IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ADVANCED DATA ANALYTICS FOR LARGE-SCALE COMPLEX DATA ENVIRONMENTS

Received September 5, 2017, accepted September 28, 2017, date of publication October 2, 2017, date of current version November 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2759058

Multi-Temporal Depth Motion Maps-Based Local Binary Patterns for 3-D Human Action Recognition

CHEN CHEN[1], (Member, IEEE), MENGYUAN LIU[2], (Member, IEEE), HONG LIU[2], (Member, IEEE), BAOCHANG ZHANG[3], (Member, IEEE), JUNGONG HAN[4], (Member, IEEE), AND NASSER KEHTARNAVAZ[5], (Fellow, IEEE)

[1]Center for Research in Computer Vision, University of Central Florida, Orlando, FL 32816 USA
[2]Shenzhen Graduate School, Peking University, Beijing 100871, China
[3]Beihang University, Beijing 100080, China
[4]School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, U.K.
[5]Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA

Corresponding author: Hong Liu (hongliu@pku.edu.cn)

This work was supported by the Natural Science Foundation of China under Grant 61672079, Grant 61473086, Grant U1613209, and Grant 6167021685.

ABSTRACT This paper presents a local spatio-temporal descriptor for action recognistion from depth video sequences, which is capable of distinguishing similar actions as well as coping with different speeds of actions. This descriptor is based on three processing stages. In the first stage, the shape and motion cues are captured from a weighted depth sequence by temporally overlapped depth segments, leading to three improved depth motion maps (DMMs) compared with the previously introduced DMMs. In the second stage, the improved DMMs are partitioned into dense patches, from which the local binary patterns histogram features are extracted to characterize local rotation invariant texture information. In the final stage, a Fisher kernel is used for generating a compact feature representation, which is then combined with a kernel-based extreme learning machine classifier. The developed solution is applied to five public domain data sets and is extensively evaluated. The results obtained demonstrate the effectiveness of this solution as compared with the existing approaches.

INDEX TERMS Action recognition, depth motion maps, ELM classifier, local binary patterns, fisher kernel.

I. INTRODUCTION

Action recognition plays a significant role in a number of computer vision applications such as context-based video retrieval, human-computer interaction and intelligent surveillance systems [1]–[5]. Many previous works have focused on recognizing actions captured by conventional RGB video cameras, e.g., [6], [7]. However, these works have limitations such as coping with various lighting conditions and cluttered backgrounds due to the fact that RGB data suffer from these variations, thus impeding effectiveness in real-world applications, such as video surveillance.

More recent advances have involved performing action recognition by exploiting depth cameras. Compared with RGB cameras, depth cameras have several advantages: 1) depth data are much more tolerant to changes in lighting conditions and depth cameras even work in dark environments; 2) color and texture do not appear in depth images, which makes the tasks of human detection and foreground extraction from cluttered backgrounds easier [8]; 3) depth cameras provide depth images, which capture the 3D structural data of subjects/objects in the scene [9]; 4) human skeleton positions (e.g., 3D joints positions and rotation angles) can be efficiently obtained from depth images providing additional information for performing action recognition [10].

Since the release of cost-effective depth cameras, such as Microsoft Kinect, many works on action recognition have been conducted using depth images. Various representations of depth sequences have been explored, including bag of 3D points [11], spatio-temporal depth cuboid [12], depth motion maps (DMMs) [13]–[16], surface normals [8], [17] and skeleton joints [18]. Among these representation schemes, DMMs-based representations transform the action recognition problem from 3D to 2D and have been successfully applied to depth-based action recognition. Specifically, DMMs are obtained by projecting the depth frames onto three orthogonal Cartesian planes and accumulating the difference

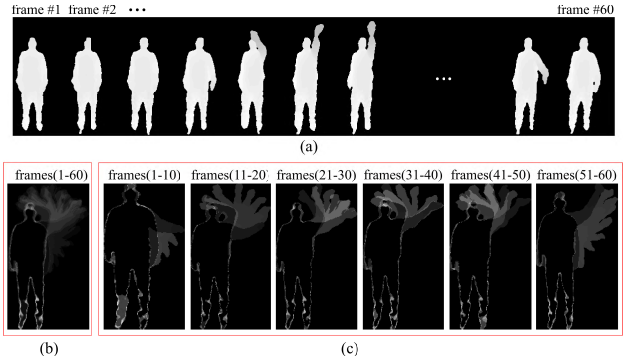


FIGURE 1. **An example of different levels of DMMs representations. (a) is an example depth action sequence "high wave." (b) shows the front view DMM generated using the entire sequence. (c) presents 6 DMMs generated using 6 different subsets of depth frames (e.g., frames 1-10, 11-20, 21-30, etc.).**

between projected maps over an entire sequence. They are basically used to describe the shape and motion cues of a depth action sequence.

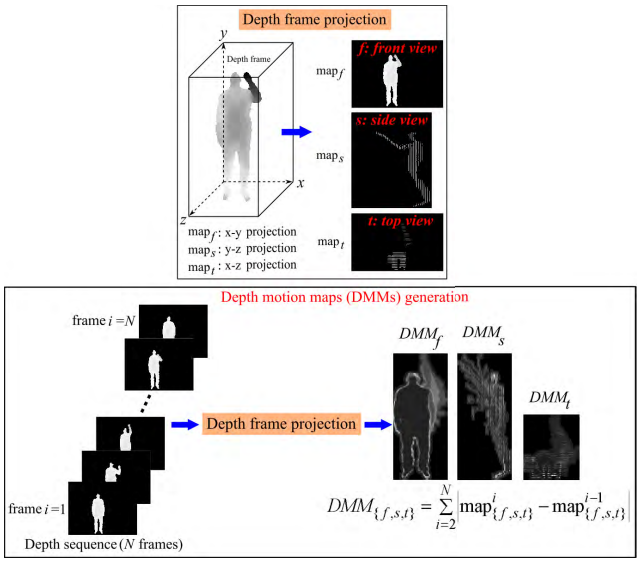


FIGURE 2. **DMMs generation for a depth sequence.**

However, DMMs of an entire depth sequence may not be able to capture detailed temporal motion in a subset of depth images. As a result, previous motion history may get overwritten when a more recent action occurs at the same point. An example of this phenomena is shown in Fig. 1 where the limitation of DMMs in capturing detailed motion cues is illustrated. In this figure, (a) is an example depth action sequence *high wave*, (b) shows the DMM of the front view projection generated using all the depth frames (60 frames) in the action sequence, and (c) shows 6 DMMs of the front view projection generated using 6 different subsets of depth frames (e.g., frames 1-10, 11-20, 21-30, etc.) in the same action sequence. It can be observed that the detailed motion (e.g., raising hand over head and waving) of a hand waving action can be observed in DMMs generated using subsets of depth frames in a depth action sequence. In other words, the waving motion exhibited in the DMMs generated from subsets of depth frames is more obvious and clear than that in the DMMs generated using the entire depth sequence (all frames). In addition, action speed variations may result in large intra-class variations in DMMs. To overcome the above shortcomings of DMM representations, a new local spatio-temporal descriptor is developed by taking into consideration the shape discrimination and action speed variations. The contributions of this work are as follows:

- The original DMM representation is improved by nonlinearly accumulating weighted motion regions of a sequence onto three orthogonal Cartesian planes. By doing so, the temporal relationships among frames are retained. In addition, to cope with speed variations in actions, different temporal lengths of depth segments are employed, eventually leading to a multi-temporal DMM representation.
- A set of local patch descriptors are built by partitioning all the DMMs into dense patches and utilize the local binary patterns (LBP) [19] to characterize local rotation invariant texture information in those patches.
- To make the representation more compact, the Fisher kernel [20] is used to encode the patch descriptors, which is fed into a kernel-based extreme learning machine (ELM) classifier [21] for recognition.

A preliminary version of the above approach appeared in [22]. This paper extends that work in the following manner. First, a more comprehensive survey on related works is provided. Second, an improved set of DMMs is proposed based on a nonlinear weighting function to assign different weights to depth frames, thereby preserving the temporal information among different frames. Third, the method developed in this paper has been evaluated on four benchmark datasets and a comprehensive comparison is provided with the state-of-the-art approaches including deep learning methods, e.g., [23]. The experimental results show that our method outperforms these existing methods. Fourth, an MSRAction 3D-Speed dataset has been put together based on the original MSRAction3D dataset [11], which is used to demonstrate the robustness of our method to speed variations. It is worth noting that our method is flexible in the sense that it can be combined with skeleton joints employing a similar multi-temporal structure.

The remainder of this paper is organized as follows. Section 2 briefly reviews related works. Section 3 provides the details of our proposed depth video representation method. The experimental results on several benchmark datasets are reported in Section 4. Finally, Section 5 concludes the paper.

II. RELATED WORK

As pointed out in [24], 3D action recognition methods can be categorized into three types: depth-based, skeleton-based, and depth-skeleton-based methods. This section reviews these three types of methods briefly.

Surface normal vectors [25] can reflect local structure of 3D objects, therefore they are widely used for 3D object retrieval. An extended version called Histogram of Oriented 4D Normal vectors (HON4D) [17] is developed to capture local structure of spatio-temporal depth data. The 4D space denotes time, depth and 2D spatial coordinates. Since HON4D captures local information of one point separately, Histogram of Oriented Principal Components (HOPC) [26] is proposed to capture relationships surrounding that point. Specifically, HON4D calculates principal directions within a volume around the point and then encodes three main principal directions. Compare with HON4D, HOPC is more robust to depth noise, due to the usage of principal directions. Moreover, HOPC is able to capture relationships among local points. However, HOPC ignores relationships among points in a large scale. To solve this problem, a depth sequence is treated as many pairwise 3D points, and the relative depth relationships are used to construct binary descriptors [27]. This representation captures both local and global relationships. An alternative solution appears in [28], where a depth sequence is split into spatio-temporal cells, and then a locality-constrained linear coding is applied to encode features extracted from these cells. Despite of above methods, many types of descriptors are developed. For example, 2D and 3D auto-correlation of gradients features are combined using a weighted fusion framework for action recognition [29]. In [30], a tensor subspace, whose dimension is learned automatically by low-rank learning, is developed for RGB-D action recognition. Recent works on depth-based 3D action recognition focus more on specific problems such as cross-view action recognition. In [31], a depth video based cross-view action recognition method is developed, which learns a general view-invariant human pose model from synthetic depth images using a convolutional neural network and a sparse Fourier Temporal Pyramid to encode action specific features for spatio-temporal representation.

Recognizing human actions by observing human body joints is an intuitive way. A real-time skeleton joints estimation method is proposed [10], which opens a new way for action representation. The first attempt treats 3D action as a cloud of skeleton joints, which are placed into 3D spatial bins. The spatial distribution of these joints are encoded by histograms of 3D joint locations (HOJ3D). Since the 3D spatial coordinates are attached to skeletons, HOJ3D feature shows robustness to view point changes to some extent. To encode both spatial and temporal information of joints, Yang *et al.* apply Principal Component Analysis (PCA) on joint differences, generating the EigenJoints to reflect differences of joints in temporal and spatial domains. Observing that the joint differences only reflect speed of joints, Zanfir *et al.* [32] propose to use position, speed and acceleration of joints as features, and develop a Moving Pose (MP) framework for action recognition. Kerola *et al.* [33] construct a spatial temporal graph by linking joints in consecutive skeletons, where edge weights are calculated by distances. A spectral graph wavelet transform (SGWT) is applied on the 3D skeleton graph to create an overcomplete representation. In [34], Cai *et al.* develop a novel action attribute mining method, where an attribute space is constructed by the geometry transformation between body parts. In [18], the 3D geometric relationships among human body parts are explicitly modeled as curves using a Lie group. With the progress of deep learning [35], recent works use a single image to encode spatio-temporal information of skeleton joints, and then fine-tune pre-defined models for transfer learning. In [23], the skeleton is divided into five parts, which are used as inputs for five bidirectional recurrent neural networks (BRNNs). Then, the representations from the subnets are fused in a hierarchical way to be the inputs to higher layers. Since recurrent neural network (RNN) can model the long-term contextual information of temporal sequences, the proposed end-to-end hierarchical RNN achieves high performances on the task of skeleton-based action recognition. In [36], a skeleton sequence is visualized as several color images, which explicitly encode both spatial distributions and temporal evolutions of skeleton joints. To enhance the discriminative power of color images, skeleton joints with salient motions are emphasized when generating these color images. Finally, enhanced color images are used as inputs for a multi-stream convolutional neural networks, which explores the complementary properties between color images. Although combining deep learning methods and some hand-crafted features can obtain high recognition performance, skeleton-based methods are not applicable for applications where skeleton information is not available.

When skeleton joints can be stably estimated, these joints reflect robust motion patterns of human actions, thereby avoiding the effect of noisy depth data. However, in human-object interaction scenarios, skeleton joints can barely capture any information about the object. Moreover, when the human body is not directly facing the depth sensor, the estimated skeleton joints are usually noisy. In these cases, original depth data provides essential cues for distinguishing similar actions. In the direction of depth and skeleton information fusion, an ensemble model [37] is proposed to associate local occupancy pattern features from depth images with skeleton joints. In this way, the object can be reflected by describing depth data surrounding skeleton joints. Moreover, traditional HOG feature is also used to describe depth data surrounding skeleton joints [38]. Although multi-modal fusion methods generally achieve higher recognition accuracy, having a depth descriptor on top of a complicated skeleton tracker makes such algorithms computationally expensive, limiting their use in real-time applications.

III. PROPOSED DEPTH VIDEO REPRESENTATION

A. IMPROVED DEPTH MOTION MAPS

According to [14], the DMMs of a depth sequence with N frames are computed as follows:

$$DMM_{\{f,s,t\}} = \sum_{i=2}^{N} |map^{i}_{\{f,s,t\}} - map^{i-1}_{\{f,s,t\}}|, \qquad (1)$$

where map_f^i, map_s^i and map_t^i indicate three projected maps of the i^{th} depth frame on three orthogonal Cartesian planes corresponding to the front view (f), side view (s) and top view (t). A graphical illustration of DMMs generation is presented in Fig. 2.

On each orthogonal Cartesian plane, the DMMs are formulated by accumulating projected maps through an entire sequence. In this case, the temporal relationships among frames are not taken into consideration. To accommodate for the temporal relationships, an improved version of DMMs is provided here.

In order to describe the motion in RGB video, in [39] Motion Energy Image (MEI) was considered, which contains the motion information through accumulating binary processed image frames. To preserve the temporal information between different frames, in [39] a linear weighting function was considered with the time as an independent variable. After assigning each frame a weight, a Motion History Image (MHI) was generated, where more recently moving pixels are brighter. An analogy to this approach is adopted here between DMMs and MEI, and an improved DMMs is thus developed that follows a similar design flow of MHI. Different from the linear weighting function in [39], a nonlinear weighting function is designed which facilitates the weighting scheme. By adjusting one parameter, one would be able to obtain a set of weighting functions, which incorporates the linear weighting function.

Specifically, our weight in the weighting function is defined as follows:

$$weight(i) = \frac{1 - e^{-wi}}{1 - e^{-wN}}, \tag{2}$$

where i indicates the i^{th} frame in a sequence with N total frames and the parameter w controls the shape of the weighting function. Accordingly, the improved DMMs is defined as:

$$DMM_{\{f,s,t\}} = \sum_{i=2}^{N} |map^i_{\{f,s,t\}} - map^{i-1}_{\{f,s,t\}}| * weight(i), \tag{3}$$

which accumulates weighted motion regions through an entire sequences on three orthogonal Cartesian planes. The effect of parameter w on the weighting function is shown in Fig. 3. When $w \to \infty$, the same weight is assigned to each frame, and the improved DMMs becomes the original DMMs. When $w \to 0$, the weighting function becomes a linear function, which is similar to the weighting scheme in [39].

B. MULTI-TEMPORAL DEPTH MOTION MAPS

As aforementioned, DMMs based on an entire depth sequence may not be able to capture detailed motion cues. To capture more motion information, a depth sequence is divided into a set of overlapped 3D depth segments with equal number of frames (i.e., same frame length for each depth segment) and three DMMs are computed for each depth segment. Since different people may perform an action at different speeds, multiple frame lengths are also used to represent multiple temporal resolutions to cope with speed variations. The proposed multi-temporal DMMs representation framework is shown in Fig. 4. This figure illustrates an example where DMMs are generated by using the entire depth sequence (i.e., all the frames in the sequence) is considered to be the default level of the temporal resolution (denoted by Level 0 in Fig. 4). In the second level (Level 1 in Fig. 4), the frame length (L_1) of a depth segment is set to 5 (i.e., 5 frames in a depth segment). In the third level (Level 2 in Fig. 4), the frame length (L_2) of a depth segment is set to 10. Note that L_1 and L_2 can be changed. Obviously, the computational complexity increases by increasing temporal levels. Thus, the maximum number of levels is limited here to 3 including the default level, i.e., Level 0, which considers all the frames. The frame interval (R, $R < L_1$ and $R < L_2$) in Fig. 4 is the number of frames between the first frames (or the starting frames), respectively, in two neighboring depth segments, indicating the amount of overlap between the two segments. For simplicity, the same R in Level 1 and Level 2 is used here.

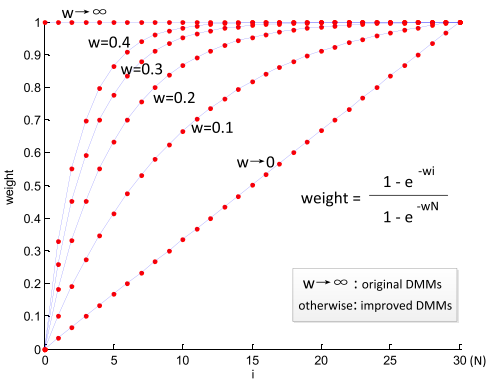


FIGURE 3. A weighting function for constructing improved DMMs. An action sequence with $N = 30$ frames is illustrated for example.

C. PATCH-BASED LBP FEATURES

DMMs can effectively capture the shape and motion cues of a depth sequence. However, DMMs are pixel-level features. To enhance the discriminative power of DMMs, the patch-based LBP feature extraction approach in [15] is adopted here to characterize the rich texture information (e.g., edges, contours, etc.) in the LBP encoded DMMs.

The LBP operator [19] is a simple yet effective gray scale and rotation invariant texture operator that has been used in various applications. It labels pixels in an image with decimal numbers that encode local texture information. Given a pixel (scalar value) g_c in an image, its neighbor set contains pixels that are equally spaced on a circle of radius r ($r > 0$) with the center at g_c. If the coordinates of g_c are $(0, 0)$ and m neighbors $\{g_i\}_{i=0}^{m-1}$ are considered, the coordinates of g_i are $(-rsin(2\pi i/m), rcos(2\pi i/m))$. The gray values of circular neighbors that do not fall in the image grids are estimated by bilinear interpolation [19]. Fig. 5 illustrates an example of a

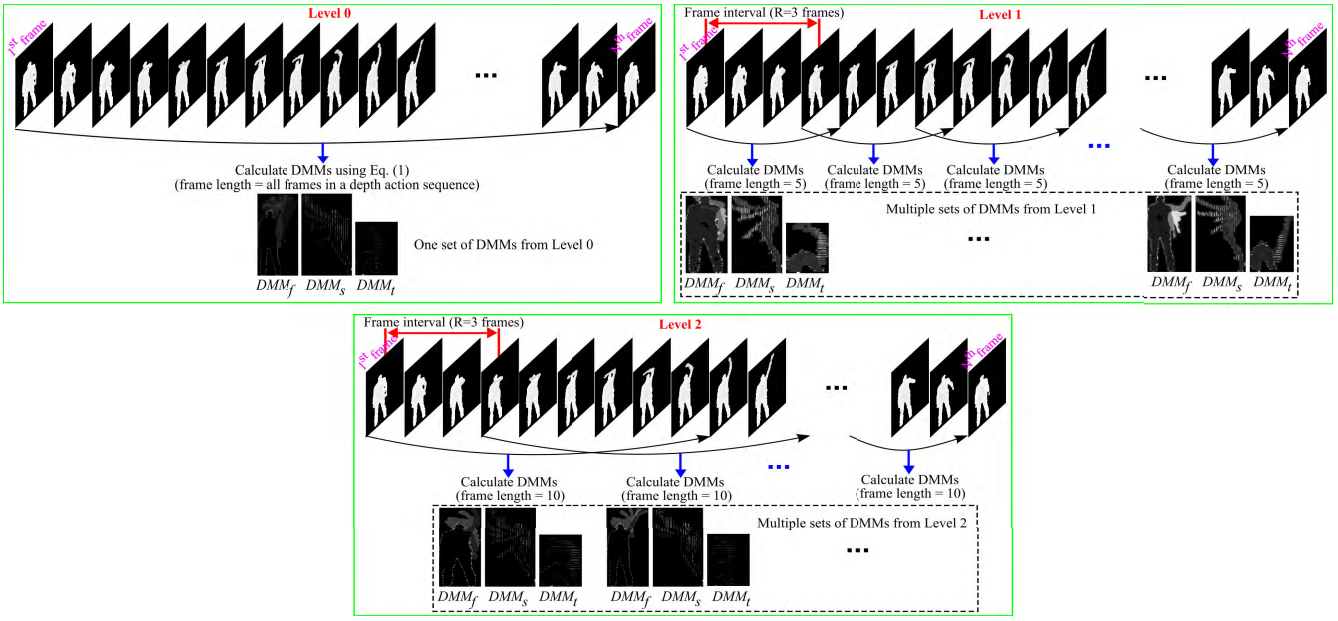


FIGURE 4. Proposed multi-temporal DMMs representation of a depth sequence.

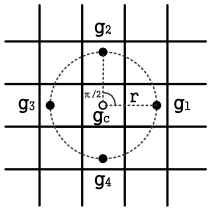


FIGURE 5. Center pixel g_c and its 4 circular neighbors $\{g_i\}_{i=0}^{3}$ with radius r for the LBP operator.

neighbor set for ($m = 4$, $r = 1$) (the values for m and r may change in practice). The LBP is created by thresholding the neighbors $\{g_i\}_{i=0}^{m-1}$ with the center pixel g_c to generate a m-bit binary number. The resulting LBP for g_c can be expressed in decimal form as follows:

$$LBP_{m,r}(g_c) = \sum_{i=0}^{m-1} U(g_i - g_c)2^i, \tag{4}$$

where $U(g_i - g_c) = 1$ if $g_i \geq g_c$ and $U(g_i - g_c) = 0$ if $g_i < g_c$. Although the LBP operator in Eq. (4) produces 2^m different binary patterns, a subset of these patterns, named uniform patterns, is thus able to describe image texture [19]. After obtaining the LBP codes for pixels in an image, an occurrence histogram is computed over an image or a region to represent the texture information.

Fig. 6 shows the process of patch-based LBP feature extraction. The overlap between two patches is controlled by the pixel shift (*ps*) illustrated in Fig. 6. Under each projection view, a set of patch-based LBP histogram features are generated to describe the corresponding multi-temporal DMMs. Therefore, three feature matrices $\mathbf{H}_f$, $\mathbf{H}_s$ and $\mathbf{H}_t$ are

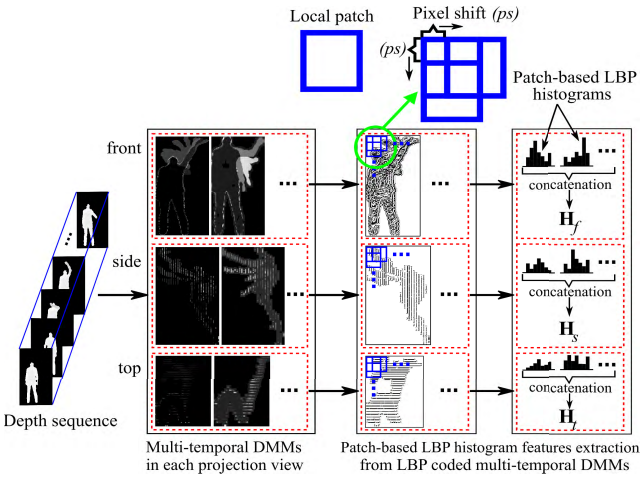


FIGURE 6. Patch-based LBP feature extraction.

generated which are associated with front view DMMs, side view DMMs and top view DMMs, respectively. Each column of the feature matrix (e.g., $\mathbf{H}_f$) denotes a histogram feature vector of a local patch.

D. A FISHER KERNEL REPRESENTATION

Fisher kernel representation [20] is an effective patch aggregation mechanism to characterize a set of low-level features, which shows superior performance over the popular Bag-of-Visual-Words (BoVW) model. Therefore, the Fisher kernel is employed here to build a compact and descriptive representation of the patch-based LBP features.

Let $\mathbf{H} = \{\boldsymbol{h}_i \in \mathbb{R}^D, 1 \leq i \leq M\}$ be a set of M D-dimensional patch-based LBP feature vectors extracted from the multi-temporal DMMs of a particular projection

view (e.g., front veiw) for a depth sequence. By assuming statistical independence, $\mathbf{H}$ can be modeled by a K-component Gaussian mixture model (GMM):

$$p(\mathbf{H}|\theta) = \prod_{i=1}^{M} \sum_{k=1}^{K} \omega_k \mathcal{N}(\boldsymbol{h}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{5}$$

where $\theta = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}, k = 1, \ldots, K$ is the parameter set with mixing parameters ω_k, means $\boldsymbol{\mu}_k$ and diagonal covariance matrices $\boldsymbol{\Sigma}_k$ with the variance vector $\boldsymbol{\sigma}_k^2$. These GMM parameters can be estimated by using the Expectation-Maximization (EM) algorithm based on a training dataset (or feature set).

Two D-dimensional gradients with respect to the mean vector $\boldsymbol{\mu}_k$ and standard deviation $\boldsymbol{\sigma}_k$ of the k^{th} Gaussian component are defined as

$$\begin{aligned} \boldsymbol{\rho}_k &= \frac{1}{M\sqrt{\pi_k}} \sum_{i=1}^{M} \gamma_{k,i} \frac{\boldsymbol{h}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}, \\ \boldsymbol{\tau}_k &= \frac{1}{M\sqrt{2\pi_k}} \sum_{i=1}^{M} \gamma_{k,i} \left(\left(\frac{\boldsymbol{h}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right)^2 - 1 \right), \end{aligned} \tag{6}$$

where $\gamma_{k,i}$ is the posterior probability that $\boldsymbol{q}_i$ belongs to the kth Gaussian component. The Fisher vector (FV) of $\mathbf{H}$ is represented as $\Phi(\mathbf{H}) = (\boldsymbol{\rho}_1^T, \boldsymbol{\tau}_1^T, \ldots, \boldsymbol{\rho}_K^T, \boldsymbol{\tau}_K^T)^T$, where Φ denotes the FV encoding operator. The dimensionality of the FV is $2KD$.

A power-normalization [20], i.e., signed square rooting (SSR) and ℓ_2 normalization, is applied to eliminate the sparseness of the FV as follows:

$$sgn(\Phi(\mathbf{H}))|\Phi(\mathbf{H})|^{\alpha}, \quad 0 < \alpha \leq 1. \tag{7}$$

The normalized FV is then denoted by $\mathbf{f}$.

Given N_T training action sequences with N_T feature matrices from a projection view $v \in \{f, s, t\}$, $\{\mathbf{H}_v^{[1]}, \mathbf{H}_v^{[2]}, \ldots, \mathbf{H}_v^{[N_T]}\}$ representing patch-based LBP descriptors from multi-temporal DMMs are obtained using the feature extraction method demonstrated in Fig. 6. For each projection view v, the corresponding feature matrices of the training data are used to estimate the GMM parameters via the EM algorithm. Therefore, for three projection views, three GMMs are created. After estimating the GMM parameters, three FVs ($\mathbf{f}_f$, $\mathbf{f}_s$ and $\mathbf{f}_t$) are generated for a depth sequence. Then, the three FVs are simply concatenated as the final feature representation $\mathbf{f}_{con} = [\mathbf{f}_f; \mathbf{f}_s; \mathbf{f}_t]$. Fig. 7 shows the steps toward generating FVs.

Extreme learning machine (ELM) was developed for single-hidden-layer feed-forward neural networks (SLFNs) [21]. Unlike traditional feed-forward neural networks that require all the parameters to be tuned, the hidden node parameters in ELM are randomly generated leading to a much faster learning rate. Compared with ELM, KELM provides a better generalization performance and is more stable. Therefore, this paper uses KELM for classification.

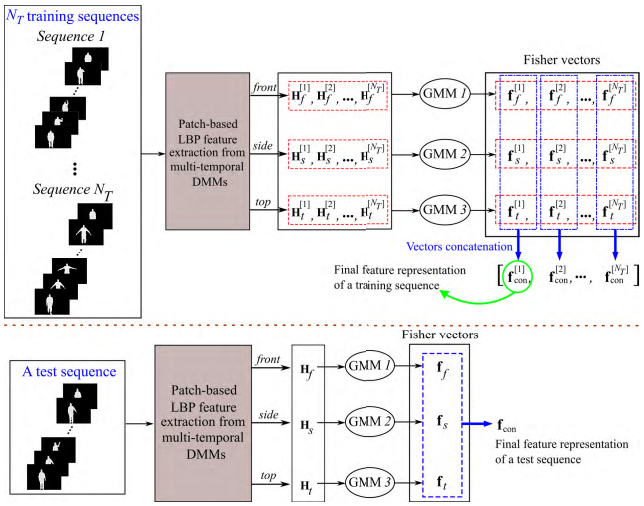


FIGURE 7. FV representation.

IV. EXPERIMENTS

To evaluate our approach, we report the outcome of our method on five datasets. The comparison between our method and related works are used to demonstrate the effect of multi-temporal DMM and LBP descriptor. In Section *F*, we report the time cost of our method and show the effect of levels on our multi-temporal structure. In Section *G*, we select proper parameter for the improved DMM and compare its performance with DMM.

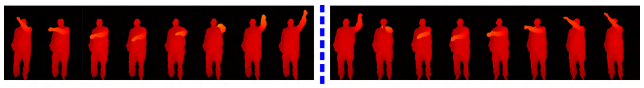


FIGURE 8. Actions "drawTick" (left) and "drawX" (right) in the *MSRAction3D* dataset.

A. MSRAction3D DATASET

1) DATASET

MSRAction3D dataset [11] is one of the most popular depth datasets for action recognition as reported in the literature. It contains 20 actions: "high arm wave", "horizontal arm wave", "hammer", "hand catch", "forward punch", "high throw", "draw x", "draw tick", "draw circle", "hand clap", "two hand wave", "sideboxing", "bend", "forward kick", "side kick", "jogging", "tennis swing", "tennis serve", "golf swing", "pick up & throw". Each action is performed 2 or 3 times by 10 subjects facing the depth camera. It is a challenging dataset due to similarity of actions and large speed variations in actions. As shown in Fig. 8, actions such as "drawX" and "drawTick" are similar except for a slight difference in the movement of one hand. We have calculated the statistics for the *MSRAction3D* dataset, which contains the actions executed by different subjects with different execution rates. To be more precise, the standard derivation of the sequence lengths (numbers of frames) across the actions is 9.21 frames (max: 13.30 frames; min: 4.86 frames), which means that execution rate difference is actually quite large.

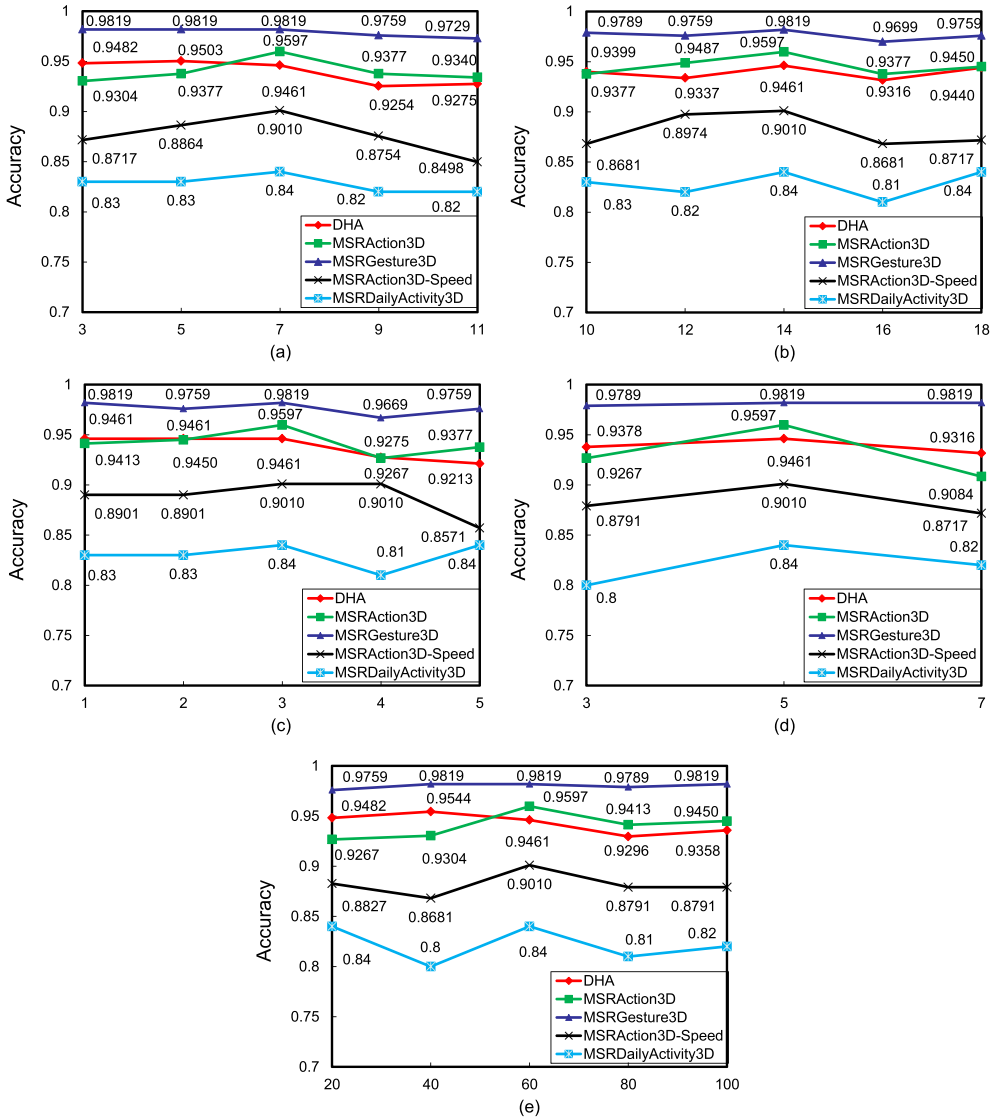


FIGURE 9. Recognition accuracies with changing parameters. (a) Frame length L_1. (b) Frame length L_2. (c) R. (d) Pixel shift (ps). (e) Number of Gussians (K)

2) SETTINGS

Following [11], the cross subject validation method was adopted here with subjects #1; 3; 5; 7; 9 for training and subjects #2; 4; 6; 8; 10 for testing. The kernel-based extreme learning machine (KELM) [21] was employed with a radial basis function (RBF) kernel as the classifier due to its general good classification performance and efficient computation. In all the experiments, the parameters for KELM (RBF kernel parameters) were chosen as the ones that maximized the training accuracy by means of a 5-fold cross-validation test.

For our feature extraction, the DMMs of different action sequences were resized to have the same size for the purpose of reducing the intra-class variation. To have fixed sizes for DMM_f, DMM_s and DMM_t, the sizes of these maps for all the action samples in the dataset were found. Following our previous work in [15], the fixed size of each DMM was set to 1/2 of the mean value of all of the sizes. This made the sizes of DMM_f, DMM_s and DMM_t to be 102×54, 102×75 and 75×54, respectively. The block sizes of the DMMs were considered to be 25×27, 25×25 and 25×27 corresponding to DMM_f, DMM_s and DMM_t. The overlap between two blocks was taken to be one half of the block size. This resulted in 21 blocks for DMM_f, 35 blocks for DMM_s and 15 blocks for DMM_t.

The same parameter values in [15] were used in our experimentations for the patch sizes and parameters for the LBP operator. The other parameters were determined empirically. The overall accuracies on three datasets with different parameters are shown in Figure 9, where frame length L_1, frame length L_2, frame interval R, pixel shift ps and the number of Gaussians (K) respectively change from 3 to 11, 10 to 18, 1 to 5, 3 to 7 and 20 to 100 at equal intervals. Experiments were conducted with one parameter changes and the other

parameters were kept to the default values: $L_1 = 7, L_2 = 14$, $R = 3$, $ps = 5$ and $K = 60$.

To build multi-temporal depth motion maps, a depth sequence was divided into a set of overlapping 3D depth segments with equal number of frames. A three level structure of multi-temporal depth motion maps was considered, which needed three parameters, i.e. L_1, L_2 and R. The frame lengths of depth segments in the second and the third levels are respectively denoted by L_1 and L_2. These parameters were determined by grid search from the grid [3, 5, 7, 9, 11]. Fig. 9 (a) and 9 (b) show the results under different values of L_1 and L_2. The overall trend is that the accuracy rises and then falls with the increase of L_1 or L_2. Small values of L_1 and L_2 limited the depth information encoded in the depth motion maps. While large values reduced the discriminative power of the 3D depth segments among different levels.

The frame interval R is the number of frames between the first frames respectively in two neighboring depth segments. Fig. 9 (c) shows the results with different values of R. Small value of R led to a dense sampling of 3D depth segments at the expense of a large amount of processing time. When the value of R was larger than the frame lengths of depth segments, many depth frames between neighboring depth segments were made discarded.

To describe DMMs by patch-based LBP features, the overlap between two patches is controlled by the pixel shift ps. As shown in Fig. 9 (d), the performance was boosted when the number of ps was increased. The optimum value was found to be 5 for the number of ps. When ps was larger than this optimum value, the performances dropped. The reason is that the spatial relationships among patches become weak when the overlap rate becomes small.

The parameter K is used in our Fisher kernel representation. The value of K was set by grid searching from the grid [20, 40, 60, 80, 100]. As shown in Fig. 9 (e), the best outcome on the four datasets with different values of K was obtained.

Generally speaking, more than 90% accuracies with different parameters on three benchmark datasets was obtained, indicating the robustness of our method to parameter settings. The *MSRAction3D-Speed* dataset was more challenging than *MSRAction3D* dataset, since the execution rate difference was higher in our dataset by sampling a portion of frames from original sequences. Even so, more than 85% accuracy was achieved with different parameters on the *MSRAction3D-Speed* dataset, again indicating the robustness of our method to speed changes. Since the default values of parameter L_1, L_2, R, ps worked well for all the four datasets, the following experiments were conducted with these values as the default values. It was observed that proper values of K were needed for different datasets to achieve the best performance. In what is reported next, the value of K was set to 60 for all the datasets.

3) COMPARISON WITH RELATED WORKS

We compared our method with the state-of-the-art methods in Table 1. ''Moving Pose'' [32], ''Skeletons in a Lie group'' [18] and ''Skeletal Quads'' [42] belong to skeleton-based features, ''Range-Sample'' [27] and ''Super Normal Vector'' [8] belong to depth-based features and ''Actionlet Ensemble'' [37] belongs to skeleton+depth-based features. Since only depth information is used in our method, no comparison could be done with the methods which use RGBD data.

TABLE 1. Recognition accuracy comparison on the *MSRAction3D* dataset.

Method	Accuracy
Bag of 3D Points [11]	74.70%
Random Occupancy Pattern [40]	86.50%
Actionlet Ensemble [37]	88.20%
Depth Motion Maps [13]	88.73%
HON4D [17]	88.89%
DSTIP [12]	89.30%
H3DF [41]	89.45%
Skeletons Lie group [18]	89.48%
Skeletal Quads [42]	89.86%
HOG3D+LLC [28]	90.90%
Moving Pose [32]	91.70%
Hierarchical 3D Kernel [43]	92.73%
DMM-LBP-DF [15]	93.00%
Super Normal Vector [8]	93.09%
Depth Context [44]	94.28%
Hierarchical RNN [23]	94.49%
Range-Sample [27]	95.62%
Our Method with DMMs	**95.97%**
Our Method with Improved DMMs	**96.70%**

''Moving Pose'' [32] encodes 3D position, speed and acceleration of skeleton joints and achieves best performance among skeleton-based features. Despite the good performance of these methods, the skeleton data may not be reliable when the subject is not in an upright position. More over, the skeleton data is not available from depth sequences which contains partial human bodies, e.g. hands, arms and legs. Therefore, the application areas of skeleton-based methods are limited. Our method outperforms these methods for two reasons: first, skeleton joints used by these methods contain a lot of noises, which bring ambiguities to distinguish similar actions; second, our method directly uses DMMs, thus providing more effective motion information.

''Range-Sample'' [27] and ''Super Normal Vector'' [8] stand out from the depth-based features. In [27], the binary range-sample feature in depth is based on τ tests, which showed reasonable invariance to changes in scale, viewpoint and background. However, the temporal information among frames were not considered. In [8], an adaptive spatial-temporal pyramid based on the motion energy was proposed to globally capture the spatial and temporal orders. The dimension of action representation increased with the usage of larger number of levels. In [8], three levels were used, which achieved limited performances. Our result is better than the recent depth-based features such as ''Super Normal Vector'' [8] and ''Range-Sample'' [27], demonstrating the superior discriminatory power of our multi-temporal DMMs representation.

Using only patch-based LBP feature, DMM-LBP-DF [15] achieved an accuracy of 93.00%. Our Method with DMMs

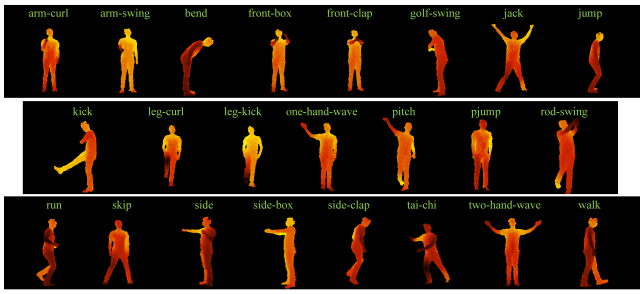


FIGURE 10. Action snaps in the *DHA* dataset.

outperformed DMM-LBP-DF by nearly 3%, which verifies that multi-temporal DMMs can efficiently capture temporal information. Using improved DMMs instead of DMMs, our method allowed achieving the highest accuracy of 96.70%, indicating that the improved DMMs outperformed DMMs by preserving additional temporal information. It is noted that the performance of the improved DMMs depends on a parameter w. Therefore, a proper parameter value of w for different datasets needs to be selected as illustrated in Figure 14. More details about the comparison between DMMs and the improved DMMs are stated in Section IV-G.

B. DHA DATASET

1) DATASET

DHA dataset is discussed in [45], whose action types are extended from the Weizmann dataset [46] which is widely used in action recognition from RGB sequences. It contains 23 action categories:"arm-curl", "arm-swing", "bend", "front-box", "front-clap", "golf-swing", "jack", "jump", "kick", "leg-curl", "leg-kick", "one-hand-wave", "pitch", "pjump", "rod-swing", "run", "skip", "side", "side-box", "side-clap", "tai-chi", "two-hand-wave", "walk", see Fig. 10 for some example depth frames. Each action is performed by 21 subjects (12 males and 9 females), resulting in 483 depth sequences. In the *DHA* dataset, "golf-swing" and "rod-swing" actions share similar motions by moving hands from one side up to the other side.

2) SETTINGS

Similar to [48], the leave-one-subject-out evaluation scheme was considered here, in which samples from one subject were chosen for testing and the remaining samples from the other subjects were used for training. Then, the overall accuracy was served as the evaluation criteria. The sizes of DMMs and blocks were set to the same values as the MSRAction3D dataset. All the other parameters were set to the default values.

3) COMPARISON WITH RELATED WORKS

The *DHA* dataset was originally collected by [45], which only contains 17 action categories. An extended version of the *DHA* dataset was used here where extra 6 action categories are involved. In [45], depth sequences were split into space-time volume and 3bit binary patterns were constructed as depth features, which achieved an accuracy of 86.80% on the original dataset. By incorporating the multi-temporal information to the DMMs, our method achieved higher accuracy even on the extended *DHA* dataset. From Table 2, it can be seen that our method outperformed D-DMHI-PHOG [48] by 3.04% and outperformed DMPP-PHOG [48] by 0.44%. These improvements show that operating LBP on multi-temporal DMMs can produce more informative features than operating PHOG on depth difference motion history images (D-MHI).

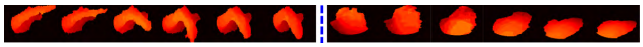


FIGURE 11. Actions "milk" (left) and "hungry" (right) in the *MSRGesture3D* dataset.

TABLE 2. Recognition accuracy comparison on the *DHA* dataset.

Method	Accuracy
D-STV/ASM [45]	86.80%
SDM-BSM [47]	89.50%
DMM-LBP-DF [15]	91.30%
D-DMHI-PHOG [48]	92.40%
DMPP-PHOG [48]	95.00%
Our Method with DMMs	**95.44%**
Our Method with Improved DMMs	**96.27%**

C. MSRGesture3D DATASET

1) DATASET

MSRGesture3D dataset [40] is a benchmark dataset for depth-based hand gesture recognition. It consists of 12 gestures defined by American Sign Language: "bathroom", "blue", "finish", "green", "hungry", "milk", "past", "pig", "store", "where", "j", "z". Each action is performed 2 or 3 times by each subject, resulting in 336 depth sequences. Not that 333 depth sequences from *MSRGesture3D* dataset were used here and 3 sequences which contains no depth data were discarded. In the *MSRGesture3D* dataset, actions such as "milk" and "hungry" are alike, see Fig. 11, since both actions involve the motion of bending palm.

2) SETTINGS

Similar to [17], the leave-one-subject-out evaluation scheme was employed and the overall accuracy was used to serve as the evaluation criteria. The sizes for DMM_f, DMM_s and DMM_t were 118×133, 118×29 and 29×133, respectively. The block sizes of the DMMs were considered to be 30×27, 30×15 and 15×27 corresponding to DMM_f, DMM_s and DMM_t. All the other parameters were assigned the default values.

3) COMPARISON WITH RELATED WORKS

In addition, a comparison with several existing methods were conducted whose results appear in Table 3. As can be seen

from this table, our method outperformed Histogram of Oriented Principal Components (HOPC) [26] by 1.96%.

TABLE 3. Recognition accuracy comparison on the *MSRGesture3D* dataset.

Method	Accuracy
Random Occupancy Pattern [40]	88.50%
HON4D [17]	92.45%
HOG3D+LLC [28]	94.10%
DMM-LBP-DF [15]	94.60%
Super Normal Vector [8]	94.74%
H3DF [41]	95.00%
Depth Gradients+RDF [49]	95.29%
Hierarchical 3D Kernel [43]	95.66%
HOPC [26]	96.23%
Our Method with DMMs	**98.19%**
Our Method with Improved DMMs	**99.39%**

D. MSRAction3D-SPEED DATASET

1) DATASET

MSRAction3D-Speed dataset is built using the *MSRAction3D* dataset. This dataset was used to test the robustness of our method to frame rate difference. Specifically, the sequences performed by subjects 1, 3, 5, 7, 9, (the original action samples) were used as the training data. One half of the frames (odd number frames, e.g., 1, 3, 5 ...) of the sequences performed by subjects 2, 4, 6, 8, 10 were selected. Based on the original time order, the selected frames were concatenated to form new sequences. Two types of sampling methods were used for sampling frames, i.e. linear sampling and random sampling. The linear sampling method sampled frames with $1/2$ of the original frame rate. While the random sampling method randomly sampled one half of the frames from the original sequences. As shown in Fig. 12, the random sampling-based *MSRAction3D-Speed* dataset was found to be more challenging than the linear sampling-based, since the speeds in the sampling-based dataset changed dramatically in a non-linear manner. Furthermore, many key frames got ignored by random sampling.

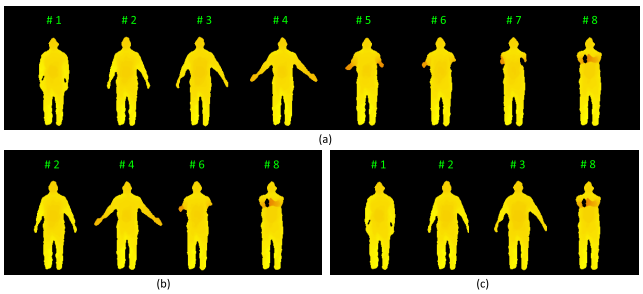


FIGURE 12. Comparison between linear sampling method and random sampling method. (a) A depth sequence of action "front-clap" from MSRAction3D dataset. (b) Linear sampling. (c) Random sampling.

2) SETTINGS

To facilitate a fair comparison with the results on *MSRAction3D* dataset, the cross subject validation method was conducted with subjects #1; 3; 5; 7; 9 for training and subjects #2; 4; 6; 8; 10 for testing. All the other parameters were set to the same ones as used for *MSRAction3D*.

3) COMPARISON WITH RELATED WORKS

In view of the achieved 95.97% recognition rate on the *MSRAction3D* dataset, our method exhibited resistance to the execution rate. The achieved recognition result of our method on linear sampling-based *MSRAction3D-Speed* dataset was 93.27%. Therefore, our method was capable of dealing with frame rate changes considering the fact that $1/2$ frame rate reduction was actually unrealistic. The recognition result of our method on random sampling-based *MSRAction3D-Speed* dataset was 90.10%. Using only patch-based LBP feature, DMM-LBP-DF [15] achieved an accuracy of 83.88%. Our method outperformed DMM-LBP-DF by 6.22%, verifying that our multi-temporal structure can efficiently capture temporal information, coping with the effect of speed changes.

E. MSRDailyActivity3D DATASET

1) DATASET

MSRDailyActivity3D dataset [50] is a daily activity dataset, which contains 16 activities: "drink", "eat", "read book", "call cellphone", "write on a paper", "use laptop", "use vacuum cleaner", "cheer up", "sit still", "toss paper", "play game", "lay down on sofa", "walk", "play guitar", "stand up" and "sit down". Each action is performed in two different poses: "sitting on sofa" and "standing" by each subject, resulting in 320 depth sequences. This dataset contains cluttered backgrounds and noise. Moreover, most of the actions contain human-object interactions which are illustrated in Fig. 13.

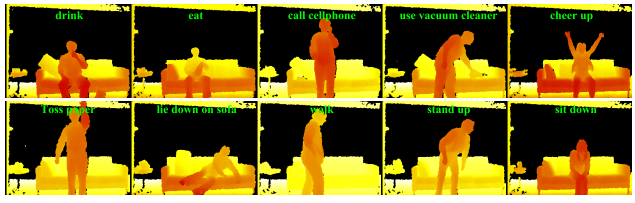


FIGURE 13. Action snaps from the *MSRDailyActivity3D* dataset.

2) SETTINGS

Similar to [12], the sequences in which the subject was almost still were removed. As a result, our experiments were conducted with ten types of actions. A cross-subject validation was performed with subjects 1,3,5,7,9 for training and subjects 2,4,6,8,10 for testing [50].

3) COMPARISON WITH RELATED WORKS

In Table 4, the comparison of our method with related works on the MSRDailyActivity3D dataset is provided. LOP feature [37] and Random Occupancy Pattern [40] are two typical features specially designed for encoding

overall accuracy (%)	DHA	MSRAction3D	MSRGesture3D	MSRAction3D-Speed	MSRDailyActivity3D
improved DMMs (w→0.0)	95.65	**96.70**	**99.39**	**91.57**	87
improved DMMs (w = 0.1)	95.65	96.34	**99.39**	90.10	88
improved DMMs (w = 0.2)	95.23	**96.70**	98.87	90.47	87
improved DMMs (w = 0.3)	**96.27**	95.97	99.20	90.47	88
improved DMMs (w = 0.4)	95.85	**96.70**	99.20	89.74	**89**
original DMMs (w→∞)	95.44	95.97	98.19	90.10	84

FIGURE 14. Recognition accuracies with different parameter *w*.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Recall(%)
highArmWave(1)	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
horizArmWave(2)	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
hammer(3)	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
handCatch(4)	0	0	0	9	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	75
forwardPunch(5)	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
highThrow(6)	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
drawX(7)	0	1	0	1	0	0	10	0	0	0	0	0	0	0	0	0	1	0	0	0	76.92
drawTick(8)	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	100
drawCircle(9)	0	1	0	0	0	0	0	1	13	0	0	0	0	0	0	0	0	0	0	0	86.67
handClap(10)	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	100
twoHandWave(11)	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	100
side-boxing(12)	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	100
bend(13)	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	100
forwardKick(14)	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	100
sideKick(15)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	100
jogging(16)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	100
tennisSwing(17)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	100
tennisServe(18)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	14	0	0	93.33
golfSwing(19)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	100
pickup&throw(20)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	100
Precision(%)	100	85.71	100	90	100	91.67	100	88.24	100	100	100	100	100	100	91.67	100	88.24	100	100	100	**96.70%**

FIGURE 15. Confusion matrice of our method with improved DMMs on the *MSRAction3D* dataset. The number in red indicates the overall accuracy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Recall(%)
arm-curl(1)	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	95.24
arm-swing(2)	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
bend(3)	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
front-box(4)	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
front-clap(5)	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
golf-swing(6)	0	0	0	0	0	16	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	76.19
jack(7)	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	95.24
jump(8)	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
kick(9)	0	0	0	0	0	0	0	0	20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	95.24
leg-curl(10)	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	100
leg-kick(11)	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	100
onhand-wave(12)	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	100
pitch(13)	0	0	0	0	0	0	0	0	0	0	0	0	20	0	1	0	0	0	0	0	0	0	0	95.24
pjump(14)	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	100
rod-swing(15)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	1	0	0	0	0	0	95.24
run(16)	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	17	0	0	0	2	0	0	0	80.95
side(17)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	100
side-box(18)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	100
side-clap(19)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	20	0	0	0	0	95.24
skip(20)	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	18	0	0	0	85.71
taichi(21)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	100
twohand-wave(22)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	100
walk(23)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	100
Precision(%)	100	100	100	100	100	100	100	87.5	100	100	95.45	100	100	100	76.92	89.47	100	91.3	100	90	100	91.3	100	**96.27%**

FIGURE 16. Confusion matrix of our method with improved DMMs on the *DHA* dataset. The number in red indicates the overall accuracy.

TABLE 4. Recognition accuracy comparison on the MSRDailyActivity3D dataset.

Methods	Accuracy
LOP feature [37]	42.50%
STIPs (Harris3D+HOG3D) [51]	60.60%
Random Occupancy Pattern [40]	64.00%
Joint position feature [37]	68.00%
STIPs (Cuboids+HOG/HOF) [52]	70.60%
DSTIP+DCSF [12]	83.60%
Actionlet Ensemble [37]	86.00%
Our Method with DMMs	**84.00%**
Our Method with Improved DMMs	**89.00%**

depth data. These methods achieved limited accuracies, which reflects the challenges (e.g. noise and cluttered backgrounds) of this dataset. To tackle with these challenges, DSTIP+DCSF [12] was recently designed, which achieves an accuracy of 83.60%. Our method achieved an improvement of 2.4% over DSTIP+DCSF, since more depth data was captured by our multi-temporal depth motion maps. Actionlet Ensemble in [37] combines both depth data and skeleton data and achieves the state-of-the-art result of 86%. Without using skeleton data, our method is still competitive with Actionlet Ensemble. Since our method does not rely on the skeleton data, our method is more suitable for the real-world scenes, where the skeleton data can be barely captured (e.g. facing the problems of partial occlusions, viewpoint changes).

F. COMPUTATION TIME

In our method, three levels for the multi-temporal DMMs representation is used. Our algorithm on the *MSRAction3D* dataset was tested using different numbers of temporal levels.

Class	1	2	3	4	5	6	7	8	9	10	11	12	Recall(%)
z(1)	27	0	1	0	0	0	0	0	0	0	0	0	96.43
j(2)	0	27	0	0	0	0	0	0	0	1	0	0	96.43
where(3)	0	0	28	0	0	0	0	0	0	0	0	0	100
store(4)	0	0	0	28	0	0	0	0	0	0	0	0	100
pig(5)	0	0	0	0	25	0	0	0	0	0	0	0	100
past(6)	0	0	0	0	0	28	0	0	0	0	0	0	100
hungry(7)	0	0	0	0	0	0	28	0	0	0	0	0	100
green(8)	0	0	0	0	0	0	0	28	0	0	0	0	100
finish(9)	0	0	0	0	0	0	0	0	28	0	0	0	100
blue(10)	0	0	0	0	0	0	0	0	0	28	0	0	100
bathroom(11)	0	0	0	0	0	0	0	0	0	0	28	0	100
milk(12)	0	0	0	0	0	0	0	0	0	0	0	28	100
Precision(%)	100	100	96.55	100	100	100	100	100	100	96.55	100	100	**99.39%**

FIGURE 17. Confusion matrix of our method with improved DMMs on the *MSRGesture3D* dataset. The number in red indicates the overall accuracy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Recall(%)
highArmWave(1)	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
horizArmWave(2)	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
hammer(3)	0	0	9	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	75
handCatch(4)	0	0	0	7	1	1	0	1	0	0	0	0	0	0	2	0	0	0	0	0	58.33
forwardPunch(5)	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	81.82
highThrow(6)	0	0	0	1	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90.91
drawX(7)	0	0	0	0	0	0	9	3	0	0	0	0	0	0	0	0	0	0	1	0	69.23
drawTick(8)	1	0	0	0	0	0	0	13	1	0	0	0	0	0	0	0	0	0	0	0	86.67
drawCircle(9)	0	0	0	0	0	0	1	2	12	0	0	0	0	0	0	0	0	0	0	0	80
handClap(10)	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	100
twoHandWave(11)	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	100
side-boxing(12)	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	100
bend(13)	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	100
forwardKick(14)	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	100
sideKick(15)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	2	0	81.82
jogging(16)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	100
tennisSwing(17)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	100
tennisServe(18)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	14	0	0	93.33
golfSwing(19)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	100
pickup&throw(20)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	100
Precision(%)	92.31	100	100	87.5	81.82	83.33	90	65	92.31	100	100	100	100	100	81.82	100	83.33	100	83.33	100	**91.57%**

FIGURE 18. Confusion matrix of our method with improved DMMs on the random sampling-based *MSRAction3D-Speed* dataset. The number in red indicates the overall accuracy.

Class	1	2	3	4	5	6	7	8	9	10	Recall(%)
drink(1)	10	0	0	0	0	0	0	0	0	0	100
eat(2)	0	10	0	0	0	0	0	0	0	0	100
call cellphone(3)	4	1	5	0	0	0	0	0	0	0	50
use vacuum(4)	0	0	0	10	0	0	0	0	0	0	100
cheer up(5)	0	0	0	0	9	0	1	0	0	0	90
toss paper(6)	1	0	3	1	0	5	0	0	0	0	50
lay down(7)	0	0	0	0	0	0	10	0	0	0	100
walk(8)	0	0	0	0	0	0	0	10	0	0	100
stand up(9)	0	0	0	0	0	0	0	0	10	0	100
sit down(10)	0	0	0	0	0	0	0	0	0	10	100
Precision(%)	66.67	90.91	62.5	90.91	100	100	90.91	100	100	100	**89.00%**

FIGURE 19. Confusion matrix of our method with improved DMMs on the *MSRDailyActivity3D* dataset. The number in red indicates the overall accuracy.

TABLE 5. Recognition accuracy and average feature computation time of our method with different numbers of temporal levels on the *MSRAction3D* dataset.

Temporal levels	Accuracy	Time/sequence (s)
1 level (Level 0)	89.95%	0.35
2 levels (Levels 0, 1)	93.34%	2.51
3 levels (Levels 0, 1, 2)	95.97%	4.49

The recognition accuracy and average feature computation time are reported in Table 5. It is worth mentioning that our algorithm is implemented in MATLAB and executed on CPU platform with an Intel(R)Core(TM)i7 CPU @2.60GHz and 8GB of RAM. The algorithm can be made more efficient by converting the code to C++ and running the multi-temporal DMMs representation in parallel.

G. IMPROVED DEPTH MOTION MAPS

In Figure 14, our improved DMMs are compared with the original DMMs. To implement the improved DMMs, the parameter w was changed from 0 to 0.4 in 0.1 intervals. In practice, "$w \rightarrow 0$" was implemented by setting w to 0.0001. It is noted that "$w \rightarrow 0$" infers to apply a linear weighting function and that "$w \rightarrow \infty$" infers to directly apply the original DMMs, without using the weighing scheme. On *MSRAction3D*, *MSRGesture3D* and *MSRAction3D-Speed*, linear weighting function ($w \rightarrow 0$) generates high performance. Meanwhile, our nonlinear weighting function obtains better results on *DHA* and *MSRDailyActivity3D*. In our experiments, proper w values need to be chosen for the improved DMMs, to achieve high performance. Generally speaking, our improved DMMs out-

performed original DMMs on all the datasets. The improvements indicate that the temporal information among frames increases the discriminant power of action representations.

The best performance results are shown in Figure 15,16,17,18,19. In Figure 15, the confusion matrix of the *MSRAction3D* dataset is shown with an accuracy of 96.70%. It is observed that large ambiguities exist between similar action pairs, for example ''handCatch'' and ''high-Throw'', and ''drawX'' and ''drawTick'', due to the similarities of their DMMs. In Figure 16, the confusion matrix of our method on the *DHA* dataset is shown with an accuracy of 96.27%. In Figure 17, the confusion matrix of the *MSRGesture3D* dataset is shown with an accuracy of 99.39%. It is observed that similar action pairs like ''milk'' and ''hungry'' can be distinguished with high accuracy. The confusion matrices of the *MSRAction3D-Speed* dataset is shown in Fig. 18, where the action ''drawX'' and ''drawTick'' have maximum confusion with each other since both actions contain similar motion and appearance. The recall rate and precision for most of the actions are beyond 91.57%, which verifies the robustness of our method to speed variations. In Figure 19, the confusion matrix of the *MSRDailyActivity3D* dataset is shown with an accuracy of 89.00%, indicating that our method can properly handle human-object interactions.

V. CONCLUSION

In this paper, we have presented a multi-temporal DMMs descriptor, which involves a nonlinear weighting function to accumulate the temporal depth frames. The inclusion of the temporal information helps to distinguish similar actions. To cope with speed variations in actions, a set of temporal intervals are utilized to construct multi-temporal DMMs. Moreover, instead of encoding DMMs as pixel-level features, the patch-based LBP feature extraction approach is adopted to characterize the rich texture information (e.g., edges, contours, etc.) in the LBP coded DMMs. The Fisher kernel representation is considered to aggregate local patch features into a compact and discriminative representation. The proposed method is extensively evaluated on five benchmark datasets. The experimental results show that our method outperforms the state-of-the-art methods in all datasets. Additional tests on our collected MSRAction3D-Speed dataset confirm that our method is able to handle depth sequences which contain dramatic frame rate differences. With the implementation of kernel-based extreme learning machine (ELM) classifier, our method can classify actions accurately in real-time.

REFERENCES

[1] N. Zhao, L. Zhang, B. Du, L. Zhang, D. Tao, and J. You, ''Sparse tensor discriminative locality alignment for gait recognition,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 4489–4495.

[2] C. Chen, N. Kehtarnavaz, and R. Jafari, ''A medication adherence monitoring system for pill bottles based on a wearable inertial sensor,'' in *Proc. EMBC*, Aug. 2014, pp. 4983–4986.

[3] L. Zhang, L. Zhang, D. Tao, and B. Du, ''A sparse and discriminative tensor to vector projection for human gait feature representation,'' *Signal Process.*, vol. 106, pp. 245–252, Jan. 2015.

[4] C. Chen, K. Liu, R. Jafari, and N. Kehtarnavaz, ''Home-based senior fitness test measurement system using collaborative inertial and depth sensors,'' in *Proc. EMBC*, Aug. 2014, pp. 4135–4138.

[5] V. Bloom, D. Makris, and V. Argyriou, ''G3D: A gaming action dataset and real time action recognition evaluation framework,'' in *Proc. CVPRW*, Jun. 2012, pp. 7–12.

[6] D. W. Tjondronegoro and Y.-P. P. Chen. ''Knowledge-discounted event detection in sports video,'' *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 5, pp. 1009–1024, Sep. 2010.

[7] H. Wang and C. Schmid, ''Action recognition with improved trajectories,'' in *Proc. ICCV*, Dec. 2013, pp. 3551–3558.

[8] X. Yang and Y. Tian, ''Super normal vector for activity recognition using depth sequences,'' in *Proc. CVPR*, Jun. 2014, pp. 804–811.

[9] B. Ni, G. Wang, and P. Moulin, ''RGBD-HuDaAct: A color-depth video database for human daily activity recognition,'' in *Proc. ICCVW*, Nov. 2011, pp. 1147–1153.

[10] J. Shotton *et al.*, ''Real-time human pose recognition in parts from single depth images,'' in *Proc. CVPR*, Jun. 2011, pp. 1297–1304.

[11] W. Li, Z. Zhang, and Z. Liu, ''Action recognition based on a bag of 3D points,'' in *Proc. CVPRW*, Jun. 2010, pp. 9–14.

[12] L. Xia and J. K. Aggarwal, ''Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera,'' in *Proc. CVPR*, Jun. 2013, pp. 2834–2841.

[13] X. Yang, C. Zhang, and Y. L. Tian, ''Recognizing actions using depth motion maps-based histograms of oriented gradients,'' in *Proc. ACM MM*, 2012, pp. 1057–1060.

[14] C. Chen, K. Liu, and N. Kehtarnavaz, ''Real-time human action recognition based on depth motion maps,'' *J. Real-Time Image Process.*, vol. 12, no. 1, pp. 155–163, 2013.

[15] C. Chen, R. Jafari, and N. Kehtarnavaz, ''Action recognition from depth sequences using depth motion maps-based local binary patterns,'' in *Proc. WACV*, Jan. 2015, pp. 1092–1099.

[16] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, ''Action recognition using 3D histograms of texture and a multi-class boosting classifier,'' *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4648–4660, Oct. 2017.

[17] O. Oreifej and Z. Liu, ''Hon4D: Histogram of oriented 4D normals for activity recognition from depth sequences,'' in *Proc. CVPR*, 2013, pp. 716–723.

[18] R. Vemulapalli, F. Arrate, and R. Chellappa, ''Human action recognition by representing 3D skeletons as points in a lie group,'' in *Proc. CVPR*, Jun. 2014, pp. 588–595.

[19] T. Ojala, M. Pietikäinen, and T. Mäenpää, ''Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[20] F. Perronnin, J. Sánchez, and T. Mensink, ''Improving the fisher kernel for large-scale image classification,'' in *Proc. ECCV*, 2010, pp. 143–156.

[21] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, ''Extreme learning machine: Theory and applications,'' *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[22] C. Chen, M. Liu, B. Zhang, J. Han, J. Jiang, and H. Liu, ''3D action recognition using multi-temporal depth motion maps and fisher vector,'' in *Proc. IJCAI*, 2016, pp. 1–7.

[23] Y. Du, W. Wang, and L. Wang, ''Hierarchical recurrent neural network for skeleton based action recognition,'' in *Proc. CVPR*, 2015, pp. 1110–1118.

[24] J. K. Aggarwal and X. Lu, ''Human activity recognition from 3D data: A review,'' *Pattern Recognit. Lett.*, vol. 48, no. 1, pp. 70–80, 2014.

[25] S. Tang *et al.*, *Histogram of Oriented Normal Vectors for Object Recognition With a Depth Sensor*. Berlin, Germany: Springer-Verlag, 2013.

[26] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, *HOPC: Histogram of Oriented Principal Components of 3D Pointclouds for Action Recognition*. New York, NY, USA: Springer-Verlag, 2014.

[27] C. Lu, J. Jia, and C.-K. Tang, ''Range-sample depth feature for action recognition,'' in *Proc. CVPR*, 2014, pp. 772–779.

[28] H. Rahmani, D. Q. Huynh, A. Mahmood, and A. Mian, ''Discriminative human action classification using locality-constrained linear coding,'' *Pattern Recognit. Lett.*, vol. 72, pp. 62–71, Mar. 2015.

[29] C. Chen, B. Zhang, Z. Hou, J. Jiang, M. Liu, and Y. Yang, ''Action recognition from depth sequences using weighted fusion of 2D and 3D auto-correlation of gradients features,'' *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 4651–4669, 2016.

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

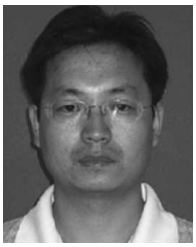
[31] H. Rahmani and A. Mian, "3D action recognition from novel viewpoints," in *Proc. CVPR*, 2016, pp. 1506–1515.
[32] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proc. ICCV*, 2013, pp. 2752–2759.
[33] T. Kerola, N. Inoue, and K. Shinoda, *Spectral Graph Skeletons for 3D Action Recognition*. New York, NY, USA: Springer-Verlag, 2014.
[34] X. Cai, W. Zhou, and H. Li, "Attribute mining for scalable 3D human action recognition," in *Proc. ACM MM*, 2015, pp. 1075–1078.
[35] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1017–1027, Apr. 2017.
[36] N. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, Aug. 2017.
[37] J. Wang, Z. Liu, J. Yuan, and Y. Wu, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, May 2014.
[38] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and HOG^2 for action recognition," in *Proc. CVPRW*, 2013, pp. 465–470.
[39] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.
[40] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *Proc. ECCV*, 2012, pp. 872–885.
[41] C. Zhang and Y. Tian, "Histogram of 3D facets: A depth descriptor for human action and hand gesture recognition," *Comput. Vis. Image Understanding*, vol. 139, pp. 29–39, Oct. 2015.
[42] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Proc. ICPR*, 2014, pp. 4513–4518.
[43] Y. Kong, B. Satarboroujeni, and Y. Fu, "Hierarchical 3D kernel descriptors for action recognition using depth sequences," in *Proc. FG*, 2015, pp. 1–6.
[44] M. Liu and H. Liu, "Depth Context: A new descriptor for human activity recognition by using sole depth sequences," *Neurocomputing*, vol. 175, pp. 747–758, Jan. 2015.
[45] Y.-C. Lin, M.-C. Hu, W.-H. Cheng, Y.-H. Hsieh, and H.-M. Chen, "Human action recognition and retrieval using sole depth information," in *Proc. ACM MM*, 2012, pp. 1053–1056.
[46] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
[47] H. Liu, L. Tian, H. Tang, and M. Liu, "SDM-BSM: A fusing depth scheme for human action recognition," in *Proc. ICIP*, 2015, pp. 4674–4678.
[48] Z. Gao, H. Zhang, G. P. Xu, and Y. B. Xue, "Multi-perspective and multi-modality joint representation and recognition model for 3D action recognition," *Neurocomputing*, vol. 151, pp. 554–564, Mar. 2015.
[49] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Real time action recognition using histograms of depth gradients and random decision forests," in *Proc. WACV*, 2014, pp. 626–633.
[50] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. CVPR*, 2012, pp. 1290–1297.
[51] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. BMVC*, 2008, pp. 275-1–275-10.
[52] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. Joint IEEE Int. Workshop Vis. Surveillance Perform. Eval. Tracking Surveillance*, Oct. 2005, pp. 65–72.

CHEN CHEN received the B.E. degree in automation from Beijing Forestry University, Beijing, China, in 2009, the M.S. degree in electrical engineering from Mississippi State University, Starkville, in 2012, and the Ph.D. degree from the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA, in 2016. He currently holds the Postdoctoral position at the Center for Research in Computer Vision, University of Central Florida. He has published over 50 papers in refereed journals and conferences in these areas. His research interests include compressed sensing, signal and image processing, pattern recognition, and computer vision.

Dr. Chen was the recipient of the David Daniel Fellowship Award (Best Doctoral Dissertation Award) from the University of Texas at Dallas in 2016. He received the top 10% paper award in the IEEE International Conference on Image Processing in 2015. He served as TPC member and the Reviewer for various conferences (e.g., ICIP, ICASSP, ICME, WACV, ACML, and CVPR). He is also an Active Reviewer for the IEEE TRANSACTIONS on GEOSCIENCE and REMOTE SENSING, the IEEE TRANSACTIONS on IMAGE PROCESSING, the IEEE TRANSACTIONS on MULTIMEDIA, the IEEE TRANSACTIONSon CIRCUITS and SYSTEMS for VIDEO TECHNOLOGY, IMAGE and VISION COMPUTING, and IEEE TRANSACTIONS on HUMAN–MACHINE SYSTEMS. He is an Associate Editor of KSII TRANSACTIONS on INTERNET and INFORMATION SYSTEMS, and SIGNAL, IMAGE and VIDEO PROCESSING JOURNAL.

MENGYUAN LIU (M'17) received the B.E. degree in intelligence science and technology in 2012, and Ph.D. degree from the School of Electrical Engineering and Computer Sciences, Peking University, China, in 2017. His research interests include action recognition using depth and skeleton data. He has published articles in Neurocomputing, MTA, ROBIO, ICIP, ICASSP, 3DV, and IJCAI.

HONG LIU (M'05) received the Ph.D. degree in mechanical electronics and automation from Peking University in 1996. He is currently a Full Professor with the School of Electrical Engineering and Computer Sciences, Peking University (PKU), China. He 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 the Open Laboratory on Human–Robot Interaction, PKU. His research fields include computer vision and robotics, image processing, and pattern recognition. He has published over 150 papers and received 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 a Vice President of Chinese Association for Artificial Intelligent, 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 the IEEE/RSJ IROS, the IEEE ROBIO, the IEEE SMC, and the IIHMSP, also serves as a reviewer for many international journals, such as *Pattern Recognition*, the IEEE TRANSACTIONS on SIGNAL PROCESSING, and the IEEE TRANACTIONS on Pattern Analysis and Machine Intelligence.



BAOCHANG ZHANG received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1999, 2001, and 2006, respectively. From 2006 to 2008, he was a Research Fellow with the Chinese University of Hong Kong, Hong Kong, and with Griffith University, Brisbane, QLD, Australia. He is currently an Associate Professor with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. He also held a Senior Postdoctoral Position with the PAVIS Department, Italian Institute of Technology, Genoa, Italy. He was supported by the Program for New Century Excellent Talents with the University of Ministry of Education of China. His current research interests include pattern recognition, machine learning, face recognition, and wavelets.

JUNGONG HAN is currently a tenured Faculty Member with the School of Computing and Communications, Lancaster University, Lancaster, U.K. He was a Faculty Member with the Department of Computer and Information Sciences, NorthUmbria University, U.K. His research interests include computer vision, image processing, machine learning, and artificial intelligence.

NASSER KEHTARNAVAZ (F'12) received the Ph.D. degree in electrical and computer engineering from Rice University, in 1987. He is currently a Professor of Electrical Engineering and the Director of the Signal and Image Processing Laboratory, University of Texas at Dallas. He has had industrial experience in various capacities with Texas Instruments, AT&T Bell Labs, the U.S. Army TACOM Research Laboratory, and the Houston Health Science Center. His research areas include signal and image processing, real-time processing on embedded processors, biomedical image analysis, and pattern recognition. He has authored or coauthored nine books and over 300 publications in these areas. He is a Fellow of SPIE and a licensed Professional Engineer. He is currently the Editor-in-Chief of *Journal of Real-Time Image Processing*, and the Chair of the SPIE Conference on Real-Time Image and Video Processing.