Pattern Recognition 68 (2017) 346–362

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Enhanced skeleton visualization for view invariant human action recognition

Mengyuan Liu[a], Hong Liu[a,*], Chen Chen[b]

[a] Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China
[b] Center for Research in Computer Vision, University of Central Florida, Orlando, FL 32816, USA

ARTICLE INFO

Article history:
Received 14 December 2016
Revised 18 February 2017
Accepted 25 February 2017
Available online 3 March 2017

Keywords:
Human action recognition
View invariant
Skeleton sequence

ABSTRACT

Human action recognition based on skeletons has wide applications in human–computer interaction and intelligent surveillance. However, view variations and noisy data bring challenges to this task. What's more, it remains a problem to effectively represent spatio-temporal skeleton sequences. To solve these problems in one goal, this work presents an enhanced skeleton visualization method for view invariant human action recognition. Our method consists of three stages. First, a sequence-based view invariant transform is developed to eliminate the effect of view variations on spatio-temporal locations of skeleton joints. Second, the transformed skeletons are visualized as a series of color images, which implicitly encode the spatio-temporal information of skeleton joints. Furthermore, visual and motion enhancement methods are applied on color images to enhance their local patterns. Third, a convolutional neural networks-based model is adopted to extract robust and discriminative features from color images. The final action class scores are generated by decision level fusion of deep features. Extensive experiments on four challenging datasets consistently demonstrate the superiority of our method.

1. Introduction

Human action recognition has been applied in various fields, e.g., human–computer interaction, game control and intelligent surveillance. Earlier works [1–5] recognize actions from RGB data, which involves complex illumination conditions and cluttered backgrounds. With rapid advances of imaging technology in capturing depth information in real-time, there has been a growing interest in solving these problems by using depth data generated from depth sensors [6–11], particularly the cost-effective Microsoft Kinect sensor [12].

Compared with RGB data, depth data generated by structured light sensors is more robust to changes in lighting conditions because depth values are estimated by infrared radiation without relating it to visible light. Subtracting foreground from cluttered background is easier using depth, as the confusing texture and color information from cluttered backgrounds are ignored. In addition, RGB-D cameras (e.g., Kinect) provide depth maps with appropriate resolution and accuracy, which provide three-dimensional information on the structure of subjects/objects in the scene.

Intuitively speaking, human body can be represented as an articulated system with hinged joints and rigid bones, and human actions can be denoted as movements of skeletons. With the implementation of capturing skeletons from Kinect in real-time [13], many works [14–16] have been conducted on skeleton-based action analysis. These works are usually designed for action analysis from a single view. However, a generic and reliable action recognition system for practical applications needs to be robust to different viewpoints while observing an action sequence. Therefore, this paper develops a view-independent action recognition method, which intends to eliminate the effect of viewpoint variations and proposes a compact yet discriminative skeleton sequence representation.

First, a sequence-based transform is applied on a skeleton sequence to make the transformed sequence invariant to the absolute body position and the initial body orientation. Since the depth sensor is usually fixed, one transform matrix is able to identify the orientation of the depth sensor. Intuitively, any skeleton from the sequence can generate the transform matrix. To eliminate the effect of noise on skeletons, we jointly use all torso joints from the sequence to formulate the transform matrix. In previous works [17–19], each skeleton is transformed by a transform matrix which is estimated from itself. These methods suffer from noisy skeletons, since each skeleton only contains limited number of skeleton joints, which are usually noisy. What's worse, the origi-

* Corresponding author.
E-mail addresses: liumengyuan@pku.edu.cn (M. Liu), hongliu@pku.edu.cn (H. Liu), chenchen870713@gmail.com (C. Chen).

http://dx.doi.org/10.1016/j.patcog.2017.02.030
0031-3203/

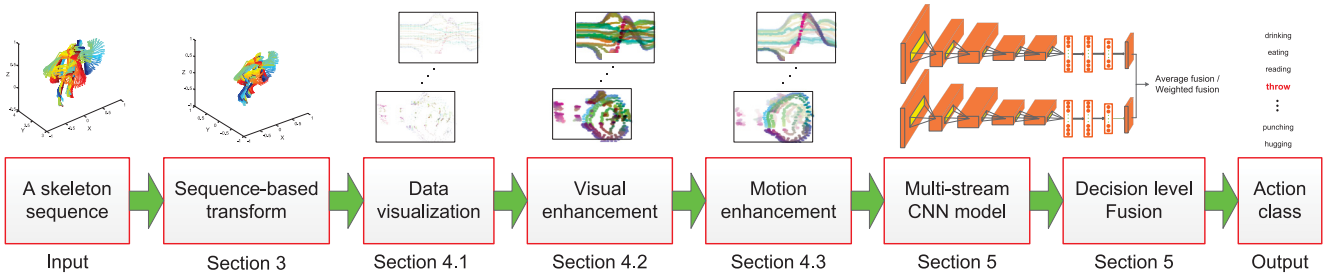

Fig. 1. Pipeline of our method.

nal spatio-temporal relationships among skeletons may be harmed by transforming skeletons with different transform matrices. Compared with these methods, our sequence-based transform is more robust to noise, since all torso joints from a sequence are used to estimate one transform matrix. Our method is also able to preserve relative spatio-temporal relationships among skeletons, since all skeletons are synchronously transformed by one transform matrix.

Second, the transformed sequence is visualized as a series of color images, which encode both spatial and temporal distributions of skeleton joints in a compact and descriptive manner. Specifically, skeleton joints are treated as points in a five dimensional (5D) space, including three dimensions of coordinates, one dimension of time label and one dimension of joint label. Then, two elements from the space are selected to construct a two dimensional (2D) coordinate space, and other three elements from the space is used to build a three dimensional (3D) color space. These two spaces are jointly used to generate color images, where each color pixel denote a point from the 5D space. To enhance the local patterns of color images, we apply the mathematical morphology method to highlight the colored pixels. To make color images more sensitive to motions, we develop a weighting scheme to emphasize skeleton joints with salient motions.

The proposed pipeline in Fig. 1 is most related to previous works [20,21], where skeleton sequences are described as color images which served to CNNs model for classification. In these methods, color images implicitly involve local coordinates, joint labels and time labels of skeleton joints. However, they either overemphasize the spatial or the temporal distribution of skeleton joints. Compared with these methods, our method captures more abundant spatio-temporal cues, since the generated color images extensively encode both spatial and temporal cues. Moreover, our method involves a new transform approach to eliminate the problem of viewpoint changes, which is ignored by [20,21]. Additionally, the weighted fusion method in our multi-stream CNNs model also performers better than the traditional average fusion method used in [20].

Generally, our method contains three main contributions:

- A sequence-based view invariant transform is developed to effectively cope with view variations. This method eliminates the effect of view variations, meanwhile preserves more relative motions among original skeleton joints than traditional skeleton-based transform methods, e.g., [17–19].
- An enhanced skeleton visualization method is proposed to represent a skeleton sequence as a series of visual and motion enhanced color images, which implicitly describe spatio-temporal skeleton joints in a compact yet distinctive manner. This method outperforms related works [20,21] by capturing more abundant spatio-temporal information of skeleton joints.
- A multi-stream CNN fusion model is designed to extract and fuse deep features from enhanced color images. Our proposed method consistently achieves the highest accuracies on four benchmark datasets as compared to the state-of-the-art skeleton-based action recognition methods.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 presents the sequence-based transform method. Section 4 provides the enhanced skeleton visualization method. Section 5 describes the structure of multi-stream CNN model. Section 6 reports the experimental results and discussions. Section 7 concludes the paper.

2. Related work

View invariant action recognition using skeletons is challenging for two main reasons. First, appearances of skeletons under different views change dramatically, leading to inter-varieties among same types of actions. Second, it remains unsolved to effectively represent spatio-temporal data [22], including the skeletons. This section reviews related works aiming at solving above challenges.

2.1. View invariant transform

As for RGB or depth data, previous works [23,24] extract self-similarity matrix (SSM) feature, which refers to the similarity between all pairs of frames. Despite that SSM shows high stability under view variations, this temporal self-similarity descriptor is not strictly view invariant. With the exact locations of skeleton joints, one can directly use estimated transform matrix to make skeletons strictly view invariant. Xia et al. [17] aligned spherical coordinates with the person's specific direction, where the hip center joint is defined as the origin, the horizontal reference vector is defined as the direction from the left hip center to the right hip center projected on the horizontal plane, and the zenith reference vector is selected as the direction that is perpendicular to the ground plane and passes through the coordinate center. Following [17], Jiang et al. [18] also translated skeletons to a new coordinate system which is invariant to the absolute body position and orientation. In [17,18], the original skeletons are assumed to be perpendicular to the ground plane. Without this assumption, Raptis et al. [19] provided a more flexible view invariant transform method, where principal components are calculated for the torso points and the zenith reference vector is selected as the first principal component which is always aligned with the longer dimension of the torso. Generally speaking, these methods establish a specific coordinate system for each skeleton. However, this scheme is sensitive to noisy skeletons and may lead to the loss of original spatio-temporal relationships among different skeletons.

2.2. Spatio-temporal data representation

Hand-crafted methods: traditional methods design hand-crafted features to represent spatio-temporal skeleton joints and use time series models to model the global temporal evolution. Xia et al.

[17] modeled the spatial-temporal skeleton joints as a time series of visual words. Skeleton joints on each frame was represented by histograms of 3D joint locations (HOJ3D) within a modified spherical coordinate system. These HOJ3D were clustered into visual words, whose temporal evolutions were modeled by discrete hidden Markov models (HMMs). Ofli et al. [25] represented spatio-temporal skeleton joints as sequence of the most informative joints (SMIJ). At each time instant, the most informative skeletal joints which show highly relation to the current action are selected to denote the current skeleton. The dynamic motion cues among skeleton joints are modeled by linear dynamical system parameters (LDSP). Yang et al. [26] used joint differences to combine static postures and overall dynamics of joints. To reduce redundancy and noise, they obtained EigenJoints representation by applying Principal Component Analysis (PCA) to the joint differences. Beyond using joint locations or the joint angles to represent a human skeleton, Vemulapalli et al. [27] modeled the 3D geometric relationships between various skeleton joints using rotations and translations in 3D space. However, hand-crafted features can barely effectively model complex spatio-temporal distributions, since these features are usually shallow and dataset-dependent.

RNN-based methods: Recurrent Neural Networks (RNN) models and Long-Short Term Memory (LSTM) neurons have been used to model temporal evolutions of skeleton sequences. Du et al. [28] proposed an end-to-end hierarchical RNN to encode the relative motion between skeleton joints. In terms of body structure, the skeleton joints are divided into five main parts, which are fed into five independent subnets to extract local features. Since LSTM is able to learn representations from long input sequences using special gating schemes, many works chose LSTM to learn complex dynamics of actions. By collecting a large scale dataset, Shahroudy et al. [29] showed that LSTM outperforms RNN and some hand-crafted features. To learn the common temporal patterns of partial joints independently, they proposed a part-aware LSTM which has part-based memory sub-cells and a new gating mechanism. To extract the derivatives of internal state (DoS), Veeriah et al. [30] proposed a differential RNN by adding a new gating mechanism to the original LSTM. Zhu et al. [31] used LSTM to automatically learn the co-occurrence features of skeleton joints. Observing that previous RNN-based methods only model the contextual dependency in the temporal domain, Liu et al. [32] introduced a spatial-temporal LSTM to jointly learn both spatial and temporal relationships among joints. However, RNN-based methods trend to overstress the temporal information [20].

CNN-based methods: CNN models have achieved promising performance in image recognition. Many methods have been developed to encode video sequences as images, which are further explored by CNN. Simonyan et al. [33] proposed a two-stream CNN architecture incorporating spatial and temporal networks. They utilized each frame as input for the spatial network and accumulated inter-frame optical flows as inputs for the temporal network. Bilen et al. [22] proposed a dynamic image representation, which is a single RGB image generated by applying approximate rank pooling operator on raw image pixels of a sequence. Wang et al. [34] accumulated motions between projected depth maps as depth motion maps (DMM), which are served as inputs for CNN. Generally speaking, these methods apply operators, e.g., subtraction, rank pooling and accumulation, on raw pixels of a sequence to convert a sequence to an image. Despite the efficiency, these operators roughly compress original data, leading to the loss of distinct spatio-temporal information.

Aiming at encoding more spatio-temporal information, Wang et al. [20] projected local coordinates of skeleton joints on three orthogonal planes. On each plane, 2D trajectories of joints formed a color image, where time labels and joint labels are mapped to colors. The generated image directly reflects the local coordinates of joints and implicitly involves the temporal evolutions and joint labels. Du et al. [21] concatenated skeleton joints in each frame according to their physical connections, and used three components (x, y, z) of each joint as the corresponding three components (R, G, B) of each pixel. The generated image directly reflects the temporal evolutions and joint labels and implicitly involves the local coordinates of skeleton joints.

Generally speaking, CNN can automatically explore distinctive local patterns of images, therefore it is an effective way to encode a spatio-temporal sequence as images. However, images generated by previous works can barely capture sufficient spatio-temporal information. Thus, a more descriptive way is needed to encode sequences as images.

3. Sequence-based view invariant transform

To make skeleton sequence invariant to viewpoints, traditional work [19] developed a skeleton-based transform method, which transforms each skeleton to a standard pose. However, this method removes partial relative motions among the original skeletons. Taking an action "rotating the waist" as an example, rotations of the waist will be removed by the skeleton-based transform in [19], since each skeleton is transformed to be a standard pose, e.g., facing the front. To this end, we propose a sequence-based transform method, which synchronously transforms all skeletons, therefore retaining relative motions among skeletons.

Specifically, given a skeleton sequence $\mathcal{I}$ with F frames, the nth skeleton joint on the fth frame is formulated as $p_n^f = (x_n^f, y_n^f, z_n^f)^{\mathrm{T}}$, where $f \in (1, \ldots, F)$, $n \in (1, \ldots, N)$, N denotes the total number of skeleton joints in each skeleton. The value of N is determined by some skeleton estimation algorithms. Fig. 2 shows two commonly used joint configurations. In this section, we use the joint configuration in the NTU RGB+D dataset [29], where N equals to 25. Each joint p_n^f contains five components, i.e., three local coordinates x, y, z, time label f and joint label n. Therefore, p_n^f can be mapped to a point in a 5D space Ω_0:

$$[x, y, z, f, n]^{\mathrm{T}} \in \Omega_0. \tag{1}$$

Since three local coordinates x, y, z are sensitive to view variations, we transform them to view invariant values $\hat{x}, \hat{y}, \hat{z}$ by:

$$[\hat{x}, \hat{y}, \hat{z}, 1]^{\mathrm{T}} = \mathcal{P}\left(\mathbf{R}_x^{\alpha}, \mathbf{0}\right)\mathcal{P}\left(\mathbf{R}_y^{\beta}, \mathbf{0}\right)\mathcal{P}\left(\mathbf{R}_z^{\gamma}, \mathbf{d}\right)[x, y, z, 1]^{\mathrm{T}}, \tag{2}$$

where the transform matrix $\mathcal{P}$ is defined as:

$$\mathcal{P}(\mathbf{R}, \mathbf{d}) = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}_{4\times 4}, \tag{3}$$

where $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is a rotation matrix, $\mathbf{d} \in \mathbb{R}^3$ is a translation vector given as:

$$\mathbf{d} = -\frac{1}{F}\sum_{f=1}^{F} p_1^f, \tag{4}$$

which moves the original origin to the "hip center". Let $\mathbf{R}_z^{\theta}$ denote rotating the original coordinate around Z axis by θ degree, which is formulated as:

$$\mathbf{R}_z^{\theta} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

Similarly, rotation matrix $\mathbf{R}_x^{\theta}$ and $\mathbf{R}_y^{\theta}$ are defined as:

$$\mathbf{R}_x^{\theta} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix}, \quad \mathbf{R}_y^{\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}. \tag{6}$$

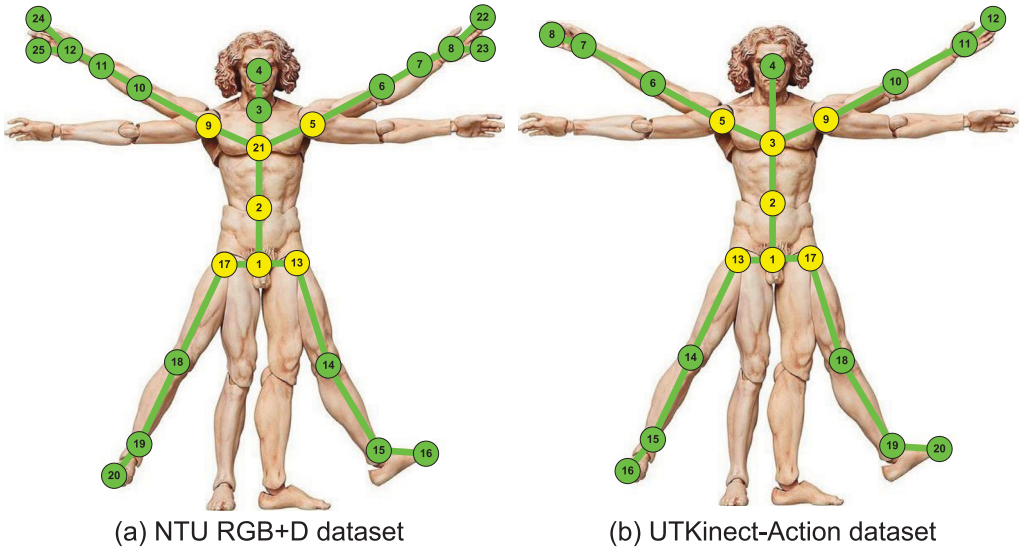

Fig. 2. Configuration of body joints in the NTU RGB+D dataset [29] and UTKinect-Action dataset [17]. Torso joints are colored in yellow. In (a), as an example, the labels of the torso joints are: 1-hip center, 2-middle of the spine, 5-left shoulder, 9-right shoulder, 13-left hip, 17-right hip, 21-spine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

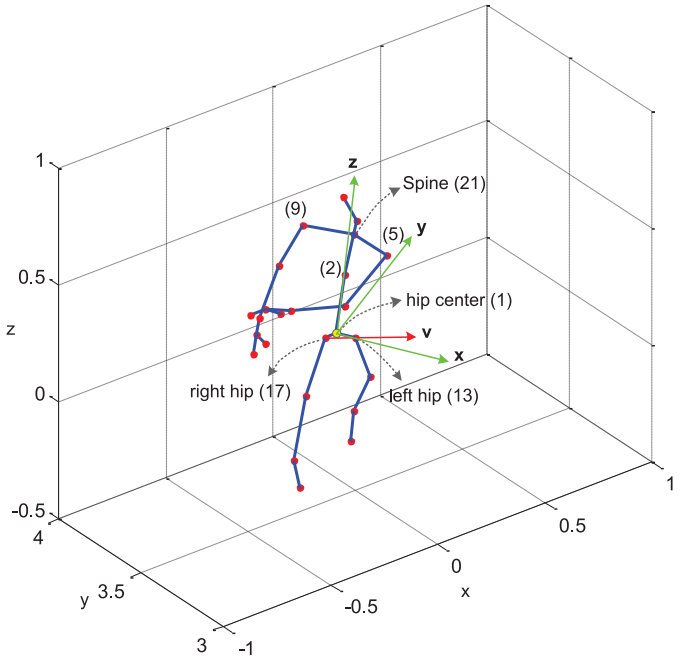

Fig. 3. Illustration of our view invariant transform. **v** denotes the direction from "right hip" to "left hip". For the new coordinate system, "hip center" is selected as origin and **x**, **y**, **z** are the directions of new axes. **z** is aligned with the longer dimension of the torso. **x** is the direction which is vertical to **z** and has minimum angle with **v**. **y** is the product of **z** and **x**. It is noted that all skeletons in a sequence are used to establish one coordinate system. Here, only one skeleton is illustrated to facilitate observation.

Suppose **z** is the first principal component of the result generated by applying PCA to a matrix **M**:

$$\mathbf{M} = \bigcup_{n\in\phi,\ f\in(1,\ldots,F)} p_n^f, \tag{7}$$

where $\phi \in \{1, 2, 5, 9, 13, 17, 21\}$ denotes the set of seven torso joints (see Fig. 2(a)). Matrix **M** is named as the torso matrix, which consists of $7 \times F$ rows and three columns. As shown in Fig. 3, the first principal component **z** is always aligned with the longer dimension of the torso, therefore it is used as the Z axis of the new coordinate system. Note that the orientation pointing from the "hip center" to the "spine" can provide a rough estimation of **z**. However, this result is not accurate, since joint location suffers from the effect of noise.

For the second principal component of **M**, the orientation is expected to denote the X (or Y) axis of the new coordinate system. While, the orientation is not so easily inferred [19]. Instead, we use **x** to denote the X axis of the new coordinate system, which is defined as:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \ arccos\langle \mathbf{x}, \mathbf{v} \rangle, \quad s.t.\ \mathbf{x} \perp \mathbf{z} \tag{8}$$

where vector **v** is defined as:

$$\mathbf{v} = \frac{1}{F}\sum_{f=1}^{F}(p_{17}^f - p_{13}^f), \tag{9}$$

which denotes the mean vector for the direction pointing from "right hip" to "left hip" based on all F frames in a skeleton sequence. The Y axis of the new coordinate system is denoted as $\mathbf{y} = \mathbf{z} \times \mathbf{x}$. Parameters α, β and γ can be determined by transforming $\mathbf{x}, \mathbf{y}, \mathbf{z}$ to $[1, 0, 0]^T$, $[0, 1, 0]^T$, $[0, 0, 1]^T$ using Formula 2. Fig. 4 shows the transformed sequences using our method and [19]. From (e) and (f), we find that our method can preserve more relative motions (e.g., rotations) among skeletons than [19].

Combining view invariant values $\hat{x}, \hat{y}, \hat{z}$ with f and n, the original space Ω_0 is transformed as a new space Ω_1:

$$\Omega_1 = [\hat{x}, \hat{y}, \hat{z}, f, n]^T, \tag{10}$$

which is invariant to viewpoint changes.

4. Enhanced skeleton visualization

4.1. Skeleton visualization

Data visualization [35] refers to the techniques used to communicate data or information by encoding it as visual objects (e.g., points, lines or bars) contained in graphics. One goal of data visualization is to communicate information in high dimensional space

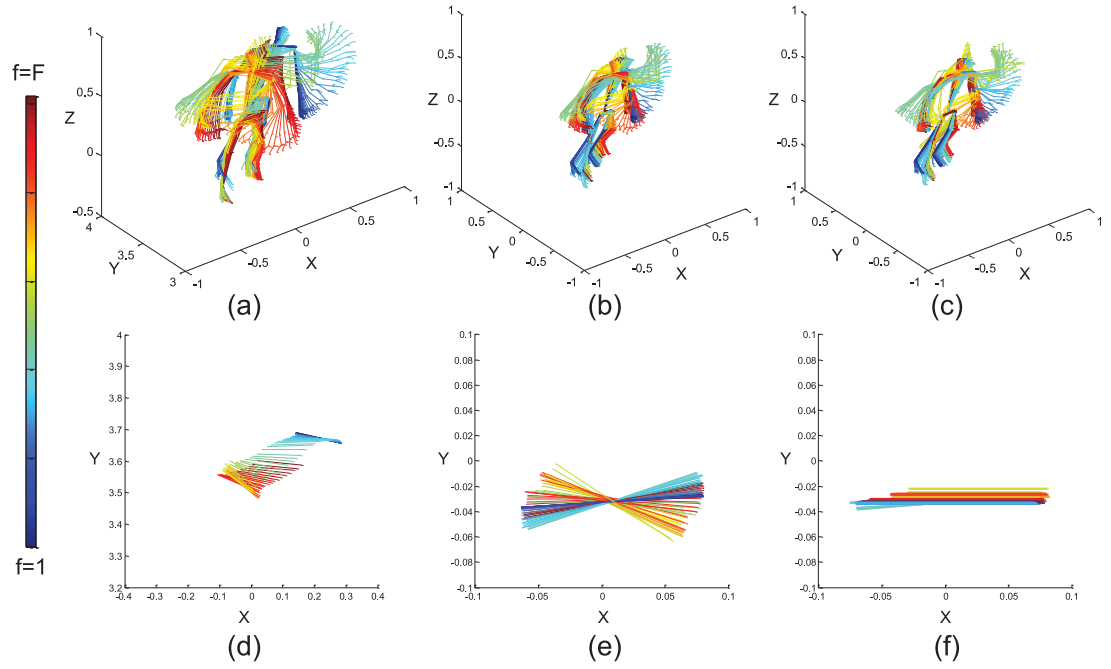

Fig. 4. Comparison between our sequence-based transform and traditional skeleton-based transform [19]. (a) is an original skeleton sequence of "throw". To facilitate observation, each skeleton is colored by inferring to the time label. (b) is the transformed sequence by using our method and (c) is the transformed sequence by using [19]. To show the differences between (a), (b) and (c), each skeleton is simplified as a link between "right hip" joint and "left hip" joint, and the skeleton sequence is correspondingly simplified as a sequence of links. (d), (e) and (f), respectively show the link sequence of (a), (b) and (c) from the view of $X-Y$ plane. (d) indicates that there exist rotations among original skeletons. (e) shows that the rotations are preserved in transformed skeletons, while (f) shows that the rotations are ignored.

clearly and efficiently to users. Diagrams used for data visualization include bar chart, histogram, scatter plot, stream graph, tree map and heat map. Heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors. There are many different color schemes that can be used to illustrate the heat map. Rainbow color maps are often used, as humans can perceive more shades of color than they can of gray, and this would purportedly increase the amount of detail perceivable in the image. In terms of action recognition, Wang et al. [34] proposed an improved rainbow transform to highlight the texture and edges of depth motion maps. However, color maps (like the "jet" color map) generated by rainbow transform have uncontrolled changes in luminance, making some regions (like yellow and cyan regions) appear more prominent than regions of the data that are actually most important [36].

We propose a new type of heat map to visualize spatio-temporal skeleton joints as a series of color images. The key idea is to express a 5D space as a 2D coordinate space and a 3D color space. As shown in Fig. 5, each joint is firstly treated as a 5D point (x, y, z, f, n), where (x, y, z) mean the coordinates, f means the time label and n means the joint label. Function Γ is defined to permute elements of the point:

$$(j, k, r, g, b) = \Gamma\left\{(\hat{x}, \hat{y}, \hat{z}, f, n), c\right\}, \tag{11}$$

where c indicates that function Γ returns the cth type of ranking. There are 10 types of ranking in total. This is because 10 equals to $choose(5, 2)$, where two variables are chosen from five variables to denote image coordinates. We use j and k as local coordinates and use r, g, b as the color values of location (j, k). To this end, r, g, b are normalized to $[0, 255]$. Using the cth type of ranking, three gray images $\mathbf{I}_c^R$, $\mathbf{I}_c^G$ and $\mathbf{I}_c^B$ are constructed as:

$$\left[\mathbf{I}_c^R(j,k)\ \mathbf{I}_c^G(j,k)\ \mathbf{I}_c^B(j,k)\right] = \left[r\ g\ b\right], \tag{12}$$

where $\mathbf{I}_c^R(j,k)$ stands for the pixel value of $\mathbf{I}_c^R$ on location (j, k). Thus, the cth color image is formulated as:

$$\mathbf{I}_c = \left\{\mathbf{I}_c^R\ \mathbf{I}_c^G\ \mathbf{I}_c^B\right\}. \tag{13}$$

Operating function Γ on the point (x, y, z, f, n) can generate $5 \times 4 \times 3 \times 2 \times 1 = 120$ types of ranking. Each type of ranking corresponds to a color image. However, generating so many images needs huge time and computation cost. Moreover, these images may contain redundant information. For example, two images share the same color space (z, f, n) while their coordinate spaces are respectively denoted as (x, y) and (y, x). We observe that one image can be transformed to the other by rotating 90 degrees. In other words, both images encode the same spatio-temporal cues of skeleton joints. For another example, two images share the same coordinate space (x, y) while their color spaces are respectively denoted as (z, f, n) and (z, n, f). We observe that both images are the same in shapes and slight different in colors, indicating that most of the spatio-temporal cues which they encoded are the same. Generally, we argue that permutation in the coordinate space or the color space will generate similar images. Therefore, this paper uses ten types of ranking shown in Fig. 5. These ranking results ensure that each element of the point (x, y, z, f, n) can be assigned to the coordinate space and the color space. Fig. 6(d) shows the ten color images extracted from an action "throw", where both spatial and temporal information of skeleton joints are encoded in these images. Fig. 6 also compares our method with [20] and [21], which can be considered as two specific cases of our visualization method. As can be seen, images in (b) are similar to sub-figure #1, #2, #5 in (d). These images mainly reflect the spatial distribution of skeleton joints. The image in (c) is similar to sub-figure #10 in (d). This image mainly reflects the temporal evolution of skeleton joints. In (d), those sub-figures highlighted by red bounding boxes provide distinct spatial and temporal distributions, which have never been explored by previous works, e.g., [20] and [21].

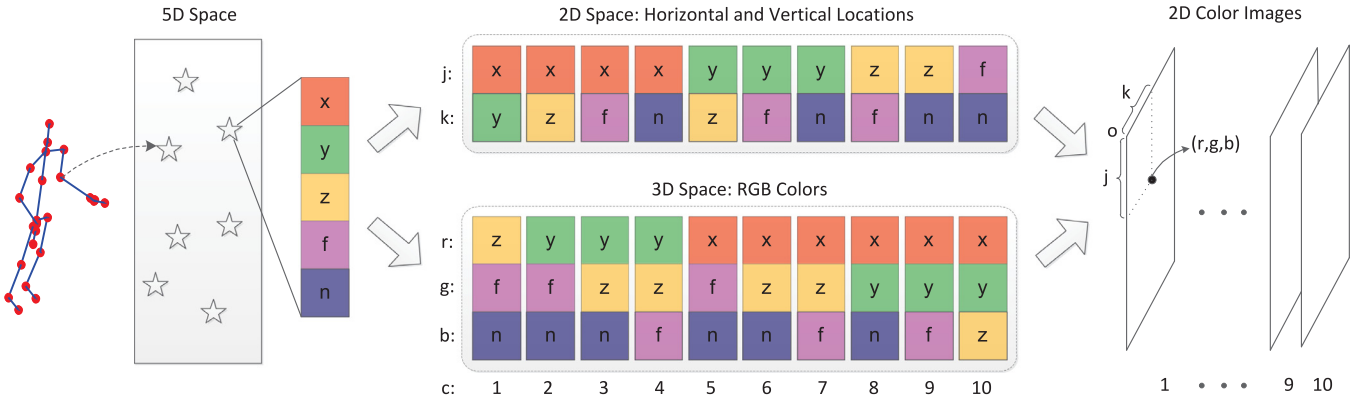

Fig. 5. Pipeline of our data visualization method.

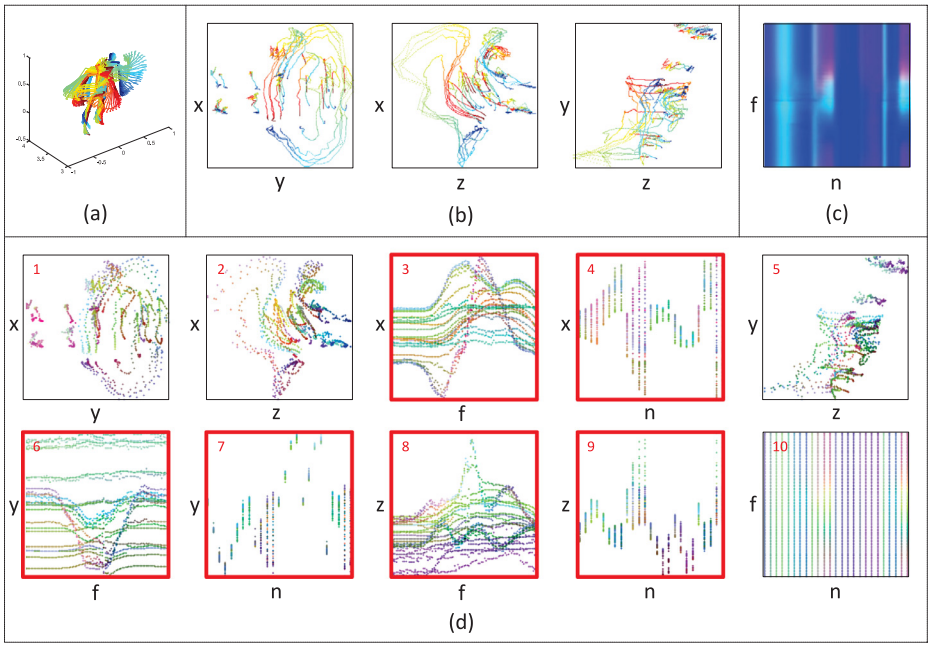

Fig. 6. Illustration of color images generated by different data visualization methods. (a) shows skeletons of an action "throw". (b), (c) and (d), respectively shows color images generated by [20], [21] and our method. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

4.2. Visual enhancement

As shown in Fig. 7(a), the visual patterns of color images are sparse, due to the limited number of skeleton joints. To enhance visual patterns, we introduce mathematical morphology (MM) [37], which is a theory and technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions. MM is most commonly applied to digital images, but it can be employed as well on graphs, surface meshes, solids, and many other spatial structures. The basic morphological operators are erosion, dilation, opening and closing, where the erosion operator means to probe a binary image with a simple, pre-defined shape, drawing conclusions on how this shape misses the shapes in the image. This simple "probe" is called the structuring element, and is itself a binary image (i.e., a subset of the space or grid). Specifically, the erode operator $\ominus$ is defined as:

$$\mathbf{A} \ominus \mathbf{E} = \bigcap_{e \in \mathbf{E}} \mathbf{A}_{-e}, \tag{14}$$

where $\mathbf{A}$ is a binary image and $\mathbf{E}$ is a structuring element.

To enlarge regions of colored pixels, we apply the erosion operator on $\mathbf{I}_c$:

$$\tilde{\mathbf{I}}_c = \left\{ \mathbf{I}_c^R \ominus \mathbf{E} \quad \mathbf{I}_c^G \ominus \mathbf{E} \quad \mathbf{I}_c^B \ominus \mathbf{E} \right\}, \tag{15}$$

where each channel of $\mathbf{I}_c$ is eroded and then composed to form the eroded color image. Fig. 7(b) shows color images processed by erosion operator. Compared with the initial images (Fig. 7(a)), textures in processed images are enhanced. Note that we set $\mathbf{E}$ as an open disk of radius 5, centered at the origin.

4.3. Motion enhancement

Intuitively, human tends to pay attention to moving objects and ignore static parts. Motivated by this human nature, informative joints with salient motions are selected to represent original skeleton sequences [18]. With selected joints, similar actions can be distinguished easier, since these joints are more directly related to actions than those joints which keep nearly static. Therefore, we highlight the effect of motion on generating color images by weighting skeleton joints according to their motions.

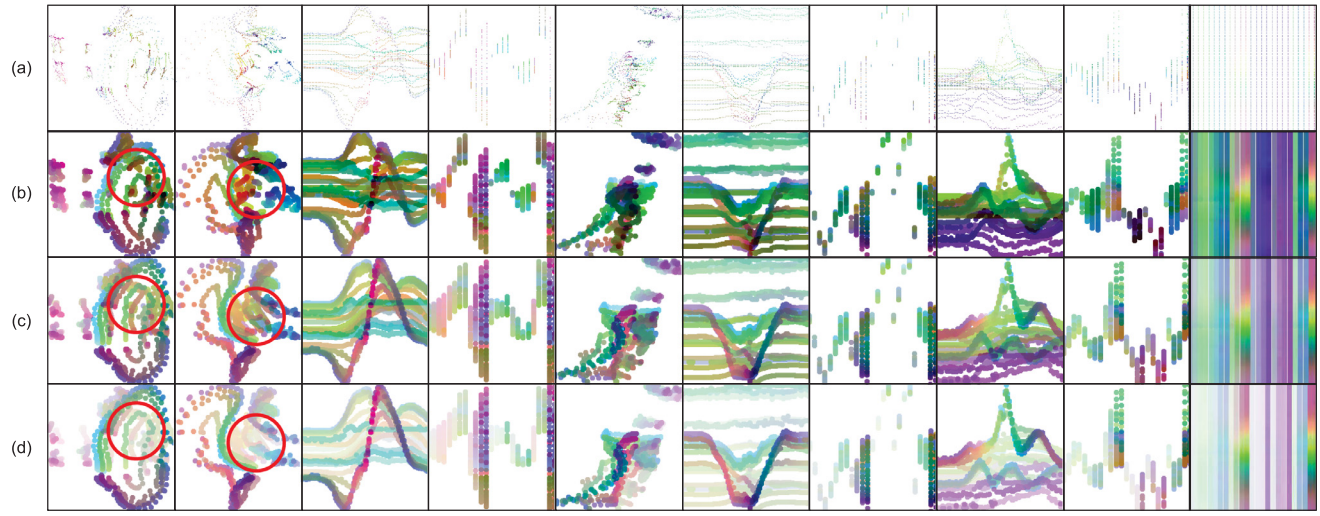

Fig. 7. A skeleton sequence "throw" is visualized as color images. (a) shows the initially obtained color images. (b) is processed by visual enhancement. (c) and (d) are processed by both visual enhancement and motion enhancement, where $\rho = 0.5$ for (c) and $\rho = 1$ for (d). See text for explanation of red circled regions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Given a skeleton joint $p_n^f = (x_n^f, y_n^f, z_n^f)^{\mathrm{T}}$, we estimate its motion energy by:

$$\xi_n^f = ||p_n^f - p_n^{f-1}||, \tag{16}$$

where $f > 1$ and operator $||\cdot||$ calculates the Euclidean metric. The accumulated motion energy of p_n^f is defined as:

$$\xi_n = \sum_{f=2}^{F} \xi_n^f. \tag{17}$$

Fig. 8 shows weighted skeleton joints, where skeleton joints with larger weights are colored in brighter red. Obviously, skeleton joints in brighter red are more related to the action "throw".

To control the effect of motion on color images, we introduce a parameter ρ and define the weight of p_n^f as:

$$w_n = \rho \cdot norm\{\xi_n\} + (1 - \rho), \tag{18}$$

where $0 \le \rho \le 1$ and function *norm* normalizes ξ_n to [0, 1]. Suppose one pixel is generated by the nth joint and its color values are denoted as $[r\ g\ b]$. Then, we use w_n to weight the color values:

$$[\tilde{r}\ \tilde{g}\ \tilde{b}] = (1 - w_n)[255\ 255\ 255] + w_n[r\ g\ b], \tag{19}$$

where pixel with larger w_n will mostly keep its original color, and the color of pixel with smaller w_n will fade (turns from original color to white). Fig. 7(c) and (d) show color images generated by weighted skeleton joints. Red circles highlight the regions which are dramatically affected by using different weights. We observe that the colors of pixels, generated by joints with small motions, tend to fade when the parameter ρ increases. In this way, those pixels which are generated by joints with salient motions are emphasized. As shown in the first column of Fig. 7(d), the highlighted joints on the hands are more related to the action "throw".

5. Multi-stream CNN fusion

To obtain more discriminative feature from spatio-temporal skeleton joints, we propose a multiple CNN-based model to extract deep features from color images generated in previous section. Inspired by two-stream deep networks [33], the proposed model (shown in Fig. 9) involves 10 modified AlexNet [38], where each CNN uses one type of color images as input. The posterior probabilities generated from each CNN are fused as the final class score.

For an input sequence $\mathcal{I}^m$, we obtain a series of color images: $\{\mathbf{I}_c^m\}_{c=1}^{10}$. Each image is normalized to 224 × 224 pixels to take advantage of pre-trained models. Mean removal is adopted for all input images to improve the convergence speed. Then, each color image is processed by a CNN. For the image $\mathbf{I}_c^m$, the output Y_c of the last fully-connected (*fc*) layer is normalized by the softmax function to obtain the posterior probability:

$$prob(l \mid \mathbf{I}_c^m) = \frac{e^{\Upsilon_c^l}}{\sum_{k=1}^{L} e^{\Upsilon_c^k}}, \tag{20}$$

which indicates the probability of image $\mathbf{I}_c^m$ belonging to the lth action class. L is the number of total action classes.

The objective function of our model is to minimize the maximum-likelihood loss function:

$$\mathcal{L}(\mathbf{I}_c) = -\sum_{m=1}^{M} ln \sum_{l=1}^{L} \delta(l - s_m)\ prob(l \mid \mathbf{I}_c^m), \tag{21}$$

where function δ equals one if $l = s_m$ and equals zero otherwise, s_m is the real label of $\mathbf{I}_c^m$, M is the batch size. For sequence $\mathcal{I}$, its class score is formulated as:

$$score(l \mid \mathcal{I}) = \frac{1}{10} \sum_{c=1}^{10} prob(l \mid \mathbf{I}_c), \tag{22}$$

where $score(l \mid \mathcal{I})$ is the average of the outputs from all ten CNN and $prob(l \mid \mathbf{I}_c)$ is the probability of image $\mathbf{I}_c$ belonging to the lth action class. To explore the complementary property of deep features generated from each CNN, we introduce a weighted fusion method:

$$score(l \mid \mathcal{I}) = \frac{1}{10} \sum_{c=1}^{10} \eta_c\ prob(l \mid \mathbf{I}_c), \tag{23}$$

where η_c equals to one or zero, indicating whether the cth CNN is selected or not. Therefore, $score(l \mid \mathcal{I})$ is the fused class score based on the selected CNNs. The method of choosing parameter η_c is discussed in Section 6.6.4. In following, these two types of fusion strategies are respectively named as average fusion and weighted fusion methods.

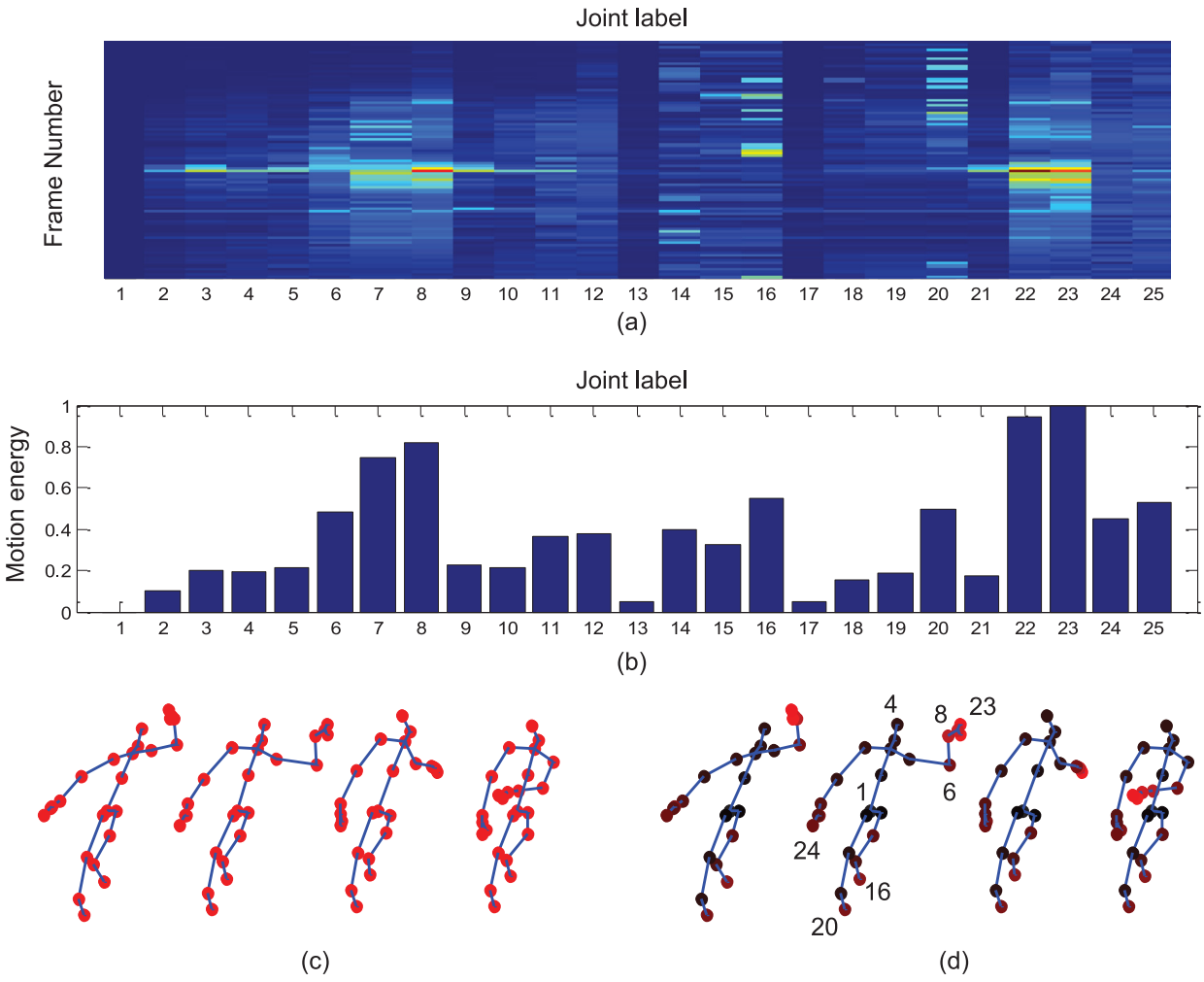

Fig. 8. Illustration of weighing skeleton joints according to the motion energies. In (a), the nth row and fth column shows the motion energy of skeleton joint p_n^f. Note that the joint label indices indicate joints shown in Fig. 2 (a). In (b), the nth bar shows the accumulated motion energy of the nth skeleton joint. (c) shows several snaps from a skeleton sequence "throw". (d) shows several weighted snaps, where skeleton joints with larger weights are colored in brighter red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Experiments and discussions

Our method is evaluated on Northwestern-UCLA [39], UWA3DII [40], NTU RGB+D [29] and MSRC-12 [41] datasets. The first three datasets contain severe view variations and noisy skeleton joints. The NTU RGB+D dataset is so far the largest dataset for skeleton-based action recognition, which contains challenges like inter-similarities and intra-varieties. Since related works [20,21] cannot tackle with view variations, we ensure fair comparison between our method and them on MSRC-12 dataset.

6.1. Implementation details

In our model, each CNN contains five convolutional layers and three *fc* layers. The first and second *fc* layers contain 4096 neurons, and the number of neurons in the third one is equal to the total number of action classes. Filter sizes are set to 11 × 11, 5 × 5, 3 × 3, 3 × 3, 3 × 3. Local Response Normalisation (LRN), max pooling and ReLU neuron are adopted and the dropout regularisation ratio is set to 0.5. The network weights are learned using the mini-batch stochastic gradient descent with the momentum value set to 0.9 and weight decay set to 0.00005. Learning rate is set to 0.001 and the maximum training cycle is set to 200. In each cycle, a mini-batch of 50 samples is constructed by randomly sampling 50 images from training set. The implementation is based on MatConvNet [42] with one NVIDIA GeForce GTX 1080 card and 8 G RAM.

Our model can be directly trained using samples from the training set. In this case, all layers are randomly initialized from [0, 0.01]. To increase the number of training samples, we randomly flip the training sequences about the y-axis to generate two sequences from each sequence. It is noted that these training sequences have already been processed by the sequence-based transform method and the synthesized sequences are further visualized as color images and served as training samples. This type of data augmentation method is a standard procedure in CNN to help the model learning better due to limited number of training samples [43]. Let *Original Samples* and *Synthesized Samples* respectively denote above two settings. Instead of training the CNN model from scratch using the training samples of each action dataset, we can also take advantage of pre-trained models on large scale image datasets such as ImageNet, and fine tune our model. Specifically, we fine tune our model by initializing the third *fc* layer from [0, 0.01] and initializing other layers from pre-trained model on ILSVRC-2012 (Large Scale Visual Recognition Challenge 2012). Let *Synthesized+Pre-trained* denote fine tune our model with synthe-

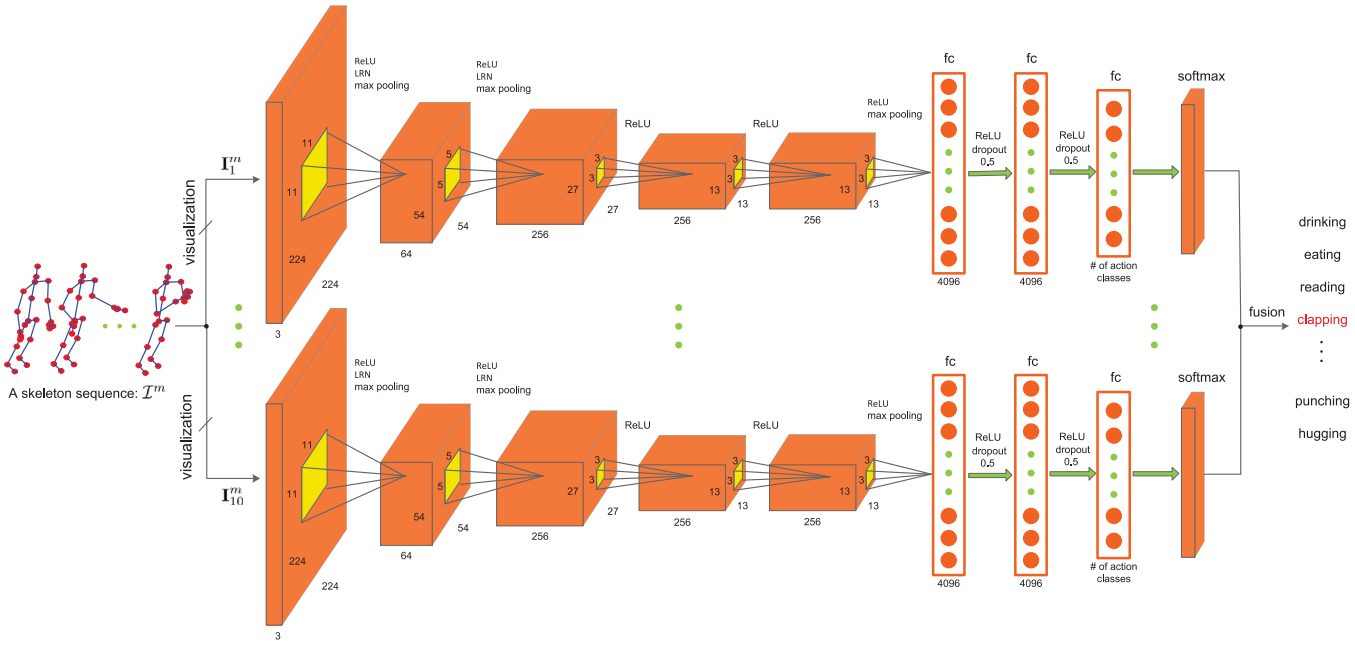

Fig. 9. Proposed skeleton-based action recognition using multi-stream CNN.

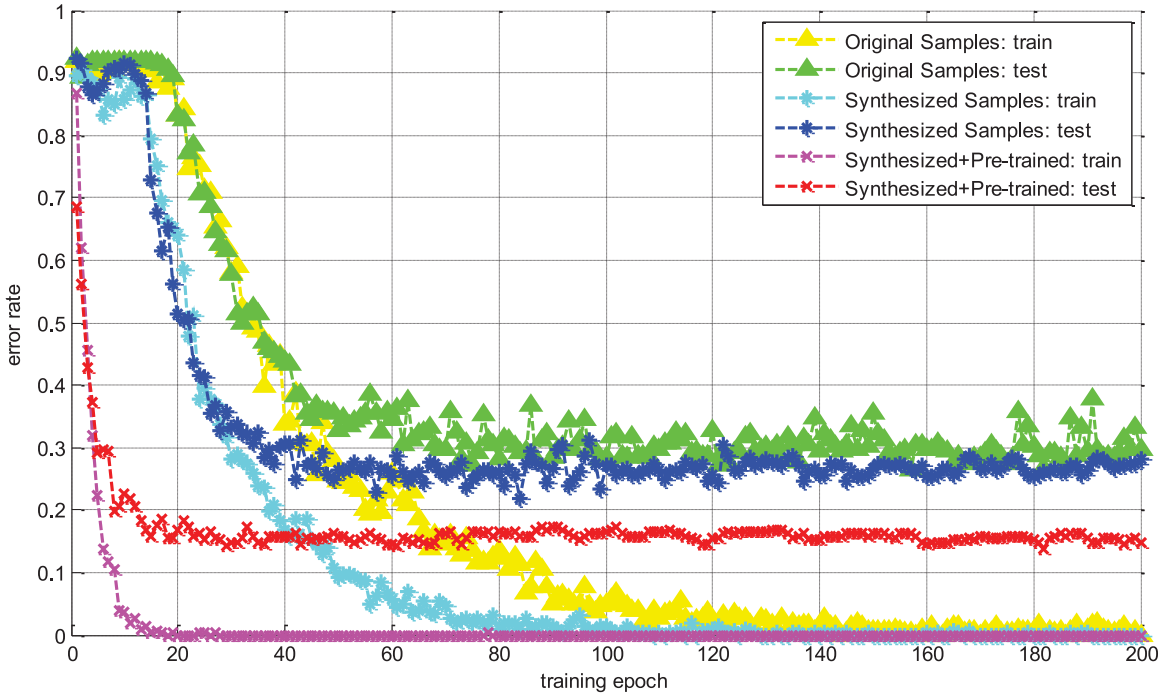

Fig. 10. Convergence curves on the MSRC-12 dataset [41]. The first type of color image is used as input for CNN. Error rate almost converges when the training epoch equals to 200.

sized samples. Fig. 10 shows the convergence curves on the MSRC-12 dataset [41], where the error rate trends to converge when the training epoch grows to 200. This result shows the effectiveness of our implementations for CNN model.

6.2. Northwestern-UCLA dataset

The Northwestern-UCLA dataset [39] contains 1494 sequences covering 10 action categories: “pick up with one hand”, “pick up with two hands”, “drop trash”, “walk around”, “sit down”, “stand up”, “donning”, “doffing”, “throw” and “carry”. Each action is performed one to six times by ten subjects. This dataset contains data taken from a variety of viewpoints (see Fig. 11). Following [39], we use samples from the first two cameras as training data, and the samples from the third camera as test data.

Table 1 shows overall recognition accuracies of various methods. According to input data, these methods can be categorized into depth-based, skeleton-based and hybrid-based methods. Here, the hybrid data includes depth and skeleton data. According to the type of extracted features, we can also classify these methods into hand-crafted-based, RNN-based and CNN-based methods. Since we use sole skeleton data, those skeleton-based methods, i.e., HOJ3D [17], LARP [27] and HBRNN-L [28], are most related to our

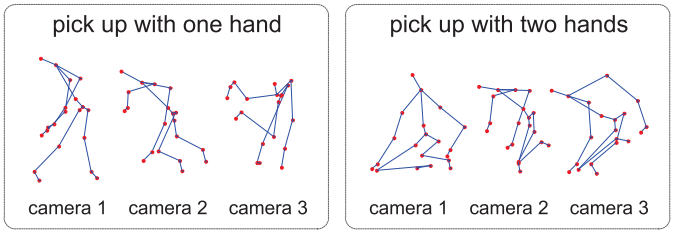

Fig. 11. Skeleton snaps from the Northwestern-UCLA dataset [39].

Table 1
Results on the Northwestern-UCLA dataset [39] (cross view protocol [39]).

Data	Feature	Method	Accuracy(%)
Depth	Hand-crafted	HON4D [44]	39.90
		SNV [45]	42.80
		AOG [39]	53.60
		HOPC [40]	80.00
	CNN	HPM+TM [46]	92.00
Hybrid Skeleon	Hand-crafted	AE [47]	76.00
	Hand-crafted	HOJ3D [17]	54.50
		LARP [27]	74.20
	RNN	HBRNN-L [28]	78.52
	CNN	*Original Samples* (*ours*)	86.09
		Synthesized Samples (*ours*)	89.57
		Synthesized+Pre-trained (*ours*)	**92.61**

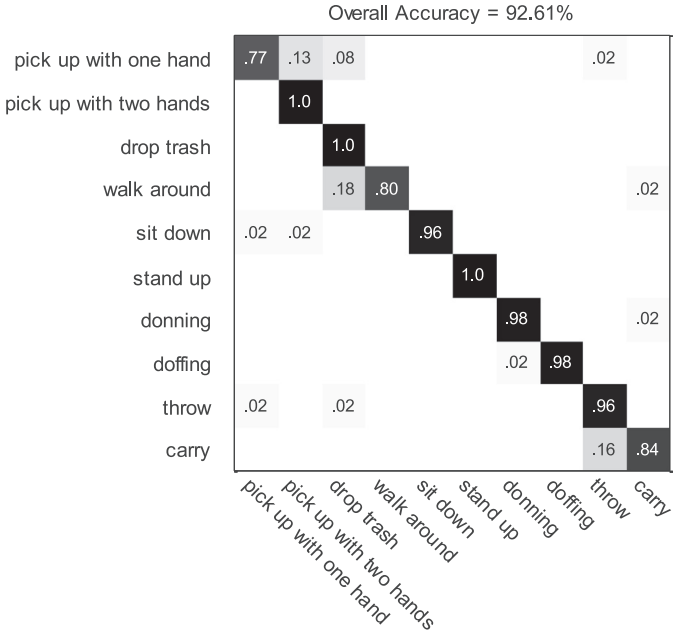

Fig. 12. Confusion matrix on the Northwestern-UCLA dataset [39].

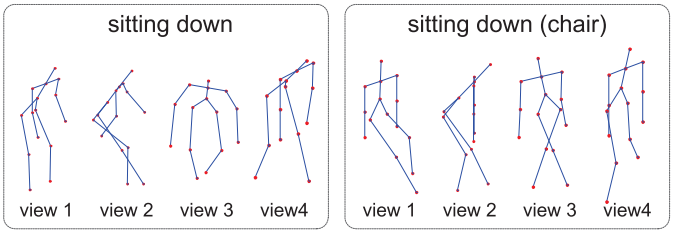

Fig. 13. Skeleton snaps from the UWA3DII dataset [40].

method. HOJ3D [17] is designed to tackle with viewpoint changes. However, HOJ3D only achieves 54.50% on this dataset. The reason is that each skeleton is assumed to be vertical to the ground [17]. Therefore, HOJ3D can barely perform well with various views, such as top view in this dataset. LARP [27] performs better than HOJ3D, since LARP models relationships among skeletons by transform parameters which suffer less from view changes. However, the temporal information among skeletons are not properly encoded by LARP. To this end, HBRNN-L [28] models dynamic information among skeletons and achieves 78.52%.

Our method using *Original Samples* for training achieves 86.09%, which outperforms LARP by 11.89% and outperforms HBRNN-L by 7.57%. The reason is that our method implicitly encodes both spatial and temporal relationships among skeletons. Moreover, skeleton joints are transformed to be view invariant, therefore our method suffers less from view changes. Since *Synthesized Samples* provides more data for training, it achieves 3.48% higher than *Original Samples*. *Synthesized+Pre-trained* achieves state-of-the-art result of 92.61%, which outperforms all other comparison methods. This result shows the superiority of initializing our CNN with pre-trained model. The confusion matrix of our method is shown in Fig. 12, where action "pick up with one hand" and action "pick up with two hands" have large confusion with each other because both actions contain similar motions and appearances, shown in Fig. 11. For similar reason, action "drop trash" and action "walk around" also have high confusion.

6.3. UWA3DII dataset

The UWA3DII dataset contains 30 human actions performed four times by ten subjects. Each action is observed from front view, left and right side views, and top view. The dataset is challenging because of varying viewpoints, self-occlusion and high similarity among actions. For example, action "drinking" and action "phone answering" have slightly different in the location of the hand. Action "sitting down" and action "sitting down (chair)" are also similar (see Fig. 13), since the chair is not captured in skeleton data. For cross-view action recognition, we follow the cross view protocol in [40], which uses samples from two views as training data, and samples from the two remaining views as test data. Table 2 shows overall recognition accuracies of different methods. Our method achieves best performances under all types of settings. Based on the mean performance, our method outperforms the second best method HOPC [40] by a significant margin, which is 21.6%.

6.4. NTU RGB+D dataset

The NTU RGB+D dataset [29] contains 60 actions performed by 40 subjects from various views (Fig. 14(a)), generating 56,880 skeleton sequences. This dataset also contains noisy skeleton joints (see Fig. 14(b)), which bring extra challenge for recognition. Following the cross subject protocol in [29], we split the 40 subjects into training and testing groups. Each group contains samples captured from different views performed by 20 subjects. For this evaluation, the training and testing sets have 40,320 and 16,560 samples, respectively. Following the cross view protocol in [29], we use all the samples of camera 1 for testing and samples of cameras 2 and 3 for training. The training and testing sets have 37,920 and 18,960 samples, respectively.

Table 3 shows the performances of various methods on this dataset. Since this dataset provides rich samples for training deep models, the RNN-based methods, e.g., ST-LSTM [29], achieves high accuracy. Our method achieves nearly 10% higher than ST-LSTM [29] for both cross subject and cross view protocols. The confusion matrix is shown in Fig. 15. This result shows the effectiveness of our method to tackle with challenges like view variations and noisy skeletons in large scale of data.

Table 2
Results on the UWA3DII dataset [40] (cross view protocol [40]).

Data	Training views	V_1 & V_2		V_1 & V_3		V_1 & V_4		V_2 & V_3		V_2 & V_4		V_3 & V_4		Mean
	Test view	V_3	V_4	V_2	V_4	V_2	V_3	V_1	V_4	V_1	V_3	V_1	V_2	
RGB	AOG [39]	47.3%	39.7%	43.0%	30.5%	35.0%	42.2%	50.7%	28.6%	51.0%	43.2%	51.6%	44.2%	42.3%
Depth	HON4D [44]	31.1%	23.0%	21.9%	10.0%	36.6%	32.6%	47.0%	22.7%	36.6%	16.5%	41.4%	26.8%	28.9%
	SNV [45]	31.9%	25.7%	23.0%	13.1%	38.4%	34.0%	43.3%	24.2%	36.9%	20.3%	38.6%	29.0%	29.9%
	HOPC [40]	52.7%	51.8%	59.0%	57.5%	42.8%	44.2%	58.1%	38.4%	63.2%	43.8%	66.3%	48.0%	52.2%
Skeleton	HOJ3D [17]	15.3%	28.2%	17.3%	27.0%	14.6%	13.4%	15.0%	12.9%	22.1%	13.5%	20.3%	12.7%	17.7%
	AE [47]	45.0%	40.4%	35.1%	36.9%	34.7%	36.0%	49.5%	29.3%	57.1%	35.4%	49.0%	29.3%	39.8%
	LARP [27]	49.4%	42.8%	34.6%	39.7%	38.1%	44.8%	53.3%	33.5%	53.6%	41.2%	56.7%	32.6%	43.4%
	***Original Samples* (*ours*)**	66.4%	68.1%	56.8%	66.1%	58.8%	66.2%	74.2%	67.0%	76.9%	64.8%	72.2%	54.0%	66.0%
	***Synthesized Samples* (*ours*)**	69.9%	72.3%	59.6%	69.6%	60.7%	68.4%	79.3%	71.2%	79.1%	68.3%	76.6%	58.5%	69.5%
	***Synthesized+Pre-trained* (*ours*)**	**72.3%**	**76.3%**	**64.7%**	**75.5%**	**63.5%**	**74.0%**	**83.1%**	**75.1%**	**82.4%**	**71.1%**	**83.5%**	**63.5%**	**73.8%**

Fig. 14. Skeleton snaps from the NTU RGB+D dataset [29].

Table 3
Results on the NTU RGB+D dataset [29] (protocols of [29]).

Data	Feature	Method	Cross subject(%)	Cross view(%)
Depth	Hand-crafted	HON4D [44]	30.56	7.26
		SNV [45]	31.82	13.61
Hybrid	Hand-crafted	HOG^2 [48]	32.24	22.27
Skeleton	Hand-crafted	Skeletal quads [49]	38.62	41.36
		LARP [27]	50.08	52.76
		Dynamic skeletons [50]	60.23	65.22
	RNN	HBRNN-L [28]	59.07	63.97
		Deep RNN [29]	56.29	64.09
		Deep LSTM [29]	60.69	67.29
		Part-aware LSTM [29]	62.93	70.27
		ST-LSTM [32]	61.70	75.50
		ST-LSTM+TG [32]	69.20	77.70
	CNN	*Original Samples* (*ours*)	75.97	82.56
		Synthesized Samples (*ours*)	77.69	83.67
		Synthesized+Pre-trained (*ours*)	**80.03**	**87.21**

6.5. MSRC-12 dataset

The MSRC-12 dataset [41] contains 594 sequences, i.e., 719,359 frames (approx. 6 h 40 min), collected from 30 people performing 12 gestures. This is a single view dataset, i.e., action samples are captured from a single view. Therefore, the sequence-based transform method is not used to implement our method on this dataset. Following the cross-subject protocol in [20], we use sequences performed by odd subjects for training and even subjects for testing. In Table 4, ConvNets [21] and JTM [20] are most related to our visualization method. By extracting deep features, [21] achieves 84.46% and [20] achieves 93.12% on this dataset. Our method achieves accuracy of 96.62% (see Fig. 16), which outperforms these methods by 12.16 and 3.50%. The reason is that our method can properly encode both temporal and spatial cues of skeleton joints, while ConvNets [21] and JTM [20] overemphasize either the temporal information or the spatial information. More detailed analysis can be found in Section 6.6.4.

Table 4
Results on the MSRC-12 dataset [41] (cross subject protocal [20]).

Data	Feature	Method	Accuracy(%)
Skeleton	Hand-crafted	ELC-KSVD [51]	90.22
		Cov3DJ [15]	91.70
	CNN	ConvNets [21]	84.46
		JTM [20]	93.12
		Original Samples (*ours*)	93.92
		Synthesized Samples (*ours*)	94.93
		Synthesized+Pre-trained (*ours*)	**96.62**

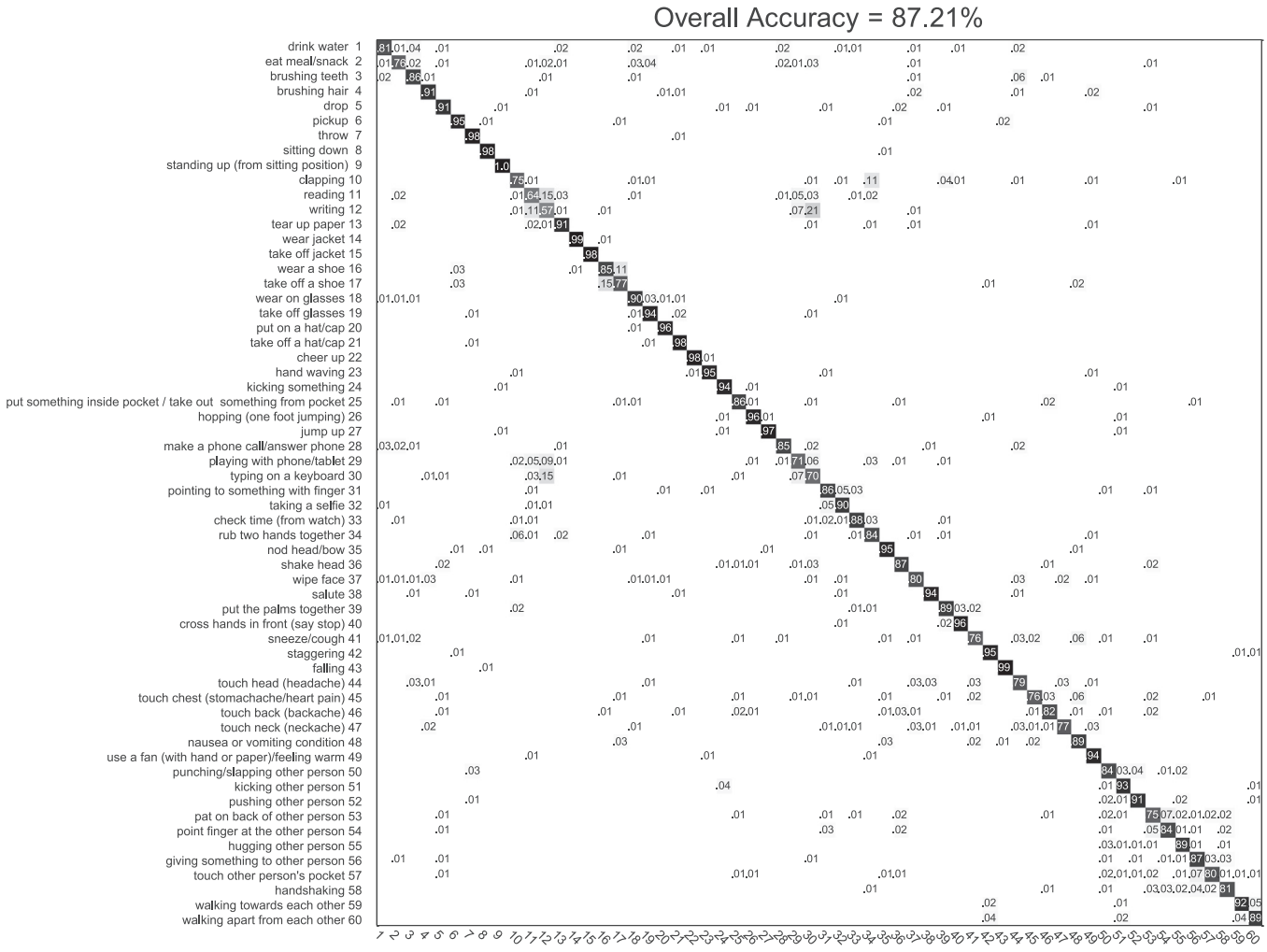

Fig. 15. Confusion matrix on the NTU RGB+D dataset [29].

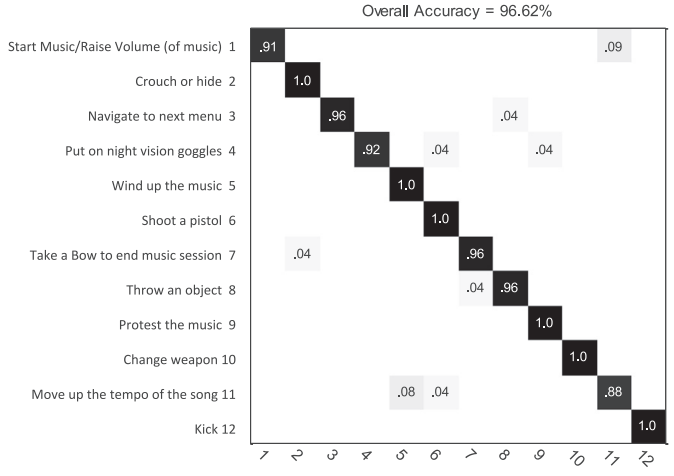

Fig. 16. Confusion matrix on the MSRC-12 dataset [41].

6.6. Evaluation of individual components

We use five settings to evaluate each component of our method. As shown in Table 5, the *Data Visualization* is used as a baseline method and other four settings are variations of the baseline method.

6.6.1. Sequence-based transform

Table 6 evaluates view invariant transform methods. The results show that view invariant transform methods outperform the *Data Visualization* which uses the original skeleton sequences on multi-view action datasets, e.g., Nothwestern-UCLA and NTU RGB+D. We also observe that our *Sequence-Based Transform* outperforms *Skeleton-Based Transform* on both multi-view datasets. The reason is that our method preserves more spatio-temporal cues and shows more robustness to noisy data.

6.6.2. Visual enhancement

Table 7 evaluates visual enhancement method. By enhancing the visual patterns of color images, *Visual Enhancement* method respectively achieves 3.27% and 4.18% higher than *Data Visualization* on Northwestern-UCLA and NTU RGB+D datasets. These improvements verify the validity of using mathematical morphology to conduct visual enhancement.

6.6.3. Motion enhancement

Table 8 evaluates motion enhancement method. *Motion Enhancement* with parameter ρ=1 respectively achieves 3.92% and 3.71% higher than *Data Visualization* on Northwestern-UCLA and NTU RGB+D datasets. This result indicates that skeleton joints with salient motions show more distinctive power to represent actions than those skeleton joints which keep mostly static.

Table 5
Five settings for evaluating different components of our proposed metho d.

Data Visualization	A skeleton sequence ⇒ data visualization (Section 4.1) ⇒ multiple CNN (Section 5) ⇒ weighted fusion (Section 5)
Skeleton-Based Transform	A skeleton sequence skeleton-based transform [13] ⇒ data visualization (Section 4.1) ⇒ multiple CNN (Section 5) ⇒ weighted fusion (Section 5)
Sequence-Based Transform	A skeleton sequence ⇒ sequence-based transform (Section 3) ⇒ data visualization (Section 4.1) ⇒ multiple CNN (Section 5) ⇒ weighted fusion (Section 5)
Visual Enhancement	A skeleton sequence ⇒ data visualization (Section 4.1) ⇒ visual enhancement (Section 4.2) ⇒ multiple CNN (Section 5) ⇒ weighted fusion (Section 5)
Motion Enhancement	A skeleton sequence ⇒ data visualization (Section 4.1) ⇒ motion enhancement (Section 4.3) ⇒ multiple CNN (Section 5) ⇒ weighted fusion (Section 5)

Table 6
Evaluation of view invariant transform.

Method	Dataset	
	Nothwestern-UCLA (Cross View)(%)	NTU RGB+D (Cross View)(%)
Data Visualization	85.43	80.36
Skeleton-Based Transform	87.61	82.25
Sequence-Based Transform	**91.52**	**84.27**

Table 7
Evaluation of visual enhancement

Method	Dataset	
	Nothwestern-UCLA (Cross View)(%)	NTU RGB+D (Cross View)(%)
Data Visualization	85.43	80.36
Visual Enhancement	**88.70** ($r = 5$)	**84.54** ($r = 5$)

Table 8
Evaluation of motion enhancement.

Method	Dataset	
	Nothwestern-UCLA (Cross View)(%)	NTU RGB+D (Cross View)(%)
Data Visualization	85.43	80.36
Motion Enhancement	**89.35** ($\rho = 1$)	**84.07** ($\rho = 1$)

6.6.4. Decision level fusion

Table 9 evaluates average fusion and weighted fusion on four datasets. As can be seen, the results of average fusion outperforms that of each single type. This result indicates that different types of color images show complementary property to each other. It is interesting to find that the significance of each type varies from different datasets. For example, the 3rd channel outperforms the 2nd channel on NTU RGB+D dataset (with cross view protocol), while the result is opposite on MSRC-12 dataset. This observation motivates us to apply the weighted fusion method, which select proper type of features for fusion. In practice, a five-fold validation method is applied on training samples to learn values of η_c, using which best accuracy is achieved. In Table 9, the selected types are colored in green and the discarded types are colored in red. With these selected features, the weighted fusion outperforms the average fusion on all datasets, verifying the effect of feature selection.

We find that the 1st and 6th types are always selected for all datasets. This phenomenon shows that these types show distinctive power to identify similar actions. Fig. 17 shows the 6th type of color image representing 12 types of actions from MSRC-12 dataset. Obviously, these images contain distinct patterns which benefit the recognition of actions. As mentioned before, the color image in [21] is similar to the 10th type, which is not always selected for fusion. The color images in [20] are similar to the 1st, 2nd and 5th types, where only the 1st type is always selected for fusion. From the selected types, we claim that color images developed in our method can encode more sufficient and more distinct spatio-temporal information than [20] and [21].

6.7. Evaluation of robustness to partial occlusions

SmartHome dataset[1] is collected by our lab, which contains six types of actions: “box”, “high wave”, “horizontal wave”, “curl”, “circle”, “hand up”. Each action is performed 6 times (three times for each hand) by 9 subjects in 5 situations: “sit”, “stand”, “with a pillow”, “with a laptop”, “with a person”, resulting in 1620 depth sequences. Skeleton joints in SmartHome dataset contain much noises, due to the effect of occlusions. The noisy skeleton snaps of action “*wave*” are illustrated in Fig. 18.

For evaluation, we use subjects #1, 3, 5, 7, 9 for training and subjects #2, 4, 6, 8, 10 for testing. On this dataset, JTM [20] and ConvNets [21] achieve 71.11% and 67.22%, respectively. These results show that noise brought by occlusions increases the ambiguities among similar skeleton sequences. Our *Synthesized+Pre-trained* achieves an accuracy of 78.61% on this dataset. The improvements over [20] and [21] verify that our method is robust to partial occlusions to some extend.

6.8. Evaluation of parameters

Fig. 19 shows the effects of parameter r and ρ. The left figure shows the accuracy of *Visual Enhancement* method with parameter r ranging from 0 to 10 at an interval of 1. Parameter r stands for the radius of the structuring element **E**. From the left figure, we find that the accuracy of *Visual Enhancement* firstly increases and then drops when the value of r grows larger. The reason is illustrated in Fig. 20, which shows the effect of r on the first type of color images. Obviously, the visual patterns become more salient when r changes from 0 to 5. While, details of visual patterns become ambiguous when r changes from 5 to 10. In other words, proper value of r should be selected to ensure the saliency of visual patterns. As Fig. 19 suggests, we set r to 5 as default value, using which we achieve highest accuracies on both datasets. The right figure shows that the accuracy of *Motion Enhancement* trends to increase with parameter ρ growing from 0 to 1, therefore the default value of ρ is set to 1, which means directly using the motion energy of joint as weight.

6.9. Evaluation of computation time

On the Northwestern-UCLA dataset, the computation time of our method is tested with default parameters of $r = 5$ and $\rho = 1$. The average computational time required for extracting an enhanced color image is 0.0484 s on a 2.5 GHz machine with 8 GB RAM, using Matlab R2014a. The total training time is 26525.75 s

[1] The proposed dataset is provided in https://github.com/NewDataset/dataset.git.

Table 9
Evaluation of average fusion and weighted fusion. The types of color images colored in green are selected to generate the weighted fusion results.

Dataset	Protocol		Type										Fusion	
			1	2	3	4	5	6	7	8	9	10	Average	Weighted
Northwestern-UCLA	Cross View		86.30%	78.26%	81.96%	78.26%	84.35%	88.04%	87.17%	68.04%	61.74%	83.26%	90.44%	**92.61%**
NTU RGB+D	Cross View		74.97%	68.83%	74.52%	71.27%	73.17%	79.44%	73.17%	67.20%	65.21%	84.03%	86.70%	**87.21%**
NTU RGB+D	Cross Subject		64.13%	60.85%	59.58%	59.28%	65.79%	65.54%	62.75%	55.18%	57.05%	73.45%	79.81%	**80.03%**
MSRC-12	Cross Subject		85.47%	81.76%	78.04%	72.64%	93.24%	86.15%	83.11%	78.72%	76.35%	84.46%	94.60%	**96.62%**
UWA3DII	V_1&V_2	V_3	51.65%	34.71%	30.17%	38.84%	57.85%	61.98%	50.00%	38.12%	40.91%	42.56%	70.25%	**72.31%**
	V_1&V_2	V_4	60.47%	36.76%	40.32%	45.85%	36.36%	51.38%	47.83%	23.32%	28.85%	40.32%	73.91%	**76.29%**
	V_1&V_3	V_2	42.86%	23.81%	23.02%	30.16%	50.79%	58.33%	49.21%	36.91%	31.35%	38.10%	60.71%	**64.68%**
	V_1&V_3	V_4	61.66%	37.95%	43.87%	47.04%	37.55%	57.31%	45.85%	33.60%	33.20%	45.06%	72.33%	**75.49%**
	V_1&V_4	V_2	44.44%	26.98%	28.57%	28.57%	42.86%	55.16%	45.24%	29.37%	28.18%	37.70%	60.32%	**63.49%**
	V_1&V_4	V_3	52.89%	35.12%	34.30%	40.91%	46.69%	65.70%	56.20%	33.47%	28.93%	45.87%	71.07%	**73.97%**
	V_2&V_3	V_1	64.31%	36.86%	40.00%	46.67%	54.51%	69.02%	60.78%	42.75%	37.65%	51.37%	79.61%	**83.14%**
	V_2&V_3	V_4	50.59%	22.53%	34.39%	36.76%	32.02%	58.89%	50.99%	22.13%	20.95%	43.48%	70.75%	**75.10%**
	V_2&V_4	V_1	70.20%	37.65%	42.75%	47.06%	60.78%	65.88%	61.57%	36.47%	35.29%	33.73%	80.00%	**82.35%**
	V_2&V_4	V_3	49.17%	26.45%	29.34%	31.82%	50.41%	55.37%	47.93%	36.36%	23.14%	41.32%	68.60%	**71.07%**
	V_3&V_4	V_1	63.14%	40.78%	38.43%	49.02%	61.96%	67.45%	65.10%	43.14%	37.26%	47.45%	81.57%	**83.53%**
	V_3&V_4	V_2	42.46%	26.19%	19.44%	25.40%	47.62%	52.78%	44.05%	32.14%	31.75%	29.37%	59.92%	**63.49%**

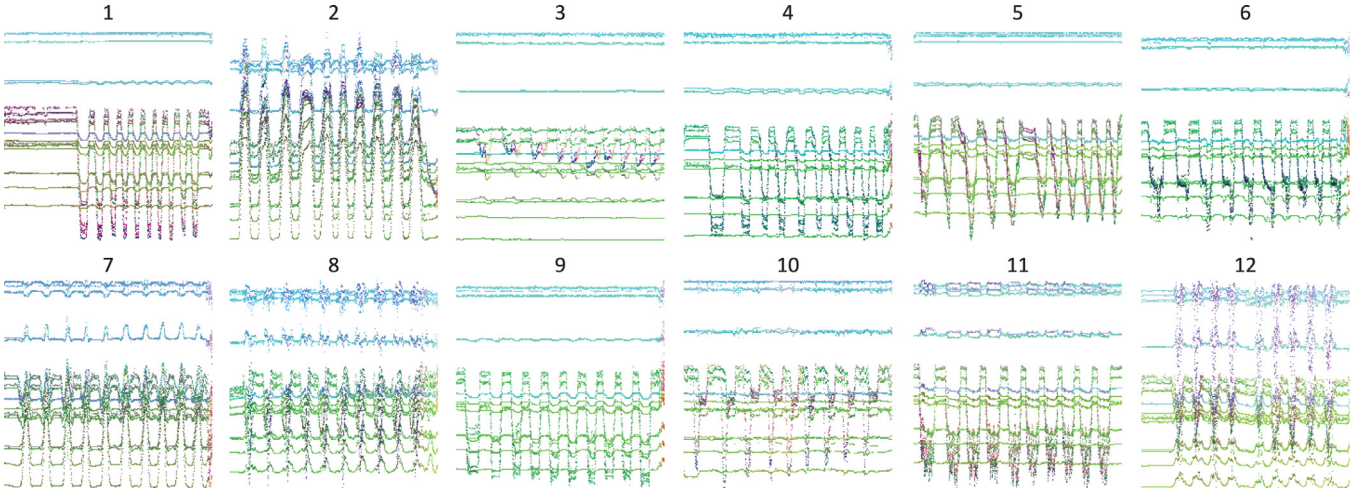

Fig. 17. The 6th type of color images generated by 12 actions of the MSRC-12 dataset [41].

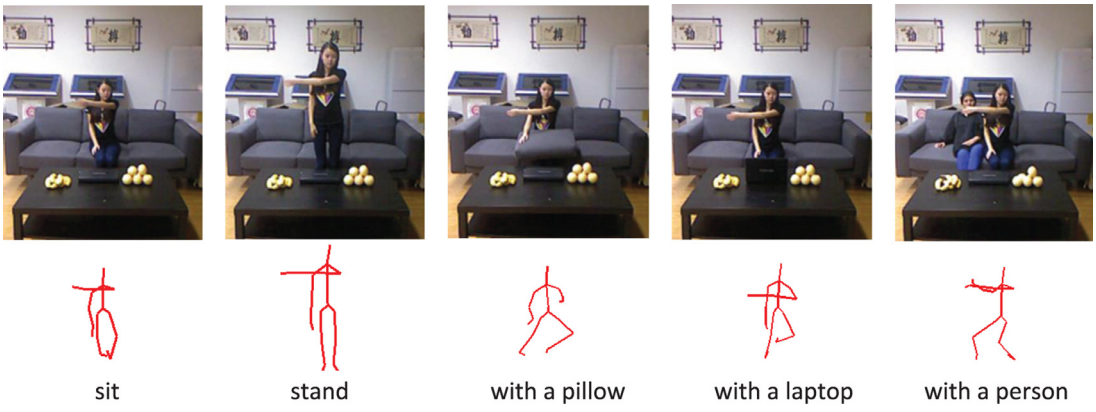

Fig. 18. Skeletons of action "*wave*" in SmartHome dataset. The estimated skeleton joints contain much noise, since the body is partial occluded by objects, e.g., desk and pillow.

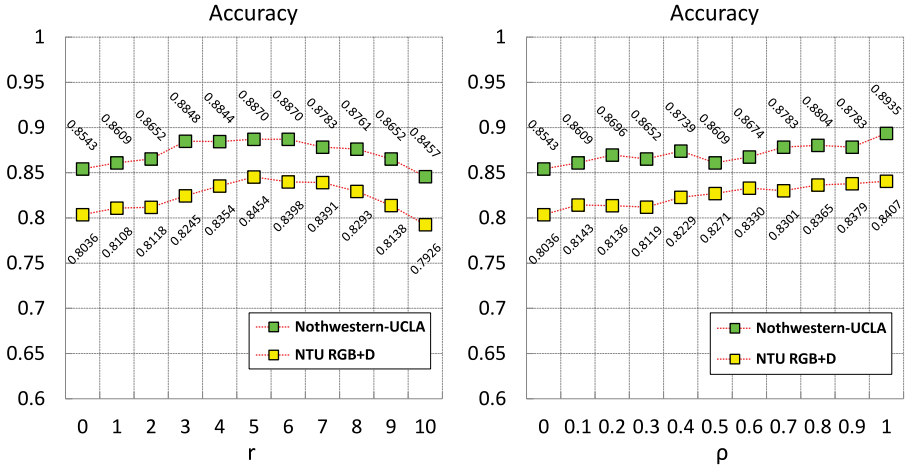

Fig. 19. Evaluation of parameter r and ρ. Left figure shows the accuracy of *Visual Enhancement* with r ranging from 0 to 10 at an interval of 1. Right figure shows the accuracy of *Motion Enhancement* with parameter ρ ranging from 0 to 1 at an interval of 0.1.

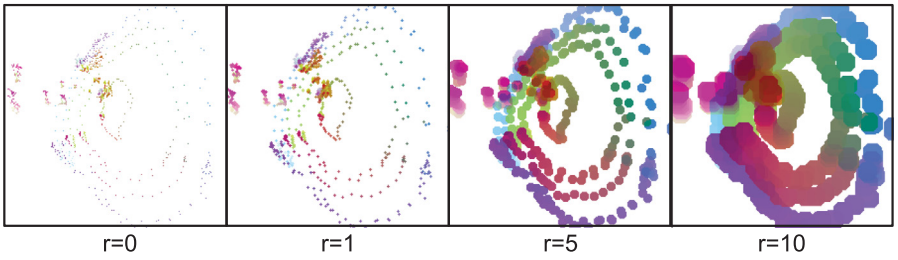

Fig. 20. The effect of parameter r on the first type of color images.

using one NVIDIA GeForce GTX 1080 card. The average testing time, including feature extraction and classification, is 0.65 s using average fusion method and is 0.39 s using weighted fusion method. Note that average fusion may need more time than weighted fusion, because in weighted fusion, some CNNs are not used and we know which CNNs to select beforehand based on tuning on training data.

7. Conclusions and future works

This paper aims at recognizing skeleton sequences under arbitrary viewpoints. First, a sequence-based transform method is designed to map skeleton joints into a view invariant high dimensional space. Second, points in the space are visualized as color images, which are compact and distinct to encode both spatial and temporal cues of original skeleton joints. Further, visual and motion enhancement methods are developed to increase the discriminative power of color images. Finally, a multi-stream CNN-based model is applied to extract and fuse deep features from enhanced color images. Extensive experiments on benchmark datasets verify the robustness of our method against view variations, noisy skeletons, inter-similarities and intra-varieties among skeleton sequences. Our method obtains nearly 10% improvement on the NTU RGB+D dataset, the largest dataset for skeleton-based recognition. This result verifies the efficiency of our method compared with the state-of-the-art LSTM-based methods. In future work, instead of using hard selection of the ten CNNs in the weighted probability fusion, i.e., $\eta_c = 0$ or 1, we can use soft probability fusion, i.e., $\eta = [0, 1]$, to provide more flexibility. In addition, we can explore other fusion method, e.g., fusing CNNs in the Softmax loss layer in the training stage. Also, we can further enhance the color images by involving temporal and spatial saliency. Data augmentation methods like adding gaussian noise to training samples can be obtained to improve the performance. Recognizing actions from untrimmed sequences is also a new field for exploring.

Acknowledgments

This work is supported by National High Level Talent Special Support Program National Natural Science Foundation of China (NSFC, Nos. 61340046, 61673030, 61672079, U1613209), Specialized Research Fund for the Doctoral Program of Higher Education (No.20130001110011), Natural Science Foundation of Guangdong Province (No.2015A030311034), Scientific Research Project of Guangdong Province (No.2015B010919004).

References

[1] C. Manresa, J. Varona, R. Mas, F.J. Perales, Hand tracking and gesture recognition for human–computer interaction, ELCVIA Electron. Lett. Comput. Vis. Image Anal. 5 (3) (2005) 96–104.
[2] R.H. Baxter, N.M. Robertson, D.M. Lane, Human behaviour recognition in data-scarce domains, Pattern Recognit. 48 (8) (2015) 2377–2393.
[3] H. Chen, G. Wang, J.-H. Xue, L. He, A novel hierarchical framework for human action recognition, Pattern Recognit. 55 (2016) 148–159.
[4] T. De Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, D. Windridge, An evaluation of bags-of-words and spatio-temporal shapes for action recognition, in: Proceedings of the Winter Conference on Applications of Computer Vision, 2011, pp. 344–351.
[5] Y. Yi, M. Lin, Human action recognition with graph-based multiple-instance learning, Pattern Recognit. 53 (2016) 148–162.
[6] C. Chen, M. Liu, B. Zhang, J. Han, J. Jiang, H. Liu, 3D action recognition using multi-temporal depth motion maps and fisher vector, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 3331–3337.
[7] A. Hernández-Vela, M.A. Bautista, X. Perez-Sala, V. Ponce, X. Baró, O. Pujol, C. Angulo, S. Escalera, BoVDW: bag-of-visual-and-depth-words for gesture recognition, in: Proceedings of the International Conference on Pattern Recognition, 2012, pp. 449–452.
[8] C. Chen, R. Jafari, N. Kehtarnavaz, A survey of depth and inertial sensor fusion for human action recognition, Multimed. Tools Appl. (2015a) 1–21.
[9] C. Chen, R. Jafari, N. Kehtarnavaz, Improving human action recognition using fusion of depth camera and inertial sensors, IEEE Trans. Human-Mach. Syst. 45 (1) (2015b) 51–61.

[10] M. Liu, H. Liu, Depth context: a new descriptor for human activity recognition by using sole depth sequences, Neurocomputing 175 (2016) 747–758.
[11] C. Chen, R. Jafari, N. Kehtarnavaz, A real-time human action recognition system using depth and inertial sensor fusion, IEEE Sens. J. 16 (3) (2016) 773–781.
[12] Z. Zhang, Microsoft kinect sensor and its effect, IEEE Multimed. 19 (2) (2012) 4–10.
[13] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Commun. ACM 56 (1) (2013) 116–124.
[14] X. Yang, Y. Tian, Eigenjoints-based action recognition using Naive–Bayes-nearest-neighbor, in: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 14–19.
[15] M.E. Hussein, M. Torki, M.A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2013, pp. 2466–2472.
[16] M. Ding, G. Fan, Multilayer joint gait-pose manifolds for human gait motion modeling, IEEE Trans. Cybern. 45 (11) (2015) 2413–2424.
[17] L. Xia, C.-C. Chen, J. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 20–27.
[18] M. Jiang, J. Kong, G. Bebis, H. Huo, Informative joints based human action recognition using skeleton contexts, Signal Process. Image Commun. 33 (2015) 29–40.
[19] M. Raptis, D. Kirovski, H. Hoppe, Real-time classification of dance gestures from skeleton animation, in: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2011, pp. 147–156.
[20] P. Wang, Z. Li, Y. Hou, W. Li, Action recognition based on joint trajectory maps using convolutional neural networks, in: Proceedings of the ACM International Conference on Multimedia, 2016, pp. 102–106.
[21] Y. Du, Y. Fu, L. Wang, Skeleton based action recognition with convolutional neural network, in: Proceedings of the Asian Conference on Pattern Recognition, 2015, pp. 579–583.
[22] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, S. Gould, Dynamic image networks for action recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2016, pp. 3034–3042.
[23] I.N. Junejo, E. Dexter, I. Laptev, P. Perez, View-independent action recognition from temporal self-similarities, IEEE Trans. Pattern Anal. Mach. Intell. 33 (1) (2011) 172–185.
[24] Y. Hsu, C. Liu, T. Chen, L. Fu, Online view-invariant human action recognition using RGB-D spatio-temporal matrix, Pattern Recognit. 60 (2016) 215–226.
[25] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Sequence of the most informative joints (SMIJ): a new representation for human skeletal action recognition, J. Vis. Commun. Image Represent. 25 (1) (2014) 24–38.
[26] X. Yang, Y. Tian, Effective 3D action recognition using eigenjoints, J. Vis. Commun. Image Represent. 25 (1) (2014) 2–11.
[27] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3D skeletons as points in a lie group, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2014, pp. 588–595.
[28] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2015, pp. 1110–1118.
[29] A. Shahroudy, J. Liu, T.T. Ng, G. Wang, NTU RGB+D: a large scale dataset for 3D human activity analysis, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2016, pp. 1010–1019.
[30] V. Veeriah, N. Zhuang, G.J. Qi, Differential recurrent neural networks for action recognition, in: Proceedings of the International Conference on Computer Vision, 2015, pp. 4041–4049.
[31] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016, pp. 3697–3704.
[32] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal LSTM with trust gates for 3D human action recognition, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 816–833.
[33] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Proceedings of the Conference and Workshop on Neural Information Processing Systems, 2014, pp. 568–576.
[34] P. Wang, W. Li, Z. Gao, C. Tang, J. Zhang, P. Ogunbona, Convnets-based action recognition from depth maps through virtual cameras and pseudocoloring, in: Proceedings of the ACM International Conference on Multimedia, 2015, pp. 1119–1122.
[35] W.J. Schroeder, B. Lorensen, K. Martin, The Visualization Toolkit, Kitware Inc., 2004.
[36] D. Borland, R.M. Taylor II, Rainbow color map (still) considered harmful., IEEE Comput. Gr. Appl. 27 (2) (2007) 14–17.
[37] J. Serra, Image Analysis and Mathematical Morphology, 1, Academic press, 1982.
[38] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Conference and Workshop on Neural Information Processing Systems, 2012, pp. 1097–1105.
[39] J. Wang, X. Nie, Y. Xia, Y. Wu, S.C. Zhu, Cross-view action modeling, learning, and recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2014, pp. 2649–2656.
[40] H. Rahmani, A. Mahmood, D.Q. Huynh, A.S. Mian, Histogram of oriented principal components for cross-view action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 38 (12) (2016) 2430–2443.
[41] S. Fothergill, H. Mentis, P. Kohli, S. Nowozin, Instructing people for training gestural interactive systems, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2012, pp. 1737–1746.
[42] A. Vedaldi, K. Lenc, MatConvNet: convolutional neural networks for MATLAB, in: Proceedings of the ACM International Conference on Multimedia, 2015, pp. 689–692.
[43] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, in: Proceedings of the British Machine Vision Conference, 2014, pp. 1–12.
[44] O. Oreifej, Z. Liu, HON4D: histogram of oriented 4D normals for activity recognition from depth sequences, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2013, pp. 716–723.
[45] X. Yang, Y.L. Tian, Super normal vector for human activity recognition with depth cameras, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2016), doi:10.1109/TPAMI.2016.2565479.
[46] H. Rahmani, A. Mian, 3D action recognition from novel viewpoints, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2016, pp. 1506–1515.
[47] J. Wang, Z. Liu, Y. Wu, J. Yuan, Learning actionlet ensemble for 3D human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 36 (5) (2014) 914–927.
[48] E. Ohn-Bar, M. Trivedi, Joint angles similarities and HOG^2 for action recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 465–470.
[49] G. Evangelidis, G. Singh, R. Horaud, Skeletal quads: human action recognition using joint quadruples, in: Proceedings of the International Conference on Pattern Recognition, 2014, pp. 4513–4518.
[50] J.-F. Hu, W.-S. Zheng, J. Lai, J. Zhang, Jointly learning heterogeneous features for RGB-D activity recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2015, pp. 5344–5352.
[51] L. Zhou, W. Li, Y. Zhang, P. Ogunbona, D.T. Nguyen, H. Zhang, Discriminative key pose extraction using extended LC-KSVD for action recognition, in: Proceedings of the Proceedings of Digital Image Computing: Technqiues and Applications, 2014, pp. 1–8.

Mengyuan Liu received the B.E. degree in intelligence science and technology in 2012, and is working toward the Ph.D. degree in the School of EE&CS, Peking University (PKU), China. His research interests include action recognition and action detection. He has already published articles in MTA, Neurocomputing, ROBIO2013, ICIP2014, ICASSP2014, ICIP2015, ROBIO2016, 3DV2016, IJCAI2016, ICASSP2017. https://scholar.google.com/citations?user=woX_4AcAAAAJ&hl=zh-CN.

Hong Liu received the Ph.D. degree in mechanical electronics and automation in 1996, and serves as a Full Professor in the School of EE&CS, Peking University (PKU), China. Prof. Liu has been selected as Chinese Innovation Leading Talent supported by "National High-level Talents Special Support Plan" since 2013.

He is also the Director of Open Lab on Human Robot Interaction, PKU, his research fields include computer vision and robotics, image processing, and pattern recognition. Liu has published more than 150 papers and gained Chinese National Aero-space Award, Wu Wenjun Award on Artificial Intelligence, Excellence Teaching Award, and Candidates of Top Ten Outstanding Professors in PKU. He is an IEEE member, vice president of Chinese Association for Artificial Intelligent (CAAI), and vice chair of Intelligent Robotics Society of CAAI. He has served as keynote speakers, co-chairs, session chairs, or PC members of many important international conferences, such as IEEE/RSJ IROS, IEEE ROBIO, IEEE SMC and IIHMSP, recently also serves as reviewers for many international journals such as Pattern Recognition, IEEE Trans. on Signal Processing, and IEEE Trans. on PAMI. https://scholar.google.com/citations?user=4CQKG8oAAAAJ&hl=zh-CN.

Chen Chen received the B.E. degree in automation from Beijing Forestry University, Beijing, China, in 2009 and the M.S. degree in electrical engineering from Mississippi State University, Starkville, in 2012 and the Ph.D. degree in the Department of Electrical Engineering at the University of Texas at Dallas, Richardson, TX in 2016. He is a Post-Doc in the Center for Research in Computer Vision at University of Central Florida (UCF). His research interests include signal and image processing, pattern recognition and computer vision. http://www.utdallas.edu/~cxc123730/.