IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 1, 2019, accepted November 10, 2019, date of publication November 20, 2019,
date of current version December 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954744

Skeleton-Based Action Recognition With Key-Segment Descriptor and Temporal Step Matrix Model

RUIMIN LI[1,2,3], HONG FU[3], WAI-LUN LO[3], (Senior Member, IEEE), ZHERU CHI[4], (Member, IEEE), ZONGXI SONG[1], AND DESHENG WEN[1]

[1]Xi'an Institute of Optics and Precision Mechanics, CAS, Xi'an 710119, China
[2]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Department of Computer Science, Chu Hai College of Higher Education, Hong Kong
[4]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Hong Fu (hongfu@chuhai.edu.hk)

The work described in this article was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Reference No.: UGC/FDS13/E02/16).

ABSTRACT Human action recognition based on skeleton has played a key role in various computer vision-related applications, such as smart surveillance, human-computer interaction, and medical rehabilitation. However, due to various viewing angles, diverse body sizes, and occasional noisy data, etc., this remains a challenging task. The existing deep learning-based methods require long time to train the models and may fail to provide an interpretable descriptor to code the temporal-spatial feature of the skeleton sequence. In this paper, a key-segment descriptor and a temporal step matrix model are proposed to semantically present the temporal-spatial skeleton data. First, a skeleton normalization is developed to make the skeleton sequence robust to the absolute body size and initial body orientation. Second, the normalized skeleton data is divided into skeleton segments, which are treated as the action units, combining 3D skeleton pose and the motion. Each skeleton sequence is coded as a meaningful and characteristic key segment sequence based on the key segment dictionary formed by the segments from all the training samples. Third, the temporal structure of the key segment sequence is coded into a step matrix by the proposed temporal step matrix model, and the multiscale temporal information is stored in step matrices with various steps. Experimental results on three challenging datasets demonstrate that the proposed method outperforms all the hand-crafted methods and it is comparable to recent deep learning-based methods.

INDEX TERMS Skeleton-based action recognition, view alignment, scale normalization, key-segment descriptor, temporal step matrix model.

I. INTRODUCTION

Human action recognition has become an important research topic in the field of computer vision, and has attracted considerable interest in the past few decades [1]–[6] due to its wide range of applications in smart surveillance, human-computer interaction and medical rehabilitation. Many action recognition methods have been developed on various data sources, including RGB video, skeleton data, depth maps or some combinations of these modalities. The skeleton data consists of the 3D coordinates of the extracted key joints in the human body over time, representing a human body as an articulated system as shown in Fig.1 (a). With the development of the 3D capturing technology, such as Microsoft Kinect, 3D data collection and real-time skeleton extraction become much easier and more accurate for practical applications related to human action analysis. Compared with other modalities, skeleton data have a higher-level representation with less complexity and they are more robust to noise like background, illumination and human face. In this paper, we focus on the skeleton-based action recognition method.

There are many past researches for the skeleton-based action recognition [7]–[15]. The key to a successful action recognition is to extract an effective spatio-temporal feature of the skeleton sequence. Most methods [10], [11], [14] divide the extraction of spatio-temporal features into

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

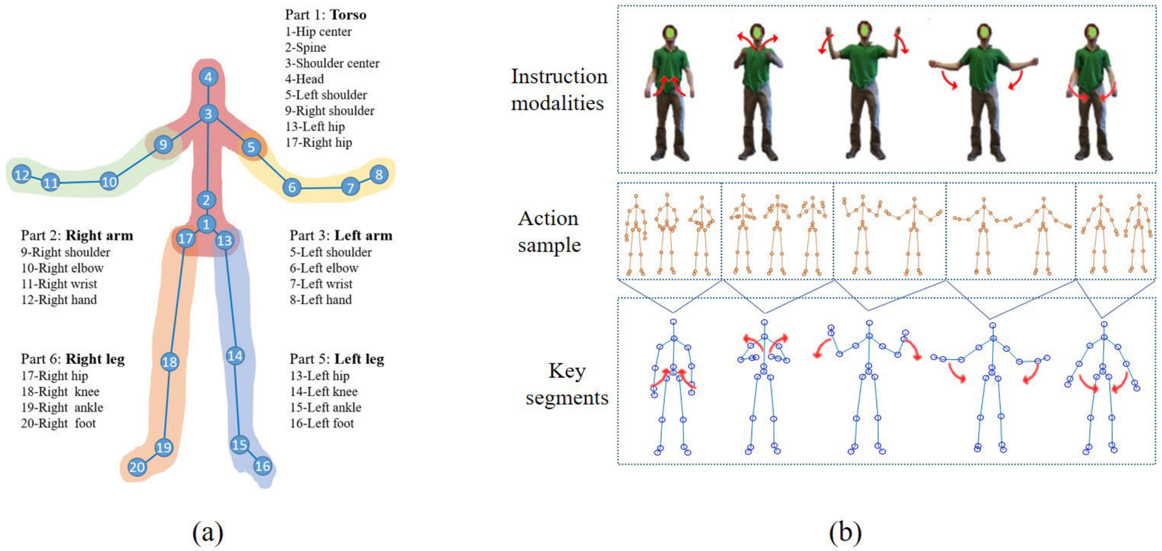


FIGURE 1. (a) Skeleton model adopted by Kinect1, which has 20 joints. These 20 skeleton joints are divided into five parts: torso, left arm, right arm, left leg, and right leg. (b) Action "Wind up the music" in the MSRC-12 dataset [16]. The first row is instruction modality consisting of static images and arrows, which is shown to subjects and the subjects perform the corresponding actions [16]. The second row is an ordered skeleton sequence of an action "Wind up the music" sample and the last row are the key segments of this action sample. A segment is presented by a key frame with motion direction expressed by the red arrows. And skeletons in the second row separated by dashed lines belong to different segments.

two independent steps: extract the spatial information of the frame-level skeleton, and establish a dynamic model to code the temporal information. The features obtained by the above methods have disadvantages, such as spatial redundancy and single temporal scale information. There is a hypothesis of these methods: each frame plays an equal role in action recognition. However, not every frame provides absolutely valid information in an action sequence. And the spatial information of adjacent frames is often redundant. It is more reasonable to treat several continuous adjacent frames rather than a separate frame as an action unit, thus the spatial features and local temporal features of the skeleton sequence are simultaneously encoded. The dynamic model encodes the temporal information at frame-level after obtaining the spatial features of each frame. Since different action performers have different motion speeds, the temporal information at a single scale cannot well represent the temporal information of the skeleton sequence. Therefore, a multiscale temporal model should be proposed to encode the most essential temporal information of a skeleton sequence. Although the recent emerging deep learning-based methods [9], [12], [13] code the spatio-temporal information simultaneously and work well in action recognition, but these approaches have no interpretable physical meaning.

In this paper, we propose an effective yet simple skeleton sequence representation based on a sequence of atomic action units, namely, skeleton segments, which consist of several consecutive skeleton frames whose spatial variation is relatively small. The normalization preprocessing is firstly carried out on the skeleton sequence, which makes the transformed skeleton sequence robust to the absolute body size and initial body orientation. Then each skeleton sequence is divided into multiple skeleton segments based on the designed segmentation scheme as shown in Fig.1 (b). This is just like that human can distinguish and perform different actions as long as some key static images sequence with the direction of movement, which can be regarded as the feature of a segment, were given. For example, when building the MSRC-12 dataset, the instructor shows subjects an ordered series of static images of some actions with arrows annotating motion direction as appropriate as shown in Fig.1 (b), and subjects perform the actions according to this instruction modalities. Therefore, the ordered sequence of key segments can be considered as a more representative representation of an action. Finally, the temporal structure of the key segment sequence is coded into a step matrix by the proposed temporal step matrix model, and the multiscale temporal information is stored in step matrices with various steps. In this way, the spatial and local temporal information is coded in the key segment descriptor and the multiscale global temporal information is preserved in the temporal step matrix.

Generally, our method has the following contributions:

- A view alignment and a scale normalization are designed to effectively deal with the various body sizes and different body orientations.
- An interpretable and discriminative descriptor based on key-segment is proposed to code a skeleton sequence as a series of key segments, which retain both the spatial and local temporal information.
- A temporal step matrix model is developed to code the multiscale global temporal information of the key-segment descriptor.

- The proposed method is evaluated on the Northwestern-UCLA dataset, the MSRC-12 dataset and the CAD-60 dataset, and the experimental results show that our method outperforms all the hand-crafted methods and it is comparable to most deep learning-based methods.

The rest of this paper is organized as follows. In section 2, a brief review of previous works related to skeleton-based action recognition is presented. Then in Section 3, the main body of the proposed method is described in detail: skeleton normalization, key-segment based descriptor, and temporal step matrix model. The experimental results and discussions are reported in section 4. Finally, we conclude the paper in Section 5.

II. RELATED WORKS

There are plenty of works on skeleton-based human action recognition in computer vision literature. Here, we review the most relevant previous works related to our work on three aspects: skeleton normalization, key-pose based descriptor and dynamic model.

A. SKELETON NORMALIZATION

In the real application scene, the action recognition system needs to be robust to various body sizes, body positions, body orientations, and different viewpoints, so the skeleton sequence needs to be normalized to diminish the effect of the scale and view variances. The general normalization method is to transform the skeleton coordinates into a new standard coordinate system to achieve view and scale alignment. In [11] and [17], a new coordinate system was build and the original skeleton data were transformed into the new coordination system, where the hip center was defined as the origin, the horizontal x-axis was aligned with the vector whose direction from left hip joint to right hip joint, and the z-axis was defined as a vector that passed through the new origin o and was perpendicular to the new ground plane. The coordinate systems designed in [11] and [17] are based on such a hypothesis: the original skeletons is perpendicular to the ground plane. However, such a hypothesis is unreasonable for certain actions such as, bending over. Then in [18], Raptis et al. proposed a more flexible solution to build the new coordinate system, where the zenith reference vector was selected as the first principal component of the torso points which was always aligned with the longer dimension of the human torso. The above methods are all skeleton-based transformation, that is, each frame is transformed to its own new spatial coordinate system, thus destroying the motion characteristic between frames. In [19], Chen et al. proposed to transform the skeleton data into the same coordinate system by randomly selecting one skeleton as the standard skeleton. And Lee *et al.* [20] also proposed to transformed the overall skeleton sequence by the information obtained from the first frame of the skeleton sequence. The coordinate system designed in [19] and [20] are based on a single skeleton frame, rather than take advantage of all the skeleton sequence. Different from the above works, Liu *et al.* [12] transformed the skeleton sequence into a unified spatial coordinate system, which is based on the information of the entire skeleton sequence. However, this method only consider view variances and ignore scale variances.

B. KEY-POSE BASED DESCRIPTOR

The key-pose based descriptor means to learn a dictionary of key poses and represent an action sequence using these key poses. In [21], skeleton frames were clustered into K representative poses in terms of the 3D joint locations and an action sequence was represented by the histogram of its pose words. In [22], all poses composing actions were grouped into a set of clusters in terms of the Hausdorff distance and the median element in the cluster was defined as the representative pose of this cluster. However, the methods [21], [22] based on the histogram of action pose extract and utilize statistical features, while lacking temporal information. Then in [23], Lillo et al. proposed a hierarchical model of three sematic levels to describe complex human activity. Single human action was represented as a mixture of encoded poses at the intermediate level. In [14], Miranda et al. proposed a scheme for real-time action recognition based on decision forests in which every forest node was a key pose and the leaves were action labels. A similar action descriptor was proposed in [18], in which a multi-class SVM was used to detect key poses and a decision forest was adopted to recognize human action from the sequence of key-poses. In [24], Zhou et al. proposed to learn a pose lexicon which consists of semantic poses and the corresponding visual poses. In this way, action recognition was cast as a problem of finding the most appropriate translation of a semantic pose sequence with the learned pose lexicon. In the above works, key poses are generated based on frame-wise clustering and then each frame of the original skeleton is arranged a key pose to form the descriptor of the skeleton sequence. What's more, the key-pose based action descriptor lacks local temporal information because the generated key poses include the static spatial information of the current frame while ignoring the temporal motion. Key poses with the contextual motion information from a segment were not considered in the above works, which is more representative and effective.

C. DYNAMIC MODEL

Many skeleton-based action recognition methods focus on modeling the dynamics of the skeleton sequence. Majority of the approaches use Linear Dynamical Systems (LDS) [21], [25], [26], Hidden Markov Models(HMMs) [11], or dynamic forest model [14], [18] to represent the dynamic of the skeleton sequence. In [25], each action sequence was represented as a linear dynamic system that had produced the 3D joint trajectories.In [21], Ofli et al. proposed to use linear dynamical system parameters (LDSP) to model the dynamic motion cues among the sequence of the most informative joints. In [11], the temporal evolutions were modeled by Discrete Hidden Markov Models (DHMMs). In [14], Miranda et al. proposed

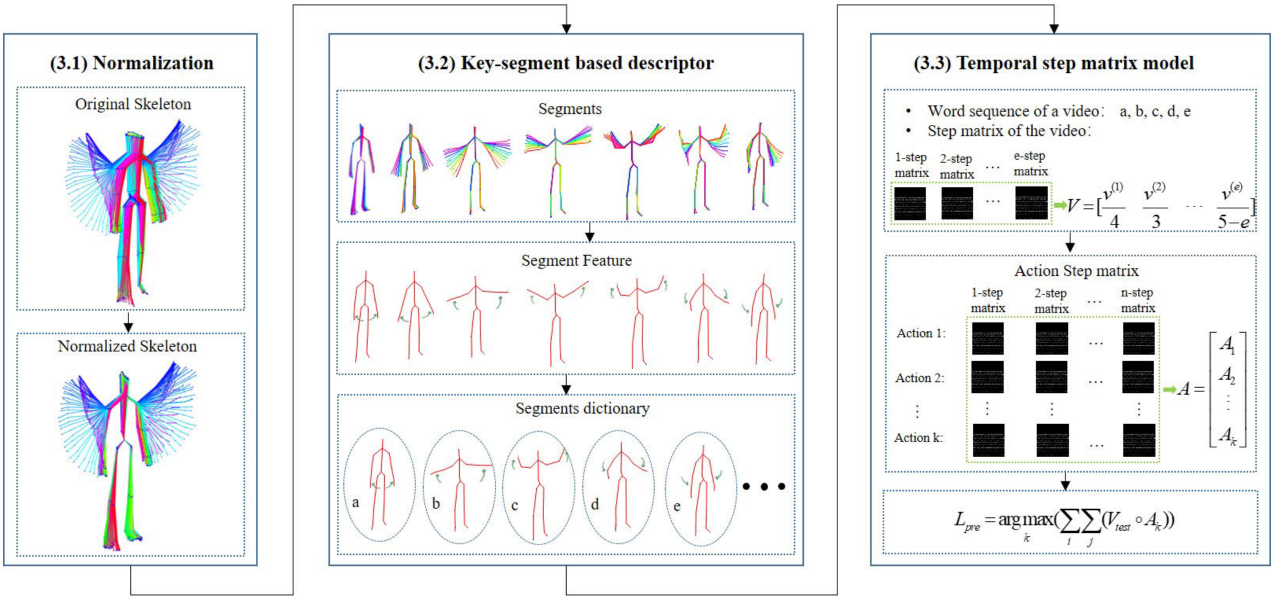

(3.1) Normalization
Original Skeleton
Normalized Skeleton
(3.2) Key-segment based descriptor
Segments
Segment Feature
Segments dictionary
a
b
c
d
e
(3.3) Temporal step matrix model
• Word sequence of a video: a, b, c, d, e
• Step matrix of the video:
1-step matrix
2-step matrix
...
e-step matrix
$V=[\frac{v^{(1)}}{4}\ \frac{v^{(2)}}{3}\ \cdots\ \frac{v^{(e)}}{5-e}]$
Action Step matrix
1-step matrix
2-step matrix
n-step matrix
Action 1:
Action 2:
Action k:
$A=\begin{bmatrix}A_1\\A_2\\\vdots\\A_k\end{bmatrix}$
$L_{pre}=\arg\max_k(\sum_i\sum_j(V_{test}\circ A_k))$

FIGURE 2. The work flow of the proposed action recognition method.

to represent an action as a sequence of key poses along the leaf node to the root node. The above methods have made some progress for human action recognition, but the recognition accuracy is still unsatisfactory.

Since recurrent Neural Networks (RNNs) encode temporal dynamic behavior, they are used to model the dynamic evolution of the skeleton sequence. Du *et al.* [9] proposed an end-to-end hierarchical RNN to present the relative motion between skeleton joints, in which the joints were divided into five parts and then fed into five independent subnets to extract local features. Long Short Term Memory network (LSTM), a special kind of RNNs, is capable of learning long-term dependencies, so it is often chosen to learn the complex motion dynamic of actions in many works. In [27], Liu et al. used LSTM to learn both spatial and temporal relationships among skeleton joints. Although deep learning-based methods [28] perform satisfactorily on action recognition, these methods have high computational cost and are easy to overfit.

III. PROPOSED METHOD

As illustrated in Fig.2, the proposed action recognition framework consists of three major parts: skeleton normalization, key-segment based descriptor, and temporal step matrix model. The original skeleton sequence is transformed into a new coordinate system firstly to eliminate the effect of various body sizes, body position, body orientations and different viewpoints. Then, the normalized skeleton sequence is automatically divided into several segments, which are treated as the action units, containing multiple skeleton frames with similar spatial information. Each skeleton sequence can be represented by an ordered words sequence according to the key segments dictionary formed by clustering the segments of all the training action samples. Finally, a temporal step matrix model is proposed to characterize the multiscale temporal information of the ordered word sequence. The step matrices of the training samples belonging to the same action class are added to get the action step matrix. The predicted class label of the test sample is the one whose action step matrix is most similar to the step matrix of the test sample.

A. SKELETON NORMALIZATION

A skeleton is composed of a plurality of joint point coordinates, and the joint point coordinates are related to the used reference coordinate systems, which are usually different in various real scenes. Even in the same shooting scene, the skeleton coordinates of humans having the same posture may be different due to the various body sizes, the difference in the angles of the sensor and the distances to the sensor. Therefore, the skeleton data needs to be normalized to eliminate the diversity of sizes and viewing angles.

Given a skeleton sequence with N frames as $J \in \mathbb{R}^{N\times K\times 3}$, the coordinate of the k^{th} joint in the n^{th} frame is $j_k^{(n)} = (x_k^{(n)}, y_k^{(n)}, z_k^{(n)})^{\mathrm{T}}$, where $k \in \{1, 2, \cdots, K\}$, $n \in \{1, 2, \cdots, N\}$, and K is the total number of the detected skeleton joints of a person. The value of K depends on the skeleton estimation method. Skeleton data are captured by Microsoft Kinect 1 and Microsoft Kinect 2 in most of the datasets. In this paper, the skeleton model and the joint configuration in the MSRC-12 dataset is used as shown in Fig.1 (a), where $K = 20$. The coordinates of the skeleton joints in the original reference coordinate system are sensitive to the view and scale, so the following two transformations are proposed to make the skeleton view-aligned and scale-normalized.

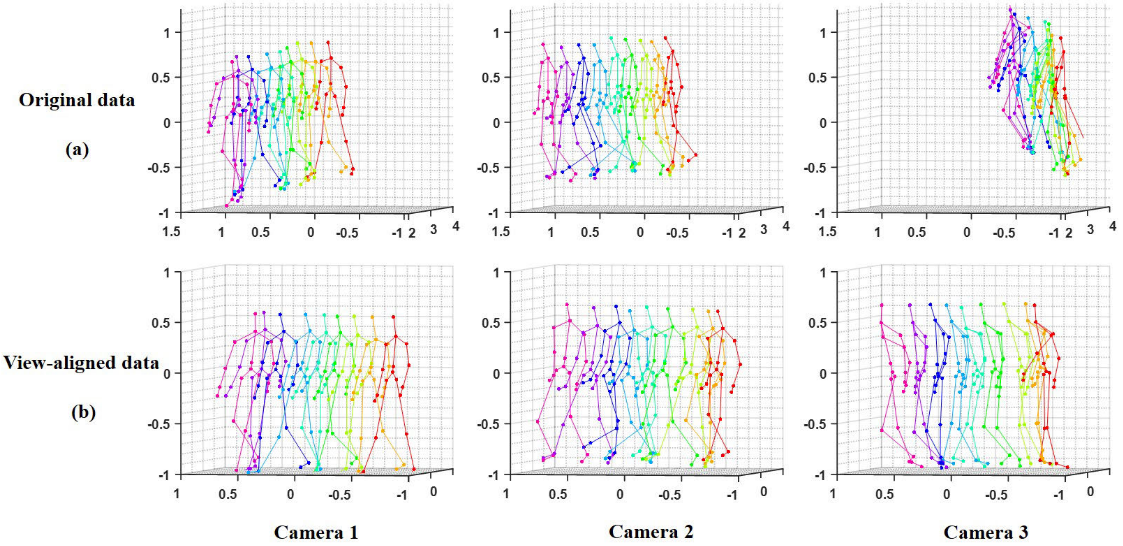


FIGURE 3. Result of view alignment. Skeleton frames are colored by inferring to the time label for facilitating observation. (a) Three skeleton sequences of the same action "walking" captured from three different viewpoints, which look quite different. (b) The corresponding view-aligned sequences of the original data, which look more similar and are easier to judge that they belong to the same action.

1) VIEW ALIGNMENT

In order to reduce the sensitivity of the skeleton sequence to the viewpoints, each skeleton frame is transformed into a new coordinate system using either a frame-based transformation [11], [17] or a sequence-based transformation [12]. The frame-based transformation transforms each frame individually in a skeleton sequence, which reduces the relative motion. While the sequence-based transformation preserves the original relative motion by performing the same translation to all the skeleton frames in the sequence. Here, the sequence-based transformation is employed.

The coordinates of all the skeleton frames in a sequence are transformed into a new coordinate system by the translation and rotation transformations:

$$\left[x', y', z', 1\right]^{\mathrm{T}} = \mathcal{T}(\boldsymbol{R}_x^{\boldsymbol{\alpha}}, \boldsymbol{R}_y^{\boldsymbol{\beta}}, \boldsymbol{R}_z^{\boldsymbol{\gamma}}, \boldsymbol{d})[x, y, z, 1]^{\mathrm{T}}, \qquad (1)$$

where, (x, y, z) is the original coordinate value, and (x', y', z') is the view-aligned coordinate value. $\mathcal{T}$ is the transformation matrix defined as:

$$\mathcal{T} = \begin{bmatrix} \boldsymbol{R}_x^{\alpha} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_y^{\beta} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_z^{\gamma} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I_3} & \boldsymbol{d} \\ 0 & 1 \end{bmatrix}, \qquad (2)$$

where, $\boldsymbol{I_3} \in \mathbb{R}^{3\times 3}$ is a unit matrix, and $\boldsymbol{d} \in \mathbb{R}^{1\times 3}$ is a translation vector. $\boldsymbol{R}_x^{\alpha}, \boldsymbol{R}_y^{\beta}, \boldsymbol{R}_z^{\gamma}$ are rotation matrices around X axis, Y axis and Z axis (right-handed coordinate system). The coordinate value of the X axis, Y axis, Z axis, and origin point of the new coordinate system in the original coordinate system is denoted as $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{o} \in \mathbb{R}^{1\times 3}$.As long as $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{o}$ are fixed, the transformation matrix $\mathcal{T}$ is determined as follows by transforming $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{o}$ to [1, 0, 0], [0, 1, 0], [0, 0, 1], [0, 0, 0] using formula 1:

$$\mathcal{T} = \begin{bmatrix} (\boldsymbol{\alpha}-\boldsymbol{o})^{\mathrm{T}} & (\boldsymbol{\beta}-\boldsymbol{o})^{\mathrm{T}} & (\boldsymbol{\gamma}-\boldsymbol{o})^{\mathrm{T}} & \boldsymbol{o}^{\mathrm{T}} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}. \qquad (3)$$

The average of the coordinates of the joint "hip center" of all skeleton frames in a skeleton sequence is used as the origin of the new coordinate system, meaning that the original origin is moved to the joint "hip center", so the value of $\boldsymbol{o}$ is computed as:

$$\boldsymbol{o} = -\frac{1}{N}\sum_{n=1}^{N} j_1^{(n)}. \qquad (4)$$

The Z axis of the new coordinates is set to be parallel to the first principal component of the torso matrix $M \in \mathbb{R}^{(N\times|\psi|)\times 3}$, which contains the joints coordinates in the torso set, and it is defined as:

$$M = \bigcup_{n\in[1,2,\cdots,N],k\in\psi} j_k^{(n)}, \qquad (5)$$

where, ψ denotes the index set of the joints in the "torso" part as shown in the Fig.1 (a). That is, $\psi = \{1, 2, 3, 4, 5, 9, 13, 17\}$. $\boldsymbol{\gamma}$ is not only parallel to the first principal component of the torso matrix, but also its direction is consistent with that from the hip center pointing head. Similar to Z axis, the X axis of the new coordinates in the original coordinate system is defined as the second principal component of the torso matrix and consistent with the direction from the left hip pointing right hip. The Y axis $\boldsymbol{\beta}$ of the new coordinates is defined as $\boldsymbol{\beta} = \boldsymbol{\alpha} \times \boldsymbol{\gamma}$.

The skeleton frames in a sequence are transformed into the new reference coordinate system according to formulas 1 and 3. The aligned coordinate $j'^{(n)}_k = (x'^{(n)}_k, y'^{(n)}_k, z'^{(n)}_k)$ is more robust to viewpoint changes. Fig. 3 (a) shows an

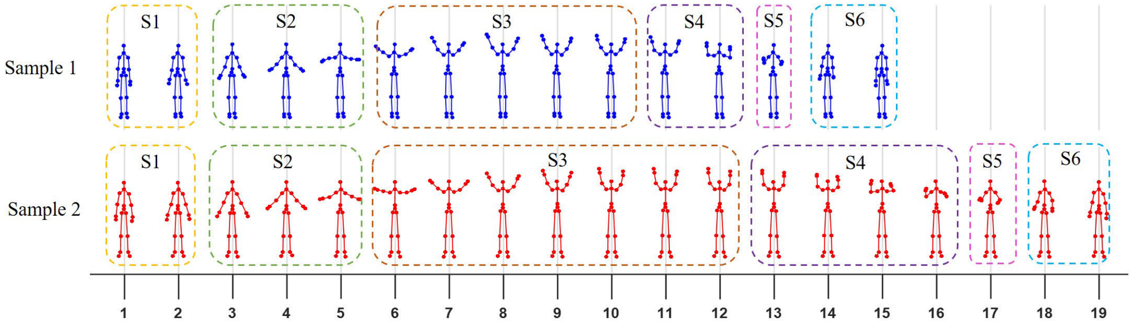


FIGURE 4. Two skeleton sequences in which two humans performed the same action named "raise outstretched arms" at different speeds. There are 15 frames in sample 1 to complete the action, while there are 19 frames in sample 2. However, both of these two samples have six skeleton segments.

action captured by cameras from three views, in which the sequences from each camera look quite different even they belong to the same action. After view alignment, the transformed skeleton sequences look more similar, and it is easier to judge that they belong to the same action category, as shown in Fig. 3 (b).

2) SCALE NORMALIZATION

The view-aligned coordinate $j'^{(n)}_k = (x'^{(n)}_k, y'^{(n)}_k, z'^{(n)}_k)$ needs to go through scale normalization and the new scale-normalized coordinate is denoted as $\hat{j}^{(n)}_k = (\hat{x}^{(n)}_k, \hat{y}^{(n)}_k, \hat{z}^{(n)}_k)$. The distance between the joint "hip center" and the joint "spine" is almost constant when human perform most actions, so the average distance between the joint "hip center" and the joint "spine" of all skeleton data in a sequence is used as a unit distance to make corresponding changes to the coordinates of other joints as follows:

$$\frac{j'^{(n)}_k - j'^{(n)}_1}{d_{1,2}} = \frac{\hat{j}^{(n)}_k - j'^{(n)}_1}{1}, \tag{6}$$

where, $d_{1,2}$ is the average distance between the joint "hip center" and the joint "spine" computed as:

$$d_{1,2} = \frac{1}{N}\sum_{n=1}^{N}\sqrt{(x'^{(n)}_1 - x'^{(n)}_2)^2 + (y'^{(n)}_1 - y'^{(n)}_2)^2 + (z'^{(n)}_1 - z'^{(n)}_2)^2}. \tag{7}$$

Therefore, the final view-aligned and scale-normalized skeleton coordinate $\hat{j}^{(n)}_k$ is computed as:

$$\hat{j}^{(n)}_k = \frac{1}{d_{1,2}} j'^{(n)}_k + (1 - \frac{1}{d_{1,2}}) j'^{(n)}_1. \tag{8}$$

B. KEY-SEGMENT BASED DESCRIPTOR

Humans usually have different speeds even when completing the same action, but their decomposition actions are almost same. For example, two persons perform the same action named "raise outstretched arms" under the same imaging capture device, and their skeleton sequences are shown in Fig.4. The total numbers of frames they used to complete the action are 15 and 19 respectively, but the numbers of the decomposition actions of them are same and equal 6 (*i.e.* S1-S6). What's more, the decomposition actions are semantically meaningful and can be interpreted and described. These six decomposition actions are unlifted arms (S1), raise arms below shoulders (S2), raise arms above shoulders (S3), put down arms above shoulders (S4), put down arms below shoulders (S5), and completely falling arms (S6). Here, the decomposition action is not a skeleton frame, but several skeleton frames with similar spatial information in a short time, namely, skeleton segment, which is treated as the action unit in this paper. Therefore, a skeleton sequence representation based on the atomic action units is proposed, which is named as the key-segment based descriptor. To form this descriptor, the skeleton sequence is firstly divided into multiple skeleton segments according to a designed segmentation scheme, and then the skeleton segment features are extracted and clustered to obtain a codebook. Each segment in the skeleton sequence can be represented by a word (the index of the cluster center) in the dictionary which is closest to it. Finally, a skeleton sequence is presented as an ordered word sequence.

1) SEGMENTATION SCHEME

A skeleton sequence is divided into a number of skeleton segments. Each segment consists of several consecutive and similar skeleton frames. A dynamic threshold $Th^{(n)}_s$ is proposed to measure the similarity between two frames and to partition the sequence into segments.The distance between the left and right hips in the current frame (that is, the distance between the joint 13 and joint 17) is taken as the standard value, and the dynamic threshold is defined as:

$$Th^{(n)}_s = p_s\sqrt{(\hat{x}^{(n)}_{13} - \hat{x}^{(n)}_{17})^2 + (\hat{y}^{(n)}_{13} - \hat{y}^{(n)}_{17})^2 + (\hat{z}^{(n)}_{13} - \hat{z}^{(n)}_{17})^2}, \tag{9}$$

where, p_s is a proportional parameter, whose value is discussed in section 4.4. Here, the skeleton joints are divided into five parts according to the physical structure and they are torso, left arm, right arm, left leg and right leg as shown in Fig.1 (a). Define the similarity S_{ij} of two skeleton frames

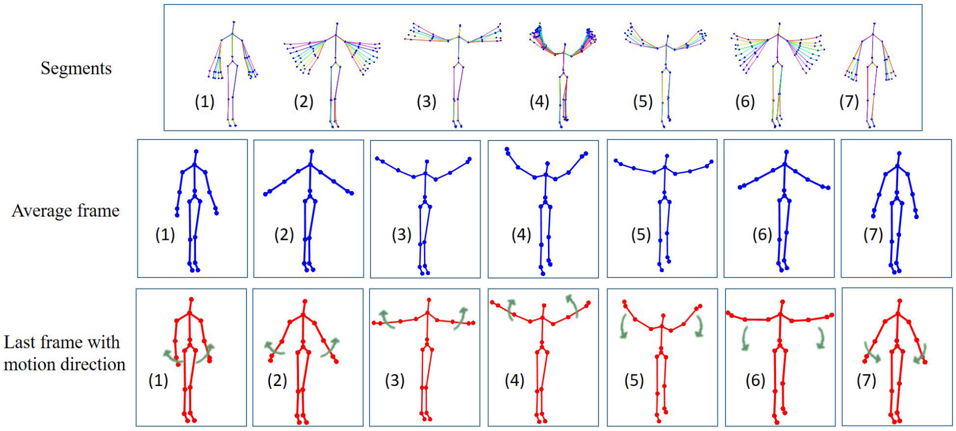


FIGURE 5. Selection of segment features. The images in the first row are ordered segments of the action "raise arms and lower arms". The blue skeletons in the second row are the mean skeletons of the corresponding skeleton segments and the red skeleton with green narrow in the last row is the extracted segment feature consisting of the last frame skeleton and the motion direction of the segment. Average frames of segment (2) and segment (6) are almost the same, but last frames with motion directions of them are easy to distinguish. The same is true for segments (1) and (7), (3) and (5).

i^{th} and j^{th} as follows,

$$S_{ij} = \sum_{t=1}^{5} (s_{ij}^{(t)}), \tag{10}$$

where, $s_{ij}^{(t)}$ is the maximum movement of the t^{th} part between i^{th} and j^{th} frames and it is calculated as:

$$s_{ij}^{(t)} = \max_{k\in\varphi_t} \sqrt{(\hat{x}_k^{(i)} - \hat{x}_k^{(j)})^2 + (\hat{y}_k^{(i)} - \hat{y}_k^{(j)})^2 + (\hat{z}_k^{(i)} - \hat{z}_k^{(j)})^2}, \tag{11}$$

where, $\varphi_t, t \in \{1, 2, \cdots, 5\}$ is the set of skeleton joints belongs to each parts as shown in Fig.1.(a).

2) FEATURE EXTRACTION

The features of a skeleton segment consist of the skeleton joint position (spatial feature) and the direction of motion (local temporal feature). The combination of the mean and standard deviation of the skeleton frame coordinates in a skeleton segment is the most intuitive way to represent the segment. However, the mean and standard deviation are similar for two motions with opposite directions. For example, as shown in Fig.5, there are two segments (Segment (3) and (5)), in which arms lift and fall to the highest position. The mean and standard deviation of these two segments are almost identical, making it impossible to distinguish between these two segments, but they can be easily distinguished by adding the motion direction information because one direction is upward and the other is downward. Since the average of multiple skeleton frames may no longer be a skeleton model, a fixed frame (the last frame used here) is used to represent the spatial information of the skeleton segment. Therefore, the feature vector $\boldsymbol{F}_i \in \mathbb{R}^{K\times3\times3}$ of a segment $\mathscr{S}_i = (\hat{j}^{(i)}, \hat{j}^{(i+1)}, \cdots, \hat{j}^{(i+|\mathscr{S}_i|)})$ is:

$$\boldsymbol{F}_i = [\boldsymbol{\Gamma}_i; \boldsymbol{\sigma}_i; \boldsymbol{\Omega}_i], \tag{12}$$

where, $\boldsymbol{\Gamma}_i = \hat{j}^{(i+|\mathscr{S}_i|)}$, $\boldsymbol{\sigma}_i \in \mathbb{R}^{K\times3}$ is the standard deviation of the skeleton frame coordinates in segment $\mathscr{S}_i$, $\boldsymbol{\Omega}_i = \hat{j}^{(i+|\mathscr{S}_i|)} - \hat{j}^{(i)}$ is the kinematic direction of this segment.

3) WORD SEQUENCE REPRESENTATION

The feature vectors of all segments from the training samples are clustered by applying the K-means algorithm. We assume that the number of clustered categories is B. The B cluster centers, namely, key segments, form a codebook. The value of B on the dataset is related to the action categories of this dataset, that is $B = p_B \times Num$, p_B is a proportional parameter and Num is the number of the action classes in the dataset. Given a skeleton sequence $J \in \mathbb{R}^{N\times K\times3}$, which has been segmented into segments $J = (\mathscr{S}_1, \mathscr{S}_2, \cdots, \mathscr{S}_l)$ with length l. Each segment $\mathscr{S}_i$ in the skeleton sequence can be found the cluster center closest to it in the dictionary. Thus each skeleton sequence is coded as a word sequence $W = (w_1^{(c_1)}, w_2^{(c_2)}, w_3^{(c_3)}, \cdots, w_l^{(c_l)})$, where, c_i is the index of the cluster center closest to the i^{th} segment.

C. TEMPORAL STEP MATRIX MODEL

Each skeleton sequence is represented as a word sequence $W = (w_1^{(c_1)}, w_2^{(c_2)}, w_3^{(c_3)}, \cdots, w_l^{(c_l)})$. Enlightened from the bag-of-words method, the action categories can be simply determined based on the frequency of the occurrence of the key skeleton segments. However, this frequency-based statistical method results in the loss of temporal information. Therefore, a temporal step matrix model is proposed to make up for this loss, because it can retain global temporal information of the word sequence.

Every segment word is regarded as a step and the step matrix records the order relationship of words in the words sequence. The e-step matrix $v^{(e)}$ is initialized to an all-zero matrix of size $B \times B$. then all the word pairs of e-step are

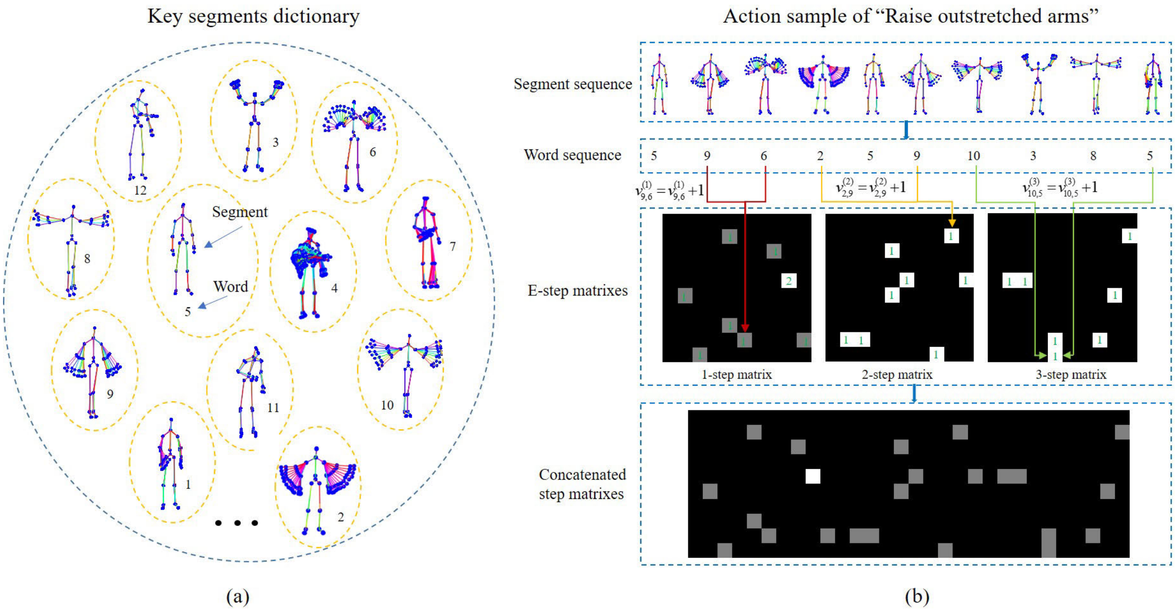

Key segments dictionary
12
3
6
8
Segment
Word
5
4
7
9
11
10
1
2
• • •
Action sample of "Raise outstretched arms"
Segment sequence
Word sequence
5 9 6 2 5 9 10 3 8 5
$v^{(1)}_{9,6}=v^{(1)}_{9,6}+1$
$v^{(2)}_{2,9}=v^{(2)}_{2,9}+1$
$v^{(3)}_{10,5}=v^{(3)}_{10,5}+1$
E-step matrixes
1-step matrix
2-step matrix
3-step matrix
Concatenated step matrixes
(a)
(b)

FIGURE 6. Generation of the step matrix. (a) The codebook formed by the cluster centers, namely, key segments dictionary. (b) An example of generating a step size matrix of an action "raise outstretched arms". The skeleton sequence is split into a sequence of segments and the word sequence is also obtained. Three $e-step$ matrices with different steps $e = 1, 2, 3$ are generated. Finally, these three step matrices are concatenated to form the final step matrix of the skeleton sequence.

added in the e-step matrix. That is,

$$v^{(e)}_{w_i,w_{i+e}} = v^{(e)}_{w_i,w_{i+e}} + 1, \quad (i = 1, 2, \cdots, l-e). \tag{13}$$

Different temporal information is recorded in the step matrix with different step e. Multiple step matrixes with different e are concatenated to form the final step matrix of the skeleton sequence $V = [\frac{v^{(1)}}{l-1}, \frac{v^{(2)}}{l-2}, \cdots, \frac{v^{(e)}}{l-e}]$. The step matrixes of the training videos belonging to the same action category are added to get the action step matrix $A_i = \frac{1}{\xi_i}\sum_{q\in\xi_i} V_q$, where, ξ_i is the set in which videos belonging to i_{th} action category. The predicted label for a test video which has the step matrix V_{test} is

$$L_{pre} = \arg\max_k(\sum_i\sum_j(V_{test} \circ A_k)). \tag{14}$$

For a better explanation, we take an action video of "raise outstretched arms" in the MSRC-12 dataset as an example to illustrate the generation of step matrix, as shown in Fig.6. The clustering centers of the segments from all training samples in the dataset form the key segments dictionary. The skeleton sequence is split into a sequence of segments and the word sequence is also obtained. Here, 1-step, 2-step, and 3-step matrix are generated, and these three step matrices with different steps are concatenated to form the final step matrix. In fact, the selection and the combination of step size e are based on the characteristics of different datasets, which is discussed in section 4.3.

Particularly, when the step matrix is a 0-step matrix, the step matrix model degenerates to the frequency-based statistical method, which is effective enough for some repetitive action datasets. This is used as a baseline method to evaluate the effectiveness of the proposed key-segments based action descriptor.

IV. EXPERIMENTS AND DISCUSSIONS

There are plenty of RGBD action recognition datsets [29] containing skeleton data, and three datasets are chose to evaluate the proposed method: Northwestern-UCLA [30] dataset, MSRC-12 [16] dataset, and CAD-60 dataset [31]. Northwestern-UCLA dataset has several viewpoints, and the actions in each action video are performed once. MSRC-12 dataset is a repetitive dataset, in which the same actions repeat several times in an action video. CAD-60 dataset is a single view and non-repeated dataset.

A. NORTHWESTERN-UCLA

The Northwestern-UCLA Multi-view 3D event dataset [30] (Northwestern-UCLA) was collected by three Kinect cameras, which contains 1494 sequences covering 10 action classes from 10 performers. And these 10 actions are: pick up with one hand, pick up with two hands, drop trash, walk around, sit down, stand up, donning, doffing, throw and carry. The subjects perform an action only one time in an action sequence, which contains an average of 39 frames. Samples in this dataset are captured from three viewpoints as shown in Fig.7. According to the cross-view action recognition protocol in this dataset [30], the samples from camera 1 and camera 2 constitute training data and the samples from camera 3 are set as testing data. The action recognition accuracies of the proposed method compared with the state-of-art algorithms are shown in Table 1.

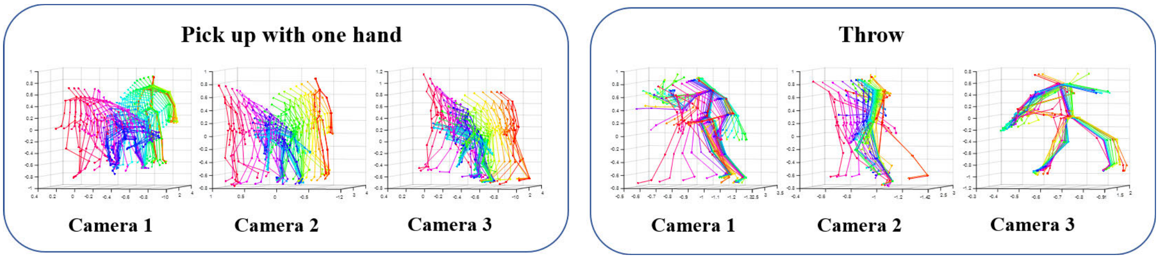


FIGURE 7. Samples in Northwestern-UCLA dataset [30] captured from three different viewpoints.

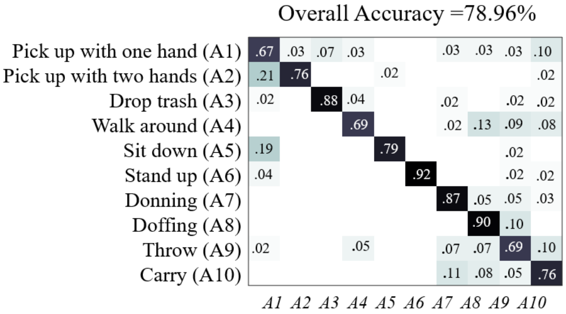
TABLE 1. Action recognition accuracies on the Northwestern-UCLA dataset [30].

Method	Feature	Accuracy(%)
Poselet [32] (2011)	Hand-crafted	24.50
Hankelet [33] (2012)	Hand-crafted	45.20
Virtual View [34] (2012)	Hand-crafted	47.80
HOJ3D [11] (2012)	Hand-crafted	54.50
AE [35] (2013)	Hand-crafted	69.9
MST-AOG [30] (2014)	Hand-crafted	73.30
LARP [36] (2014)	Hand-crafted	74.20
HBRNN-L [9] (2015)	RNN	78.52
Baseline method (ours)	Hand-crafted	**62.69**
Step Matrix method (ours)	Hand-crafted	**78.96**

The proposed step matrix method performs best in all the hand-crafted methods [11], [30], [32]–[36] and it is also comparable to the deep learning-based method [9]. Compared with the Poselet [32] method and the HOJ3D [11] method, even the baseline method is much better, which explains the superiority of the segment-based descriptor over the pose-based descriptors. In this dataset, the proposed step matrix method achieves 15.62% higher than the baseline method, which indicates that the step matrix introduces valid temporal information on the non-repeating dataset and the addition of temporal information brings a qualitative increase in the accuracy. The confusion matrix of the proposed step matrix method is shown in Fig.8. There is a large confusion between the action "pick up with one hand" and the action "pick up with two hands" due to the noisy arms skeleton data especially when the video captured on the side. Action "sit down" and action "pick up with one hand" have high confusion because these two actions contain similar appearances and motions.

B. MSRC-12 DATASET

The Microsoft Research Cambridge-12 Kinect gesture dataset [16] (MSRC-12) was built to evaluate the impact of learning gesture recognition system, but it is currently used as an action recognition dataset. In the dataset, the performer is provided with five different kinds of guides to perform the same class of action. For example, one of the guides is an ordered series of still images with arrows annotating as shown in the first row of Fig.1 (b). This dataset contains 594 sequences, which are collected from 30 people performing 12 categories of actions. The performer repeats the same action about ten times in an action video, and each video contains an average of 1214 frames. This is a single view dataset, *i.e.*, all of the action samples are captured from the same viewpoint, so the view alignment is not applied on this dataset. Following the cross-subject protocol in [37], action sequences performed by odd subjects are used for training and action sequences performed by even subjects are for testing. The performance of the proposed method compared with other state-of-art methods is listed in Table 2.


Overall Accuracy =78.96%

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Pick up with one hand (A1)	.67	.03	.07	.03			.03	.03	.03	.10
Pick up with two hands (A2)	.21	.76			.02					.02
Drop trash (A3)	.02		.88	.04			.02		.02	.02
Walk around (A4)				.69			.02	.13	.09	.08
Sit down (A5)	.19				.79				.02	
Stand up (A6)	.04					.92			.02	.02
Donning (A7)							.87	.05	.05	.03
Doffing (A8)								.90	.10	
Throw (A9)	.02			.05			.07	.07	.69	.10
Carry (A10)							.11	.08	.05	.76

FIGURE 8. Confusion matrix on the Northwestern-UCLA dataset [30].

TABLE 2. Action recognition accuracies on the MSRC-12 dataset [16].

Method	Feature	Accuracy(%)
HGM [38] (2014)	Hand-crafted	66.25
Pose-Lexicon [24] (2016)	Hand-crafted	85.86
ELC-KSVD [39] (2014)	Hand-crafted	90.22
Cov3DJ [40] (2013)	Hand-crafted	91.70
ConvNets [41] (2015)	CNN	84.46
SOS [42] (2016)	CNN	94.27
JTM [37] (2016)	CNN	94.86
Baseline method (ours)	Hand-crafted	**94.90**
Step Matrix method (ours)	Hand-crafted	**91.84**

On this dataset, the baseline method is much better than other hand-crafted based methods [24], [38]–[40], which illustrates the discriminative of the proposed segment-based action descriptor. The MSRC-12 dataset is a repeated dataset, in which the same action performs about ten times. In the step matrix, what we focus on is the context of the action units within an action, and these step pairs are recorded. However,

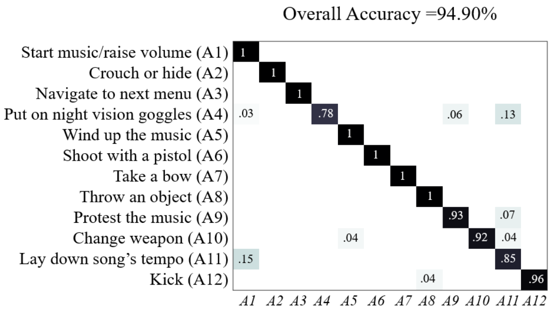


FIGURE 9. Confusion matrix on the MSRC-12 dataset [16].

on the repeated dataset, not only the step pairs within an action by also the step pairs between two consecutive actions are recorded into the matrix. The step pairs between two consecutive actions are meaningless and become noise in the step matrix, thus the performance of the step matrix method is decreased. What's more, the statistical characteristics of the action units on the repeated dataset are more obvious and easier to distinguish. Therefore, the result of the baseline method is better than the step matrix method on this dataset. Besides, the step matrix method is also in the higher band. The pose-lexicon method [24] is a key-pose based method, which is similar to our method. The pose-lexicon method constructed a pose dictionary, in which semantic poses and visual poses are in one-to-one correspondence, and then transformed the motion recognition into a problem of finding the maximum translation probability of a series of semantic poses given the visual poses. Our method is more accurate than this method by 9.04% because this key-pose based method lacks local temporal information. In addition, the textual instruction of actions and the description of extracted semantic poses need to be manually described in the pose-lexicon method, while our method is fully automated. What's more, our method is better than some CNN-based methods [37], [41], [42], which require long, computationally intensive training. The confusion matrix of our baseline method is shown in Fig.9. Action A1, A2, A3, A5, A6, A7, and A8 are distinguished by 100% and the recognition accuracies of action A9, A10, A12 are all above 92%.

C. CAD-60 DATASET

The Cornell Activity dataset [31] (CAD-60) was collected by Microsoft Kinect device. This public dataset contains 68 RGB video clips, depth sequences, and skeleton sequences. These actions are performed by four actors (two males and two females), covering 13 specific activities and one random activity. Each video lasts approximately 45 seconds and contains an average of 1181 frames. This dataset is a single view dataset and the view alignment is not applied to this dataset. Following the "new person" setting [35], the leave-one-out cross-validation is employed to test each person's data. The result of the proposed method compared with other state-of-art methods is reported in Table 3.

TABLE 3. Action recognition accuracies on the CAD-60 dataset [31].

Method	Feature	Accuracy(%)
MEMM [31] (2012)	Hand-crafted	51.90
STIP [43] (2014)	Hand-crafted	62.50
Object Affordance [44] (2013)	Hand-crafted	71.40
HON4D [45] (2013)	Hand-crafted	72.70
AE [35] (2013)	Hand-crafted	74.70
MPCCA [46] (2014)	Hand-crafted	79.10
JOULE [47] (2015)	Hand-crafted	84.10
Baseline method (ours)	Hand-crafted	**89.71**
Step Matrix method (ours)	Hand-crafted	**91.18**

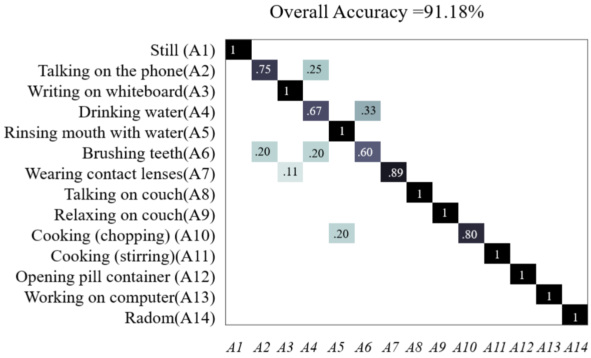


FIGURE 10. Confusion matrix on the CAD-60 dataset [31].

The proposed step matrix method and the baseline method achieve an accuracy of 91.18% and 89.71% respectively. Both results are better than these hand-crafted methods. The AE algorithm proposed an LOP feature to characterize the human motion, and a Fourier Temporal Pyramid to represent the temporal dynamics. Although this method and our method are both including the partition of the action into consecutive sub-actions, our method is better than this method by 15.01% higher in accuracy, illustrating that the proposed key-segment feature and the step matrix model are more discriminative to recognize actions with subtle differences. The confusion matrix of the proposed step matrix method on CAD-60 dataset under the "new person" setting is shown in Fig.10. The proposed method correctly classifies most of the actions. 33% of action A4 is classified into A6, and 25% of action A2 is classified into A4, because the movements of A2, A4, and A6 are tiny hand movements, and the minor differences among them are the hand position and the interactive object. Thus, it is difficult to distinguish them with the skeleton information.

D. SETTING OF STEP SIZE

The setting of step size depends on the characteristics of the dataset. Here, the step size setting is explained by taken two typical datasets (the repeated database MSRC-12 and the non-repeated dataset Northwestern-UCLA) as an example. In general, the ordering relationship between two words

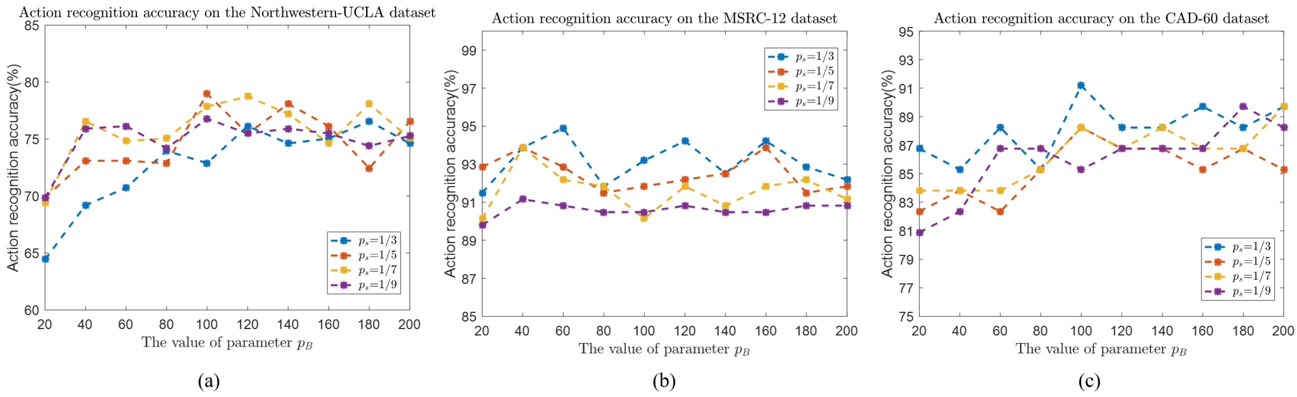


FIGURE 11. Action recognition accuracies with different parameter configurations. (a) Result on the Northwestern-UCLA dataset. (b) Result on the MSRC-12 dataset. (c) Result on the CAD-60 dataset.

TABLE 4. Action recognition accuracies on the Northwestern-UCLA dataset with different step settings.

Step setting	0-step	1-step	2-step	3-step	4-step
Accuracy(%)	62.69	71.15	62.21	64.21	55.10
Step setting	5-step	6-step	(1,3,5)-step	(1+2,3+4,5+6)-step	
Accuracy(%)	50.33	46.20	75.05	**78.96**	

TABLE 5. Action recognition accuracies on the MSRC-12 dataset with different step settings.

Step setting	0-step	1-step	2-step	3-step	4-step
Accuracy(%)	**94.90**	91.16	88.10	90.14	90.14
Step setting	5-step	6-step	(1,3,5)-step	(1+2,3+4,5+6)-step	
Accuracy(%)	90.14	90.48	91.16	91.84	

separated by six or more words is almost meaningless, so the step matrix with a step size of six or less is considered here. The action recognition accuracy on the Northwestern-UCLA dataset with different settings of step size is shown in Table 4. For such non-repeated datasets, the statistical characteristics ($0-step$ matrix) are not sufficient to represent the sequence characteristics. When the step size is greater than 1, the smaller the step size, the denser the temporal information is represented. That is the reason why the recognition accuracy decreases with the increase of the step size. However, the information represented by a single step matrix is always limited, and the combination of multiple step matrices with different sizes can represent multiple time scale information. The accuracy of the combination of 1-step, 3-step, and 5-step matrix $\big((1, 3, 5)-step\big)$ is greater than that of any other single step matrices. What's more, a soft temporal ordering scheme is added here, that is step matrices with adjacent step sizes are merged into a matrix, which is denoted as $(i, i+1)-step$ matrix. This makes the model robust to inaccurate temporal key segments localization and to partial orderings between action units. As reported in Table 4, the step size setting using the soft temporal ordering scheme $\big((1+2, 3+4, 5+6)-step\big)$ has the highest accuracy.

For repeated datasets, such as MSRC-12 dataset, statistical features ($0 - step$ matrix) are more representative than step matrix feature, because the addition of other step-length matrices brings noise and reduces the accuracy of behavior recognition. The action recognition accuracy on the MSRC-12 dataset with different settings of step size is reported in Table 5. The accuracy of $0-step$ matrix is higher than that of any other single step matrices, indicating the validity of statistical features on such datasets. Therefore, the step size for such repeated dataset is set to $0 - step$. In addition, the step size setting using the soft temporal ordering scheme $\big((1+2, 3+4, 5+6)-step\big)$ is higher than that of the single step settings that make up it.

E. EVALUATION OF PARAMETERS

There are two proportional parameters in the proposed method: the parameter p_B in the cluster centers and the parameter p_s in the segmentation scheme (formula 9). The value of p_B, which is related with the cluster centers, has a significant impact on the formed key segments dictionary. If the number of cluster centers is small, the key segments obtained by clustering are not sufficient to represent the information of all the segments. Conversely, there is redundancy in the key segments dictionary. The value of the parameter p_s directly affects the quality of the segmentation, which can be regarded as a temporal scale parameter. The ideal segmentation result should be: the movement of in one segment is less, and the motion of the adjacent segment is larger. The effect of different parameter configurations on action recognition accuracy is shown in Fig.11. On the Northwestern-UCLA dataset, when the value of the parameter p_s is equal to 1/5 or 1/7, the accuracy is high. Considering the computational complexity, the parameter p_B is set to $p_B = 100$. At this time, when $p_s = 1/5$, the accuracy reaches the maximum. On the MSRC-12 dataset, the larger the value of the parameter p_s, the higher the overall recognition accuracy. For another parameter p_B, when its value equals 60, 120 or 160, the accuracy is relatively high. However, the larger the number of cluster centers, the higher the computational complexity.

TABLE 6. Computation time of the proposed method on three datasets.

Dataset	Training (s)	Test (s)	Total time (s)
Northwestern-UCLA	138.30	146.34	284.64
MSRC-12	428.56	555.93	984.49
CAD-60	59.37	129.60	188.97

Therefore, in the case where the accuracy is not much different, the parameter p_B should be as small as possible. The parameters are set to $p_B = 60$, $p_s = 1/3$ on this dataset. On the CAD-60 dataset, the recognition accuracy achieves highest when $p_B = 100$, $p_s = 1/3$.

F. EVALUATION OF COMPUTATION TIME

All the experiments are conducted on an Intel(R) Core(TM) i7-4790 CPU at 3.60 GHz with 16GB RAM, using Matlab R2016a. The computation time of the proposed method on each dataset is reported in Table 6. On the Northwestern-UCLA dataset, the computation time of the proposed method is 284.64s, which includes 138.30s for training, 146.34s for testing. It takes an average of 0.32s to test a sample. For the MSRC-12 dataset, the total running time is 984.49s, consisting of 428.56s for training and 555.93s for testing. It takes an average of 1.89s to test a sample. On the CAD-60 dataset, the computation time is 188.97s, including 59.37s for training and 129.60s for testing. Testing a sample takes an average of 1.91s on this dataset. The training time is related to the average length of the video and the total number of videos in the training set, so the training time on the MSRC-12 dataset is the longest. The test time is primarily determined by the average length of the video, so the test time for one sample on the MSRC-12 dataset and the CAD-60 dataset is longer than the time on the Northwestern-UCLA dataset.

V. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a skeleton-based action recognition algorithm with key-segment descriptor and temporal step matrix model. The skeleton sequence is automatically divided into skeleton segments according to the segmentation scheme. The skeleton segments consist of skeleton frames with little change in motion in a short period of time, which is treated as the action unit. Then the skeleton sequence can be presented as a sequence of key segments, which are the cluster centers of all the segments from the training sample. The key-segment descriptor is proved to be more efficient than the key-pose descriptors because it can represent not only the spatial information of the skeleton, but also the movement information in a short time. Inspired by the bag-of-feature method, the statistical-based feature of the segments is enough for action recognition on the repeated dataset. For the non-repeated dataset, the step matrix is proposed to code the temporal information of the segment sequence. The experimental result shows that the step matrix method introduces valid temporal information on the non-repeated dataset and the addition of temporal information brings a qualitative increase in the accuracy. Experiments on three challenging datasets demonstrate that the proposed method outperforms all the hand-craftd methods, and it also performs better than some classical deep learning-based methods. What's more, the proposed step matrix model can be used in the other sequential data recognition. In the future work, we hope to improve the existing method from two aspects: the design of the segmentation scheme and the extraction of segment features.

REFERENCES

[1] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, May 2013.

[2] 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.

[3] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[4] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4305–4314.

[5] Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal pyramid network for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1529–1538.

[6] M. A. Uddin and Y.-K. Lee, "Feature fusion of deep spatial features and handcrafted spatiotemporal features for human action recognition," *Sensors*, vol. 19, no. 7, p. 1599, Apr. 2019.

[7] 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. 3288–3297.

[8] P. Koniusz, A. Cherian, and F. Porikli, "Tensor representations via kernel linearization for action recognition from 3D skeletons," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2016, pp. 37–53.

[9] 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.

[10] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1290–1297.

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

[12] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, Aug. 2017.

[13] H. H. Pham, H. Salmane, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Spatio–temporal image representation of 3D skeletal movements for view-invariant action recognition with deep convolutional neural networks," *Sensors*, vol. 19, no. 8, p. 1932, Apr. 2019.

[14] L. Miranda, T. Vieira, and D. Martínez, T. Lewiner, A. W. Vieira, and M. F. Campos, "Online gesture recognition from pose kernel learning and decision forests," *Pattern Recognit. Lett.*, vol. 39, pp. 65–73, Apr. 2014.

[15] Y. Hou, S. Wang, P. Wang, Z. Gao, and W. Li, "Spatially and temporally structured global to local aggregation of dynamic depth information for action recognition," *IEEE Access*, vol. 6, pp. 2206–2219, 2017.

[16] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, May 2012, pp. 1737–1746.

[17] M. Jiang, J. Kong, G. Bebis, and H. Huo, "Informative joints based human action recognition using skeleton contexts," *Signal Process., Image Commun.*, vol. 33, pp. 29–40, Apr. 2015.

[18] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, Aug. 2011, pp. 147–156.

[19] X. Chen and M. Koskela, "Skeleton-based action recognition with extreme learning machines," *Neurocomputing*, vol. 149, pp. 387–396, Feb. 2015.

[20] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1012–1020.

[21] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 24–38, 2014.

[22] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou, "Ongoing human action recognition with motion capture," *Pattern Recognit.*, vol. 47, no. 1, pp. 238–247, 2014.

[23] 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.

[24] L. Zhou, W. Li, and P. Ogunbona, "Learning a pose lexicon for semantic action recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.

[25] 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.

[26] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3D discriminative skeletal features for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 471–478.

[27] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2016, pp. 816–833.

[28] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "RGB-D-based human motion recognition with deep learning: A survey," *Comput. Vis. Image Understand.*, vol. 171, pp. 118–139, May 2018.

[29] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "RGB-D-based action recognition datasets: A survey," *Pattern Recognit.*, vol. 60, pp. 86–105, Dec. 2016.

[30] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2649–2656.

[31] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 842–849.

[32] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *Proc. CVPR*, Jun. 2011, pp. 3177–3184.

[33] B. Li, O. I. Camps, and M. Sznaier, "Cross-view activity recognition using Hankelets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1362–1369.

[34] R. Li and T. Zickler, "Discriminative virtual views for cross-view action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2855–2862.

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

[36] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[37] P. Wang, Z. Li, Y. Hou, and W. Li, "Action recognition based on joint trajectory maps using convolutional neural networks," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 102–106.

[38] S. Yang, C. Yuan, W. Hu, and X. Ding, "A hierarchical model based on latent Dirichlet allocation for action recognition," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 2613–2618.

[39] L. Zhou, W. Li, Y. Zhang, P. Ogunbona, D. T. Nguyen, and H. Zhang, "Discriminative key pose extraction using extended LC-KSVD for action recognition," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2014, pp. 1–8.

[40] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Jun. 2013, pp. 2466–2472.

[41] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 579–583.

[42] 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.

[43] Y. Zhu, W. Chen, and G. Guo, "Evaluating spatiotemporal interest point features for depth-based action recognition," *Image Vis. Comput.*, vol. 32, no. 8, pp. 453–464, Aug. 2014.

[44] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *Int. J. Robot. Res.*, vol. 32, no. 8, pp. 951–970, 2013.

[45] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 716–723.

[46] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 596–603.

[47] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5344–5352.

RUIMIN LI received the B.S. degree in information and computing science from Shanxi University, Taiyuan, China, in 2015. She is currently pursuing the Ph.D. degree in signal and information processing with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences. She is begin cultivated with the Xi'an Institute of Optics and Precision Mechanics, CAS, Xi'an, China. Since 2017, she has been a Research Assistant with the Department of Computer Science, Chu Hai College of Higher Education, Hong Kong. Her research interests include human action recognition, human action evaluation, visual-motor coordination analysis, intelligent diagnostic systems development, digital circuit design, and simulation and compressed coded hyperspectral imaging.

HONG FU received the B.S. and M.S. degrees from Xi'an Jiaotong University, in 2000 and 2003, respectively, and the Ph.D. degree from Hong Kong Polytechnic University, in 2007.

She was a Postdoctoral Fellow with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University. She is currently a Professor and the Associate Head of the Department of Computer Science, Chu Hai College of Higher Education, Hong Kong. She has published more than 70 technical articles. She is also the PI for four external funded research projects supported by the Research Grants Council of the Hong Kong Special Administrative Region, China. Her research interests include eye-tracking for strabismus diagnosis and training, attention-driven image understanding, action recognition, action evaluation, and intelligent diagnostic systems development. She was a recipient of the prize of Excellent Paper on the 2016 9th International Conference on Advanced Computer Theory and Engineering (ICACTE 2016), and the Best Short Paper on Eye Tracking Research and Applications 2010 (ETRA2010).

WAI-LUN LO received the B.Eng. degree in electrical engineering and the Ph.D. degree from the Hong Kong Polytechnic University, Hong Kong, in 1991 and 1996, respectively.

He was a Research Assistant and then a Research Associate with the Department of Electrical Engineering, Hong Kong Polytechnic University, from 1996 to 1997. From 1997 to 1999, he was a Postdoctoral Fellow with the Department of Electrical Engineering, Hong Kong Polytechnic University. In 1999, he was with the Department of Electronic Engineering, City University of Hong Kong, as a Research Fellow, before joining the Department of Computer Science, Chu Hai College of Higher Education, Hong Kong, in September 1999. He is currently a Professor and the Head of the Department of Computer Science, Chu Hai College of Higher Education. His research interest includes adaptive control, fuzzy control, and application of intelligent control in power electronics circuits.

ZHERU CHI received the B.Eng. and M.Eng. degrees from Zhejiang University, in 1982 and 1985, respectively, and the Ph.D. degree from the University of Sydney, in March 1994, all in electrical engineering. Form 1985 to 1989, he was on the faculty of the Department of Scientific Instruments, Zhejiang University. He worked as a Senior Research Assistant/Research Fellow with the Laboratory for Imaging Science and Engineering, University of Sydney, from April 1993 to January 1995. Since February 1995, he has been with The Hong Kong Polytechnic University, where he is currently an Associate Professor with the Department of Electronic and Information Engineering. Since 1997, he has also been served on the organization or program committees for a number of international conferences. He has authored or coauthored one book and 12 book chapters, and published more than 230 technical articles. His research interests include machine learning, pattern recognition, and computational intelligence. He was an Associate Editor of the IEEE Transactions on Fuzzy Systems, from 2008 to 2010.

ZONGXI SONG received the B.S. and M.S. degrees from Xi'an Jiaotong University, and the Ph.D. degree from the Xi'an Institute of Optics and Precision Mechanics, CAS, Xi'an, China.

He is currently a Research Fellow with the Xi'an Institute of Optics and Precision Mechanics, CAS. He obtained two invention patents. He is also mainly involved in low-noise electronics design, image and video processing, high-speed photoelectric information acquisition and processing, space machine vision, and other research work. He was a recipient of the second prize of the Shaanxi Science and Technology Progress Award.

DESHENG WEN received the B.S. and M.S. degrees. He is currently a Research Fellow with the Xi'an Institute of Optics and Precision Mechanics, CAS, Xi'an, China. He is also mainly involved in space optical load technology, photoelectric imaging technology, electronic technology, fast signal processing technology, and other research work.