV
ACTION RECOGNITION ACCURACY OF COMPOSITE-ATTRIBUTE LATENT STATE MODEL AND SINGLE-ATTRIBUTE LATENT STATE MODEL

Datasets	Single attribute: Random intervals	Single attribute: Uniform intervals	Our composite attributes
MSR 3D Action	0.626	0.775	0.872
Multiview 3D Event	0.849	0.904	0.914
UTKinect-Action3D	0.889	0.919	0.955

The UTKinect-Action3D Dataset was released in the work Histograms of 3D Joints [11], which reported a recognition accuracy of 0.909. The comparison methods Hanklet-Based HMM [73], MSD-HMM with 3D Skeletons [32], HIF 3D skeleton [74], Spatio-Temporal Feature Chain [75], Grassmann Manifold [76], 3D Key Pose Motifs [33], Motion Trajectories on Manifold [77], Kendall's Shape Bi-LSTM [12], and Kendall's Shape FTP-SVM [12] use 3D skeleton sequences and achieve recognition accuracies of 0.868, 0.895, 0.940, 0.915, 0.885, 0.935, 0.915, 0.985, and 0.975, respectively.

Our CLS method utilizes composite latent states to represent and recognize actions, which can better characterize the latent semantic and geometric information of actions. Therefore, our CLS method performs better than those single-attribute based methods, such as Hanklet-Based HMM [73] and MSD-HMM with 3D Skeletons [32]. The UTKinect-Action3D Dataset contains many actions with similar poses but inverse motions and most previous methods did not explicitly characterize the motions features. Our CLS method explicitly combines the pose and motion features, which is another reason why our CLS method performs better.

For action recognition accuracy, deep learning methods outperform our CLS method. However, compared to deep learning methods, CLS does not need powerful graphics processing units for training and inference, which is an appealing characteristic for applications on portable, mobile, or wearable devices. Furthermore, CLS is used not only for action recognition but also for action representation. CLS can construct composite structure representations of human actions, which plays essential roles in many applications such as human-computer interaction and human animation. Deep learning methods normally aim to predict action classes rather than construct temporal structure representations of actions.

Fig. 8 shows the confusion matrix of action recognition on UTKinect-Action3D Dataset. On many action classes, our method achieves an accuracy of 100%. On only one action class *throw*, the accuracy is under 90%. The most false positives of the action *throw* lie in the action classes *wave hands* and *clap hands* since the pose and motion features are highly similar in these action classes.

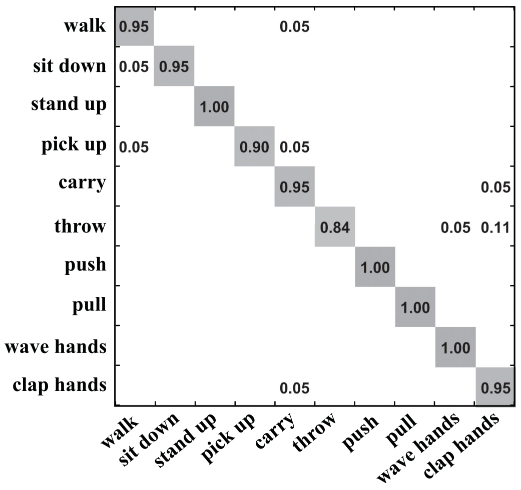


Fig. 8. Confusion matrix on UTKinect-Action3D Dataset.

D. Comparison of Composite Attributes and Single Attributes

We compare the effects of composite-attribute latent states and single-attribute latent states on action recognition. Two types of single-attribute models are compared. The first one is the random interval method, in which the latent interval variables of atomic actions are randomly set. The second one is the uniform interval method, in which the latent interval variables of atomic actions are set to be uniform, i.e. each atomic action in a sequence has equal length. For fair comparison, these two types of single-attribute state models adopt the same sequence features, learning, and inference methods. The only difference between CLS and those two single-attribute models is that the latent interval variables of CLS are mined and optimized while those two methods use random or uniform interval variables.

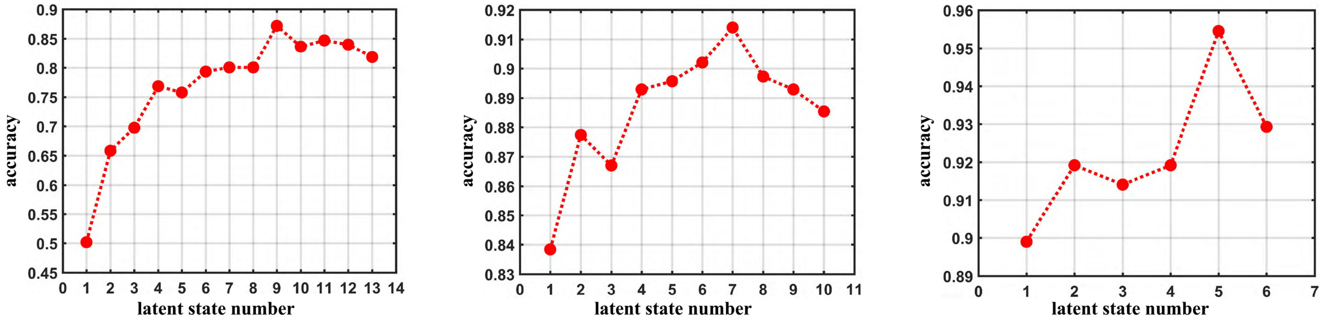
Table IV shows the action recognition accuracy of our CLS model and single-attribute models on three testing datasets. On the MSR 3D Action Dataset and the UTKinect-Action3D Dataset, the performance of our composite-attribute model is much better than the single-attribute models. This is because our composite-attribute model takes advantage of the latent temporal structure information. These results strongly demonstrate the advantage of the composite latent structures over the single-attribute latent structures in action recognition.

On the Multiview 3D Event dataset, the single-attribute model of uniform intervals achieves a comparable accuracy with our composite-attribute model (0.904 vs 0.914). This is because the human motions in Multiview 3D Event dataset have stable temporal structures and procedures. The uniform intervals characterize the stable temporal structures and procedures to some extent.

Table IV also shows that for the single-attribute model the uniform intervals outperform the random intervals on action recognition. This phenomenon further proves the importance and necessity of mining the atomic action's geometric information because the uniform intervals, at any rate, contain more useful information than the random intervals.

TABLE V
ACTION RECOGNITION ACCURACY WITH POSE FEATURES AND MOTION FEATURES

Datasets	Pose features	Motion features	Our pose-motion features
MSR 3D Action	0.794	0.865	0.872
Multiview 3D Event	0.883	0.896	0.914
UTKinect-Action3D	0.919	0.949	0.955



(a) (b) (c)

Fig. 9. Relations between the latent state number and the action recognition accuracy. (a) On MSR 3D Action Dataset. (b) On Multiview 3D Event Dataset. (c) On UTKinect-Action3D Dataset.

E. *Comparison of Pose Features and Motion Features*

As Section III-A presented, our action features extracted from 3D skeletons consist of pose parts and motion parts. These two types of features play different roles in actions. In this experiment, we compare the effects of pose information and motion information on action recognition. Table V shows the action recognition accuracy of our CLS model using pose features or motion features on the three testing datasets.

Table V shows that on all three datasets, the method using the combined features of poses and motions outperforms the methods only using one type of features. It proves the effectiveness of the pose-motion features. The combined features outperform the poses features a considerable margin but outperform the motion features a smaller margin. This proves the importance of motion information in action recognition. It does not necessarily imply that the pose information is insignificant. Indeed, the motion features are composed of the relative differences of each joint with other joints, which implicitly embodies the pose information.

F. *Effects of Latent State Numbers on Action Recognition*

In our CLS model, a sequence is divided into m composite latent states. The large m means a sophisticated representation ability but low robustness to noise. In this experiment, we compare the effects of the composite latent state number m on action recognition. Fig. 9 shows the action recognition accuracy as m varies on the three datasets.

A common observation in these three figures is that before a state number threshold is achieved, the action recognition rises as the latent state number increases. After that threshold, the action recognition falls as the latent state number increases. This phenomenon implies that there exists an optimal latent structure division to interpret an action. Too large or too small state numbers would lead to misleading interpretations.

Another observation is that the optimal state number thresholds on these three datasets are different. They are 9, 7, and 5 on the MSR 3D Action Dataset, Multiview 3D Event Dataset, and the UTKinect-Action3D Dataset, respectively. This phenomenon reflects the different characteristics of the action structures on the three datasets. The actions in the MSR 3D Action Dataset have more complex temporal structures while the actions in the UTKinect-Action3D Dataset have repetitive substructures. The actions in the Multiview 3D Event Dataset have normative procedures.

G. *Visualization of Latent Action Structures*

With our CAIP inference algorithm, we can infer the latent structures of action sequences. As basic action units, the representation of these latent structures can be used in many applications, such as human-computer interaction and human-robot collaboration. Fig. 10 shows some examples of the latent structures in actions. These latent structures are obtained as byproducts in the inference of action recognition. For each latent atomic action, we draw the average skeleton in the corresponding latent interval.

For some actions, the mined latent structures are consistent with humans' semantic experience. For example, in the sixth example, the seven latent states in the action *drink with mug* represent the sequential procedures with semantic meanings. For some actions, the mined latent structures are not necessarily with semantic meanings, such as some of the latent states in the tenth action example *tennis serve*. Whether with semantic meanings or not, these latent structures represent the action interpretations mined from data. They provide new perspectives to look into human actions.

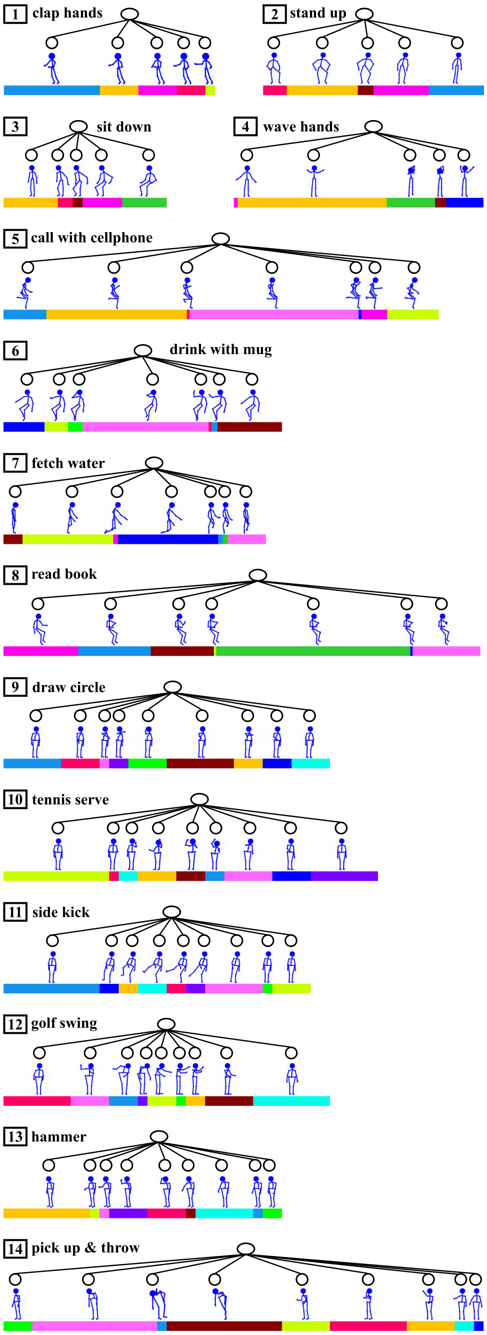


Fig. 10. Latent structure visualization of some actions. The colorful segments denote the time intervals of latent structures. The same color in different actions does not necessarily mean the same latent state.

Our algorithm can also mine the duration information of the atomic actions. For example, in the fifth example *call with cellphone*, the fourth atomic action probably denotes *hold the cellphone to call*. Its duration is much longer than other atomic actions. The atomic action *dial the number* also has a long duration while *fetch cellphone* has a short duration.

VII. Conclusion

In this paper, we propose a composite latent structure (CLS) model to recognize 3D human actions and construct the latent structures of actions. A human action sequence is divided into several sequential temporal intervals, each of which corresponds to an atomic action. An atomic action is a composite latent state which includes the latent semantic attribute and the latent geometric attribute. The hierarchical structure of an action, the atomic actions, and the sequence data is represented with a hierarchical graph. A discriminative EM-like algorithm is proposed to mine the composite latent structures of actions and learn the model parameters. Given a 3D human skeleton sequence, a composite attribute iterative programming algorithm is proposed to optimize the composite latent structures and recognize the human action.

We test the proposed method on three challenging and representative 3D human action datasets. We compare the action recognition accuracy of our method with other representative methods. We also analyze the effects of the composite attributes and the latent structure numbers on action recognition. Moreover, we compare the effects of pose features and motion features. Finally, we visualize the composite latent structures of action sequences.

In this work, we focus on action representation and recognition with composite latent structures. However, our CLS model can be potentially extended to other tasks, such as object recognition, scene understanding, and sequence generation.

In the future work, we will investigate the combination of CLS with deep learning methods to represent human actions and improve action recognition performance. We will also explore the potential of CLS in other tasks.

References

[1] F. Lv and R. Nevatia, “Recognition and segmentation of 3-D human action using HMM and multi-class adaboost,” in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 359–372.

[2] A. Quattoni, S. B. Wang, L.-P. Morency, M. Collins, and T. Darrell, “Hidden conditional random fields,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1848–1852, Oct. 2007.

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

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

[5] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu, “Modeling 4D human-object interactions for joint event segmentation, recognition, and object localization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1165–1179, Jun. 2017.

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

[7] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, 1989, pp. 257–286.

[8] J. Shotton *et al.*, “Real-time human pose recognition in parts from single depth images,” in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1297–1304.

[9] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3D points,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.—Workshops*, 2010, pp. 9–14.

[10] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu, “Modeling 4D human-object interactions for event and object recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3272–3279.

[11] L. Xia, C.-C. Chen, and J. K. Aggarwal, “View invariant human action recognition using histograms of 3D joints,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 20–27.

[12] A. Ben Tanfous, H. Drira, and B. Ben Amor, “Coding Kendall’s shape trajectories for 3D action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2840–2849.

[13] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, “A new representation of skeleton sequences for 3D action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4570–4579.

[14] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Learning action let ensemble for 3D human action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, May 2014.

[15] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu, “Concurrent action detection with structural prediction,” in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 3136–3143.

[16] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, “Real time action recognition using histograms of depth gradients and random decision forests,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 626–633.

[17] X. Yang, C. Zhang, and Y. Tian, “Recognizing actions using depth motion maps-based histograms of oriented gradients,” in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 1057–1060.

[18] X. Yang and Y. Tian, “Super normal vector for activity recognition using depth sequences,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 804–811.

[19] Y. Zhu, W. Chen, and G. Guo, “Fusing spatiotemporal features and joints for 3D action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2013, pp. 486–491.

[20] C. Jia and Y. Fu, “Low-rank tensor subspace learning for RGB-D action recognition,” *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4641–4652, Oct. 2016.

[21] R. Anirudh and P. Turaga, “Geometry-based symbolic approximation for fast sequence matching on manifolds,” *Int. J. Comput. Vis.*, vol. 116, no. 2, pp. 161–173, Jan. 2016.

[22] B. B. Amor, J. Su, and A. Srivastava, “Action recognition using rate-invariant analysis of skeletal shape trajectories,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 1–13, Jan. 2016.

[23] J. Luo, W. Wang, and H. Qi, “Group sparsity and geometry constrained dictionary learning for action recognition from depth maps,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1809–1816.

[24] L. Xia and J. Aggarwal, “Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2834–2841.

[25] B. Liang and L. Zheng, “Specificity and latent correlation learning for action recognition using synthetic multi-view data from depth maps,” *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5560–5574, Dec. 2017.

[26] M. Liu, H. Liu, and C. Chen, “Robust 3D action recognition through sampling local appearances and global distributions,” *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1932–1947, Aug. 2018.

[27] J. F. Hu, W. S. Zheng, J. H. Lai, and J. Zhang, “Jointly learning heterogeneous features for RGB-D activity recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2186–2200, Nov. 2017.

[28] B. Zhang *et al.*, “Action recognition using 3D histograms of texture and a multi-class boosting classifier, ”*IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4648–4660, Oct. 2017.

[29] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang, “Multimodal multipart learning for action recognition in depth videos,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2123–2129, 2016.

[30] Y. Yang *et al.*, “Discriminative multi-instance multitask learning for 3D action recognition, ”*IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 519–529, Mar. 2017.

[31] X. Cai, W. Zhou, L. Wu, J. Luo, and H. Li, “Effective active skeleton representation for low latency human action recognition,” *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 141–154, Feb. 2016.

[32] G. Borghi, R. Vezzani, and R. Cucchiara, “Fast gesture recognition with multiple stream discrete HMMs on 3D skeletons,” in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 997–1002.

[33] C. Wang, Y. Wang, and A. L. Yuille, “Mining 3D key-pose-motifs for action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2639–2647.

[34] P. Cui, F. Wang, L.-F. Sun, J.-W. Zhang, and S.-Q. Yang, “A matrix-based approach to unsupervised human action categorization,” *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 102–110, Feb. 2012.

[35] J.-C. Cheng and J. M. F. Moura, “Capture and representation of human walking in live video sequences,” *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 144–156, Jun. 1999.

[36] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona, “Depth pooling based large-scale 3-D action recognition with convolutional neural networks,” *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1051–1061, May 2018.

[37] P. Shukla, K. K. Biswas, and P. K. Kalra, “Recurrent neural network based action recognition from 3D skeleton data,” in *Proc. 13th Int. Conf. Signal-Image Technol. Int.-Based Syst.*, Dec. 2017, pp. 339–345.

[38] S. Wei, Y. Song, and Y. Zhang, “Human skeleton tree recurrent neural network with joint relative motion feature for skeleton based action recognition,” in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 91–95.

[39] P. Zhang *et al.*, “View adaptive recurrent neural networks for high performance human action recognition from skeleton data,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2136–2145.

[40] Y. Hou, Z. Li, P. Wang, and W. Li, “Skeleton optical spectra-based action recognition using convolutional neural networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 807–811, Mar. 2018.

[41] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[42] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1110–1118.

[43] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” *Adv. Neural Inf. Process. Syst.*, vol. 1, no. 4, pp. 568–576, 2014.

[44] J. Hou, X. Wu, Y. Sun, and Y. Jia, “Content-attention representation by factorized action-scene network for action recognition,” *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1537–1547, Jun. 2018.

[45] Y. Shi, Y. Tian, Y. Wang, and T. Huang, “Sequential deep trajectory descriptor for action recognition with three-stream CNN,” *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1510–1520, Jul. 2017.

[46] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, “Two-stream 3-D convnet fusion for action recognition in videos with arbitrary size and length,” *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 634–644, Mar. 2018.

[47] B. Yao *et al.*, “Human action recognition by learning bases of action attributes and parts, ”in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1331–1338.

[48] X. Cai, W. Zhou, and H. Li, “Attribute mining for scalable 3D human action recognition,” in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1075–1078.

[49] Y. Su, P. Jia, A.-A. Liu, and Z. Yang, “Discovering latent attributes for human action recognition in depth sequence,” *Electron. Lett.*, vol. 50, no. 20, pp. 1436–1438, 2014.

[50] Y. Pei, B. Ni, and I. Atmosukarto, “Mixture of heterogeneous attribute analyzers for human action detection,” in *Proc. Eur. Conf. Comput. Vis. Workshops*, L. Agapito, M. M. Bronstein, and C. Rother, Eds., 2015, pp. 528–540.

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

[52] B. X. Nie, C. Xiong, and S.-C. Zhu, “Joint action recognition and pose estimation from video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1293–1301.

[53] L. Lin *et al.*, “A deep structured model with radius–margin bound for 3D human activity recognition, ”*Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 256–273, Jun. 2016.

[54] F. Liu, X. Xu, S. Qiu, C. Qing, and D. Tao, “Simple to complex transfer learning for action recognition,” *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 949–960, Feb. 2016.

[55] F. Zhou, F. D. la Torre, and J. K. Hodgins, “Hierarchical aligned cluster analysis for temporal clustering of human motion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, Mar. 2013.

[56] J. Wang and Y. Wu, “Learning maximum margin temporal warping for action recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2688–2695.

[57] N. Hu, G. Englebienne, Z. Lou, and B. Krse, “Latent hierarchical model for activity recognition,” *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1472–1482, Dec. 2015.

[58] I. Lillo, A. Soto, and J. C. Niebles, “Discriminative hierarchical modeling of spatio-temporally composable human activities,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 812–819.

[59] I. Lillo, J. C. Niebles, and A. Soto, "A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1981–1990.

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

[61] Y. Cheng, Q. Fan, S. Pankanti, and A. Choudhary, "Temporal sequence modeling for video event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2235–2242.

[62] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah, "Recognition of complex events: Exploiting temporal dynamics between underlying concepts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2243–2250.

[63] M. Pei, Z. Si, B. Z. Yao, and S.-C. Zhu, "Learning and parsing video events with goal and intent prediction," *Comput. Vis. Image Understanding*, vol. 117, pp. 1369–1383, 2013.

[64] X. Zhen, F. Zheng, L. Shao, X. Cao, and D. Xu, "Supervised local descriptor learning for human action recognition," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2056–2065, Sep. 2017.

[65] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[66] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[67] F. Lv and R. Nevatia, "Recognition and segmentation of 3-D human action using HMM and multi-class adaboost," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 359–372.

[68] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2006, pp. 137–146.

[69] J. Martens and I. Sutskever, "Learning recurrent neural networks with hessian-free optimization," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1033–1040.

[70] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 872–885.

[71] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, "Histogram of oriented principal components for cross-view action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 12, pp. 2430–2443, Dec. 2016.

[72] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[73] L. Lo Presti, M. La Cascia, S. Sclaroff, and O. Camps, "Gesture modeling by hanklet-based hidden Markov model," in *Proc. 12th Asian Conf. Comput. Vis.*, 2014, pp. 529–546.

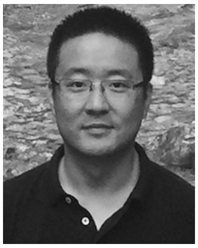
[74] S. Y. Boulahia, E. Anquetil, R. Kulpa, and F. Multon, "HIF3D: Handwriting-inspired features for 3D skeleton-based action recognition," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 985–990.

[75] W. Ding, K. Liu, F. Cheng, and J. Zhang, "STFC: Spatio-temporal feature chain for skeleton-based human action recognition," *J. Visual Commun. Image Representation*, vol. 26, pp. 329–337, 2015.

[76] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3D action recognition using learning on the Grassmann manifold," *Pattern Recognit.*, vol. 48, no. 2, pp. 556–567, 2015.

[77] M. Devanne *et al.*, "3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1340–1352, Jul. 2015.

Ping Wei received the B.S. and Ph.D. degrees from Xi'an Jiaotong University, Xian, China. He is currently an Associate Professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. He has been a Postdoctoral Researcher with the Center for Vision, Cognition, Learning, and Autonomy, University of California, Los Angeles, CA, USA, from 2016 to 2017. His research interests include computer vision, machine learning, and computational cognition. Dr. Wei serves as a Co-Organizer of the International Workshop on Vision Meets Cognition: Functionality, Physics, Intents and Causality at CVPR 2017 and 2018, respectively.



Hongbin Sun received the B.S. and Ph.D. degrees in electrical engineering from Xian Jiaotong University, Xian, China, in 2003 and 2009, respectively. He was a Visiting Ph.D. Student with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA, from 2007 to 2008. He was a Postdoc with the Department of Computer Science, Xian Jiaotong University, from 2009 to 2011. He is currently a Professor with the School of Electronic and Information Engineering, Xian Jiaotong University. His current research interests include integrated circuit and system for computing, memory, and video signal processing.

Nanning Zheng received the Ph.D. degree from Keio University, Tokyo, Japan, in 1985. He is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, computational intelligence, image processing, and hardware implementation of intelligent systems. Dr. Zheng has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition since 2000. He became a member of the Chinese Academy of Engineering in 1999.