This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

Action Recognition Using 3D Histograms of Texture and A Multi-class Boosting Classifier

Baochang Zhang*, Yun Yang*, Chen Chen, Linlin Yang, Jungong Han, Ling Shao, *Senior Member, IEEE*

***Abstract*— Human action recognition is an important yet challenging task. This paper presents a low-cost descriptor called 3D Histograms of Texture (*3DHoTs*) to extract discriminant features from a sequence of depth maps. 3DHoTs are derived from projecting depth frames onto three orthogonal Cartesian planes, i.e., the frontal, side and top planes, and thus compactly characterize the salient information of a specific action, on which texture features are calculated to represent the action. Besides this fast feature descriptor, a new multi-class boosting classifier (*MBC*) is also proposed to efficiently exploit different kinds of features in a unified framework for action classification. Compared to the existing boosting frameworks, we add a new multi-class constraint into the objective function, which helps to maintain a better margin distribution by maximizing the mean of margin whereas still minimizing the variance of margin. Experiments on the MSRAction3D, MSRGesture3D, MSRActivity3D and UTD-MHAD datasets demonstrate that the proposed system combining 3DHoTs and MBC is superior to the state-of-the-art.**

***Index Terms*— Action recognition, multi-class classification, boosting classifier, depth image, texture feature.**

I. Introduction

HUMAN action recognition has been an active research topic in computer vision in the past 15 years. It can facilitate a variety of applications, ranging from human computer interaction [1]-[3], motion sensing based gaming, intelligent surveillance to assisted living [4]. Early research mainly focuses on identifying human actions from video sequences captured by RGB video cameras. In [5], binary motion-energy images (MEI) and motion-history images (MHI) are used to represent where motion has occurred and characterize human actions. In [6], a low computational-cost volumetric action representation from different view angles is utilized to obtain high recognition rates. In [7], the notion of spatial interest points is extended to the spatio-temporal domain based on the idea of the Harris interest point operator. The results show its robustness to occlusion and noise. In [8], a motion descriptor built upon the spatio-temporal optical flow measurement is introduced to deal with low resolution images.

Despite the great progress in the past decades, recognizing actions in the real world environment is still problematic. With the development of RGB-D cameras, especially Microsoft Kinect, more recent research works focus on action recognition using depth images [9], [10] due to the fact that depth information is much more robust to changes in lighting conditions, compared with the conventional RGB data. In [11], a bag of 3D points corresponding to the nodes in an action graph is generated to recognize human actions from depth sequences. Alternatively, an actionlet ensemble model is proposed in [12] and the developed local occupancy patterns are shown to be immune to noise and invariant to translational and temporal misalignments. In [13], Histograms of Oriented Gradients (HOG) computed from Depth Motion Maps (DMMs) are generated, capturing body shape and motion information from depth images. In [14], Chen *et al.* combine Local Binary Pattern (LBP) and the Extreme Learning Machine (ELM), achieving the best performance on their own datasets. In summary, although depth based methods have been popular, they cannot perform reliably in practical applications where large intra-class variations, e.g., the action-speed difference, exist. Such a drawback is mainly caused by two algorithm designing faults. First, the visual features fed into the classifier are unable to obtain different kinds of discriminating information, the diversity of which is required in building a robust classifier. Second, few works take the theoretical bounds into account when combining different learning models for classification. We perceive that most existing works empirically stack up different learning models without any theoretical guidance, even though the results are acceptable in some situations.

To improve the robustness of the system, especially for practical application usage, we propose a feature descriptor, namely 3D Histograms of Texture (*3DHoTs*), which is able to extract discriminative features from depth images. More specifically, 3DHoT is an extension of our previous DMM-LBP descriptor in the sense that the complete local binary pattern (CLBP) proposed in [15] for texture classification is employed to capture more texture features, thereby enhancing the feature representation capacity. This new feature is able to describe the motion information from various perspectives such as sign, magnitude and local difference based

The work was supported in part by the Natural Science Foundation of China under Contract 61672079 and 61473086. The work of B. Zhang was supported in part by the Beijing Municipal Science and Technology Commission under Grant Z161100001616005.and by the Open Projects Program of National Laboratory of Pattern Recognition. (Corresponding author: Jungong Han)

Baochang Zhang, Yun Yang and Linlin Yang are with Beihang University, Beijing, China. ({bczhang, yanglinlin}@buaa.edu.cn). *Equal contribution.

Yun Yang is with Computer Vision Laboratory, Noah's Ark Lab, Huawei Technologies, Beijing, China. (yangyun18@huawei.com).

Chen Chen is with Center for Research in Computer Vision (CRCV), University of Central Florida, Orlando, FL, USA. (chenchen870713@gmail.com).

Jungong Han is with the School of Computing & Communications, Lancaster University, Lancaster LA1 4YW, UK. (jungonghan77@gmail.com).

Ling Shao is with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K. Email: ling.shao@ieee.org. (ling.shao@ieee.org).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

on the global center. Besides, we also improve the classification by combining the extreme learning machine (ELM) and a new multi-class boosting classifier (*MBC*). This paper is an extension of [60] in the sense that we provide the theoretical derivation of our objective which aims to minimize the variance of margin samples following the Gaussian Mixture Model (GMM) distribution. From the theoretical perspective, our classification technique is an ensemble of base classifiers on different types of features, making it possible to tackle extremely challenging action recognition tasks. In summary, our work differs from the existing work in two aspects.

1. The primary contribution lies in a multi-class boosting classifier, which enables to exploit different kinds of features in a unified framework. Compared to the existing boosting frameworks, we add a new multi-class constraint into the objective function, which helps to maintain a better margin distribution by maximizing the mean margin while controlling the margin variance even if the margin samples follow a complicated distribution, i.e., GMM.

2. We enhance our previous DMM-LBP descriptor [9] by using a more advanced texture extraction model CLBP [15]. This new 3DHoTs feature combining DMM and CLBP encodes motion information across depth frames and local texture variation simultaneously. Using this representation can improve the performance of depth-based action recognition, especially for realistic applications.

The rest of the paper is organized as follows. Section II briefly reviews related work on depth feature representations. Section III describes the details of 3DHoT features. Section IV introduces the multi-class boosting method as well as its theoretical discussions. Experimental results are given in Section V. Some concluding remarks are drawn in Section VI.

II. Related Work

Recently, depth based action recognition methods have gained much attention due to their robustness to changes in lighting conditions [16]. Researchers have made great efforts to obtain a distinctive action recognition system based on depth or skeleton models. This section presents a review on related work with focuses on feature representations for depth maps and classifier fusion, which are in line with our two contributions.

A. Feature representation for action recognition

Two commonly used visual features for action recognition are handcrafted feature and learned feature. The former captures certain motion, shape or texture attributes of the action using statistical approaches while the latter automatically obtains intrinsic representations from a large volume of training samples in a data-driven manner [17].

Skeleton joints from depth images are typical handcrafted features for use in action recognition, because they provide a more intuitive way to perceive human actions. In [18], robust features based on the probability distribution of skeleton data were extracted and followed by a multivariate statistical method for encoding the relationship between the extracted features. In [19], Ofli *et al.* proposed a Sequence of Most Informative Joints (SMIJ) based on the measurements, such as the mean and variance of joint angles and the maximum angular velocity of body joints. A descriptor named Histogram of Oriented Displacements (HOD) was introduced in [20], where each displacement in the trajectory voted with its length in a histogram of orientation angles. In [21], a HMM-based methodology for action recognition was developed using star skeleton as a representative descriptor of human postures. Here, a star-like five-dimensional vector based on the skeleton features was employed to represent local human body extremes, such as head and four limbs. In [22], Luo *et al.* utilized the pairwise relative positions between joints as the visual features and adopted a dictionary learning algorithm to realize the quantization of such features. Both the group sparsity and geometry constraints are incorporated in order to improve the discriminative power of the learned dictionary. This approach has achieved the best results on two benchmark datasets, thereby representing the current state-of-the-art. Despite the fact that skeleton-based human action recognition has achieved surprising performance, large storage requirement and high dimensionality of the feature descriptor make it impractical, if not impossible, to be deployed in real scenarios, where low-cost and fast algorithm is demanded.

Alternatively, another stream of research tried to capture motion, shape and texture handcrafted features directly from the depth maps. In [23], Fanello *et al.* extracted two types of features from each image, namely Global Histograms of Oriented Gradients (GHOGs) and 3D Histograms of Flow. The former was designed to model the shape of the silhouette while the latter was to describe the motion information. These features were then fed into a sparse coding stage, leading to a compact and stable representation of the image content. In [24], Tran and Nguyen introduced an action recognition method with the aid of depth motion maps and a gradient kernel descriptor which was then evaluated using different configurations of machine learning techniques such as Support Vector Machine (SVM) and kernel based Extreme Learning Machine (KELM) on each projection view of the motion map. In [25], Zhang *et al.* proposed an effective descriptor, called Histogram of 3D Facets (H3DF), to explicitly encode the 3D shape and structures of various depth images by coding and pooling 3D Facets from depth images. In [66], the kernel technique is used to improve the performance for processing nonlinear quaternion signals; in addition, both RGB information and depth information are deployed to improve representation ability.

Different from the above methods that rely on handcraft features, deep models learn the feature representation from raw depth data and appropriately generate the high level semantic representation. In our previous work [26], Wang *et al.* proposed a new deep learning framework, which only required small-scale CNNs but achieved higher performance with less computational costs. In [27], DMM-Pyramid architecture that can partially keep the temporal ordinal information was proposed to preprocess the depth sequences. In their system, Yang *et al.* advocated the use of the convolution operation to extract spatial and temporal features from raw video data automatically and extended DMM to DMM-Pyramid. Subsequently, the raw depth sequences can be accepted by both 2D and 3D convolutional networks.

From the extensive work on depth map based action

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

recognition, we have observed that depth maps actually contain rich discriminating texture information. However, most methods do not take it into account when generating their feature representations.

B. Classifier fusion

In a practical action recognition system, the classifier plays an important role in determining the performance of the system, thereby gaining much attention. Most existing systems just adapted the single classifier, such as SVM [28], ELM [29] and HMM [21], into the action recognition field, and are sufficiently accurate when recognizing simple actions like sitting, walking and running. However, for more complicated human actions, such as hammering a nail, existing works have proved that combining multiple classifiers especially weak classifiers usually improves the recognition rate. Apparently, how to combine basic classifiers becomes crucial.

In [9], Chen *et al.* employed three types of visual features, each being fed into a KELM classifier. At the decision level, a soft decision fusion scheme, namely logarithmic opinion pool (LOGP) rule, merged the probability outputs and assigned the final class label. Instead of using specific fusion rules, most algorithms adopted the boosting schemes, which iteratively weigh different single classifiers by manipulating the training dataset, and on top of it, selectively combine them depending on the weight of each classifier. For example, a boosted exemplar learning (BEL) approach [30] was proposed to recognize various actions, where several exemplar-based classifiers were learned via multiple instance learning, given a certain number of class-specific candidate exemplars. Afterwards, they applied AdaBoost to integrate the further selection of representative exemplars and action modeling.

Recently, considerable research has been devoted to multi-class boosting classification as it is able to facilitate a broad range of applications including action recognition [31]-[33]. Flowing [32] [39] and many other publications, we generally divide the existing works into two categories depending on how they solved the M-ary (M>2) problems. In the first category, the proposed approaches decompose the desired multi-class problem into a collection of multiple independent binary classification problems, basically treating an *M* class problem as an estimation of a two-class classifier on the training set *M* times. Representatives include ECOC [31], AdaBoost.MH [34], binary GentleBoost algorithm [35], and AdaBoost.M2 [36]. In general, this type of multi-class boosting methods can be easily implemented based on the conventional binary AdaBoost, however, the system performance is not satisfactory due to the fact that binary boosting scores do not represent true class probabilities. Additionally, such a two-step scheme inevitably creates resource problems by increasing the training time and memory consumption, especially when dealing with a large number of classes.

To overcome this drawback, the second approach *directly* boosts an M-ary classifier via optimizing a multi-class exponential loss function. One of the first attempts was the AdaBoost.M1 algorithm [36]. Similar to the binary AdaBoost method, this algorithm allowed for any weak classifier that has an error rate of less than 0.5. In [38], a new variation of the AdaBoost.M1 algorithm, named ConfAdaBoost.M1, was presented, which used the information about how confident the weak learners are to predict the class of the instances. Many researches boosted M-ary classifier by redefining the objective functions. For example, in [37] Zou *et al.* extended the binary Fisher-consistency result to multi-class classification problems, where the smooth convex Fisher-consistent loss function is minimized by employing gradient decent. Alternatively, Shen *et al.* [32] presented an extension of the binary totally-corrective boosting framework to the multi-class case by generalizing the concept of separation hyperplane and margin derived from the famous SVM classification. Moreover, the class label representation problem is discussed in [33], which exploited different vector encodings for representing class labels and classifier responses to model the uncertainty caused by each weak-learner. From the perspective of margin theory as shown in [39], researchers defined a proper margin loss function for M-ary classification and identified an optimal codebook. And they further derived two boosting algorithms for the minimization of the classification risk. In [40], Shen *et al.* assumed a Gaussian distribution of margin and obtained a new objective, which is one of the most well-known theoretical results in the field.

To sum up, most of existing works, especially the multi-class ones focused on solving weak classifier selection and the imbalance problem by introducing more robust loss functions. From the margin theory perspective [40], they are only able to maximize the hard-margin or the minimum margin when the data follows a simple distribution (Gaussian). According to the theoretical evidences in [40], a good boosting framework should aim for maximizing the average margin. Such problems were addressed in other learning methods, e.g., SVM, by employing the soft-margins, which actually inspired our work. Unlike [40] and other existing works [31], [32], [39], we assume a more reasonable multiple Gaussian distribution of margin. When dealing with a multiple-class (one versus all) problem, evidently it is hard to assume that the margin follows a single Gaussian. Based on our GMM assumption, we design an objective function, intending to minimize the variance of margin samples that follow the GMM distribution.

III. 3D Histograms of Texture

On a depth image, the pixel values indicate the distances between the surface of an object and a depth camera location, therefore providing 3D structure information of a scene. Commonly, researchers utilize the 3D information in the original 3D space, but we project each depth frame of a depth sequence onto three orthogonal Cartesian planes so as to make use of both the 3D structure and shape information [13]. Basically, our 3DHoTs feature extraction and description consists of two steps: salient information map generation and CLBP based feature description, each being elaborated below.

A. Salient information (SI) map generation

The idea of SI is derived from DMM [13], which is generated by stacking motion energy of depth maps projected onto three

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

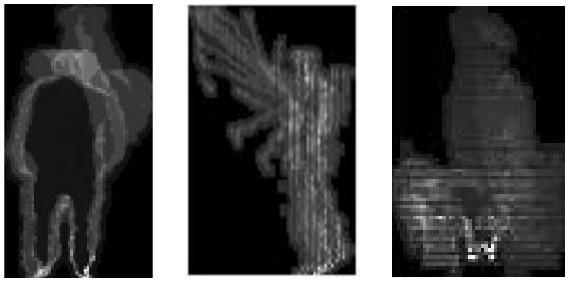

Fig. 1. Salient Information (SI) maps. From the left to the right: front (f) view, side (s) view and top (t) view.

orthogonal Cartesian planes. After obtaining each projected map, its motion energy is computed by thresholding the difference between consecutive maps. The binary map of motion energy provides a strong clue of the action category being performed and indicates motion regions or where movement happens in each temporal interval.

More specifically, each 3D depth frame generates three 2D projected maps aligning with front (f), side (s), and top (t) views, i.e., p^f , p^s and p^t , respectively. The summation of the absolute differences of consecutive projected maps can be used to imply the motion within a region. The larger the summation value, the more likely the motion frequently occurs in that region. Considering both the discriminability and robustness of feature descriptors, authors used the L1-norm of the absolute difference between two projected maps to define salient information (SI) in [14]. On the one hand, the summation of L1-norm is invariant to the length of a depth sequence. That is to say, we will be less influenced by mismatched speeds of performing the same action by different people. On the other hand, L1-norm contains more salient information than other norms (i.e., L2) and it is fast to compute. Consequently, the SI maps of a depth sequence are computed as:

$$SI^* = \sum_{i=1}^{B-v} \left| p_{i+v}^* - p_i^* \right|, \tag{1}$$

where $*$ denotes *f*, *s* or *t*. The parameter v stands for the frame interval, i represents the frame index, and B is the total number of frames in a depth sequence. An example of the SI maps of a depth action sequence is shown in Fig. 1. In the case that the sum operation in Eq. (1) is only used given a threshold satisfied, it is similar to the idea of [13].

Instead of selecting frames as in original DMM [13], however, in [60], the authors proposed that all frames should be deployed to calculate motion information. As shown in Eq. (2), the SI map for $v=1$ contains more salient information than that of $v=2$:

$$\begin{aligned} &2\left(\left|p_2 - p_1\right| + \sum_{i=2}^{N-2} \left|p_{i+1} - p_i\right| + \left|p_N - p_{N-1}\right|\right) \\ &\geq \left|p_2 - p_1\right| + 2\sum_{i=2}^{N-2} \left|p_{i+1} - p_i\right| + \left|p_N - p_{N-1}\right| \geq \sum_{i=1}^{N-2} \left|p_{i+2} - p_i\right|. \end{aligned} \tag{2}$$

The scale in the above expression affects little on the local pattern histogram. The result is evident, considering the fact that:

$$\left|p_{i+2} - p_{i+1}\right| + \left|p_{i+1} - p_i\right| \geq \left|p_{i+2} - p_i\right|. \tag{3}$$

Instead of accumulating binary maps result from comparing with the threshold, SI obtains more detailed feature than original DMM does, based on which we further introduce a powerful texture descriptor inspired by CLBP [15] method.

B. CLBP based descriptor

Our CLBP based descriptors represent SI maps from three aspects, which are:

1. Sign based descriptor for Salient Information

Given a center pixel t_c in the ***SI*** image, its neighboring pixels are equally scattered on a circle with radius r $(r>0)$. If the coordinates of t_c are $(0,0)$ and m neighbors $\{t_i\}_{i=0}^{m-1}$ are considered, the coordinates of t_i are $(-r\sin(2\pi i/m), r\cos(2\pi i/m))$. The sign descriptor is computed by thresholding the neighbors $\{t_i\}_{i=0}^{m-1}$ with the center pixel t_c to generate an m-bit binary number, so that it can be formulated as:

$$Sign_{m,r}(t_c) = \sum_{i=0}^{m-1} s(t_i - t_c)2^i = \sum_{i=0}^{m-1} s(d_i)2^i, \tag{4}$$

where $d_i = (t_i - t_c)$. $s(d_i) = 1$ if $d_i \geq 0$ and $s(d_i) = 0$ if $d_i < 0$. After obtaining the sign based encoding for pixels in an SI image, a block-wise statistic histogram named *HoT_S* is computed over an image or a region to represent the texture information.

2. Magnitude based descriptor for Salient Information

The magnitude is complementary to sign information in the sense that the difference d_i can be reconstructed based on them. Fig. 2 shows an example of the sign and magnitude components extracted from a sample block. The local differences are decomposed into two complementary components: the signs and magnitudes (absolute values of d_i , i.e. $|d_i|$). Note that "0" is coded as "-1" in the encoding process (see Fig. 2 (c)). The magnitude operator is defined as follows:

$$\begin{aligned} &Magnitude_{m,r} = \sum_{i=0}^{m-1} \varphi\left(\left|d_i\right|, c\right)2^i, \\ &\varphi(\sigma, c) = \begin{cases} 1, & \sigma \geq c \\ 0, & \sigma < c \end{cases}, \end{aligned} \tag{5}$$

where c is a threshold setting to the mean value of $|d_i|$ on the whole image. A block-wise statistic histogram named HoT_Magnitude (*HoT_M*) is subsequently computed over an image or a region.

1. Center based descriptor for Salient Information

The center part of each block which encodes the values of the center pixels also provides discriminant information. It is denoted as:

$$Center_{m,r} = \varphi\left(t_c, c_1\right), \tag{6}$$

where φ is defined in Eq. (5) and the threshold c_1 is set as the average gray level of the whole image. Subsequently, we obtain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

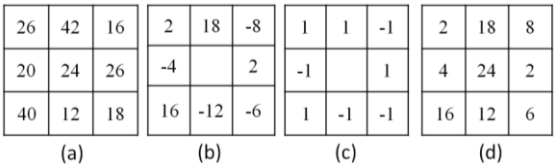

Fig. 2. Sign and magnitude components extracted from a sample block. (a) 3×3 sample block; (b) the local differences; (c) the sign component of block; and (d) the magnitude component of block.

the histograms of center based texture feature (*HoT_C*) over a SI image or a region.

To summarize, in our feature extraction method, each depth frame from a depth sequence are first projected onto three orthogonal Cartesian planes to form three projected maps. Under each projection plane, the absolute differences between the consecutive projected maps are accumulated over an entire sequence to generate a corresponding SI image. Then each SI image is divided into overlapped blocks. Each component of the texture descriptors is applied to the blocks and the resulted local histograms of all blocks are concatenated to form a single feature vector. Therefore, each SI image creates three histogram feature vectors denoted by HoT_*_S, HoT_*_M and HoT_*_C, respectively. Since there are three SI images corresponding to three projection views (i.e., front, side and top views), three feature vectors are generated as final feature vectors as follows. The feature extraction procedure is illustrated in Fig. 3.

$$3DHoT\_S = [HoT_f\_S,\ HoT_s\_S,\ HoT_t\_S]$$
$$3DHoT\_\mathrm{M} = [HoT_f\_\mathrm{M},\ HoT_s\_\mathrm{M},\ HoT_t\_\mathrm{M}]$$
$$3DHoT\_\mathrm{C} = [HoT_f\_\mathrm{C},\ HoT_s\_\mathrm{C},\ HoT_t\_\mathrm{C}]$$

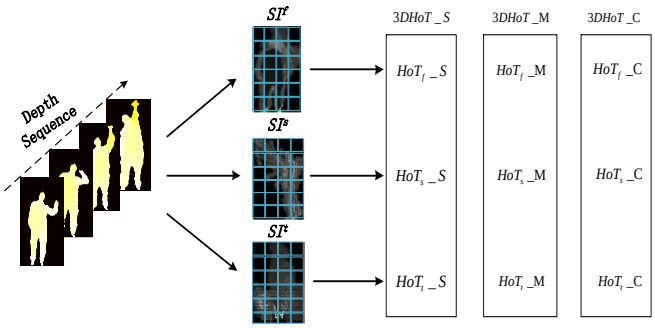

Fig. 3. Pipeline of *3DHoTs* feature extraction.

IV. Decision-level classifier fusion based on Multi-class boosting scheme

As can be seen, we use *multi-view* features in order to capture the diversity of the depth image. Normally, the dissimilarity among features from different views is large. To solve this multi-view data classification problem, the majority of the research in this field advocates the use of the boosting method. The basic idea of a boosting method is to optimally incorporate multiple weak classifiers into a single strong classifier. Here, one view of features can be fed into one weak classifier.

As an outstanding boosting representative, AdaBoost [40] incrementally builds an ensemble by training each new model instance to emphasize the training instances that are mis-classified previously. In this paper, we concentrate on this framework, based on which we introduce a new multi-class boosting method.

Supposed we have n weak/base classifiers and $h_i(x)$ denotes the i^{th} base classifier, a boosting algorithm actually seeks for a convex linear combination:

$$F(\alpha, x) = \sum_{i=1}^{n} \alpha_i h_i(x), \tag{7}$$

where α_i is a weight coefficient corresponding to the i^{th} weak classifier. Apparently, AdaBoost method can be decomposed into two modules: base classifier construction and classifier weight calculate, given training samples.

A. Base classifier: Extreme learning machine

In principle, the base classifiers in AdaBoost can be any existing classifiers performing better than random guessing. But the better a base classifier is, the greater the overall decision system performs. Therefore, we use the ELM method [29] in our work, which is an efficient learning algorithm for single-hidden-layer feed-forward neural networks (SLFNs). More specifically, let $\mathbf{y} = [y_1, ..., y_k, ..., y_C]^T \in \mathbf{R}^C$ be the class to which a sample belongs, where $y_k \in \{1, -1\}$ $(1 \le k \le C)$ and C is the number of classes. Given N training samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbf{R}^M$ and $\mathbf{y}_i \in \mathbf{R}^C$, a single hidden layer neural network having L hidden nodes can be expressed as

$$\sum_{j=1}^{L} \boldsymbol{\beta}_j h(\mathbf{w}_j \cdot \mathbf{x}_i + e_j) = \mathbf{y}_i, \quad i = 1, ..., N, \tag{8}$$

where $h(\cdot)$ is a nonlinear activation function (e.g., Sigmoid function), $\boldsymbol{\beta}_j \in \mathbf{R}^C$ denotes the weight vector connecting the j^{th} hidden node to the output nodes, $\mathbf{w}_j \in \mathbf{R}^M$ denotes the weight vector connecting the j^{th} hidden node to the input nodes, and e_j is the bias of the j^{th} hidden node. The above N equations can be written compactly as:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}, \tag{9}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T; ...; \boldsymbol{\beta}_L^T] \in \mathbf{R}^{L \times C}$, $\mathbf{Y} = [\mathbf{y}_1^T; ...; \mathbf{y}_N^T] \in \mathbf{R}^{N \times C}$, and $\mathbf{H}$ is the hidden layer output matrix. A least-squares solution $\hat{\boldsymbol{\beta}}$ of (8) is found to be

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{Y}, \tag{10}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$. The output function of the ELM classifier is

$$\mathbf{f}_L(\mathbf{x}_i) = \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{h}(\mathbf{x}_i)\mathbf{H}^T \left(\frac{\mathbf{I}}{\rho} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{Y}, \tag{11}$$

where $1/\rho$ is a regularization term and ρ is set to be 1000. The label of a test sample is assigned to the index of the output nodes with the largest value. In our experiments, we use a kernel-based ELM (KELM) with a radial basis function (RBF) kernel (the parameter gamma in RBF is set to be 10.5).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

B. Multi-class boosting classifier

Having specified the base classifier, the next step is to introduce our new multi-class boosting classifier. Our investigation is carried out from the perspective of margin sample distribution, in contrast to the traditional methods that focus on solving the weak classifier selection and the imbalance problem. One of the obvious advantages lies in the alleviation of the over-fitting problem through weighing the samples. As another intuition, inspired by [40], we investigate AdaBoost based on a more reasonable hypothesis on the margin distribution and obtain a new theoretical result.

Following Eq. (7), AdaBoost is equivalent to minimizing the exponential loss function [42]:

$$\min_{\alpha} \sum_{i=1}^{N} \exp(-y_i F(\alpha, x_i)), \quad s.t. \quad \alpha \geq 0 . \quad (12)$$

The logarithmic function $\log(\cdot)$ is a strictly monotonically increasing function and it is easy to calculate the minimum value of a non-exponential function. Therefore, after a logarithmic processing, AdaBoost equals to solve [42]:

$$\min_{\alpha} \log\left(\sum_{i=1}^{N} \exp(-y_i F(\alpha, x_i))\right), \ s.t. \ \alpha \geq 0, \|\alpha\|_1 = \delta . \quad (13)$$

The constraint $\|\alpha\|_1 = \delta$ avoids enlarging the solution α by an arbitrary large factor to make the cost function approach zero in the case of separable training data. In [43], Crammer and Singer propose to construct multiclass predictors with a piecewise linear bound. Considering the simplicity and the efficiency of a linear function, we use the following rule for this *C*-class classification,

$$\arg\max_{j=1}^{C} \{\theta^{T,j} \cdot x\}, \quad (14)$$

where θ^{j} is a vector. And then we heuristically propose the following linear objective function:

$$\max_{j} (\theta^{T,j} \cdot x - \theta^{T,m} \cdot x), \quad (15)$$

where $m \neq j$. Next, we incorporate this linear objective and a multiple-class constraint into a simple form of AdaBoost described in Eq. (13). Eventually, a multi-class boosting method to calculate the weight vector separately for each class can be achieved through minimizing the following objective:

$$\min_{j} \left(\log\left(\sum_{i} \omega_i \exp(-y_i F(\theta^{j}, x_i))\right) + \frac{1}{N_j} \sum_{i} (\theta^{T,m} \cdot x_i^j - \theta^{T,j} \cdot x_i^j) + \lambda \times \|\theta^{j}\|_1\right) \quad (16)$$

The effect of λ on the system performance is investigated in the experimental results part. x_i^j denotes the i^{th} sample in the j^{th} class with N_j samples. We make use of the interior point method to solve our objective. Here, we further discuss the theoretical advantage behind the new objective function.

The margin theory used in SVM is the state-of-the-art learning principle. The so-called dual form of AdaBoost is another significant work related to the margin theory. The latter one is quite close to our work, which is briefly introduced with the focus on explaining their difference. In [40], authors assume a Gaussian distribution of margin, and based on it, they theoretically explain the state-of-the-art margin method (AdaBoost). However, for a multiple-class (one versus all) problem, it is hard, if not impossible, to assume that the margin follows a single Gaussian. Instead, we presume that the margin follows the multiple Gaussian models. It is believed that assuming multiple Gaussian distribution models in a more complicated situation like our problem here is sensible, as a single Gaussian model is widely accepted in the theoretical analysis for a simple situation.

After settling the data distribution, the next question becomes whether our objective function maximizes the mean of margin and at the same time minimizes the variance of margin that follows Gaussian mixture models. It was stated in [40] that the success of a boosting algorithm can be understood in terms of maintaining a better margin distribution by maximizing margins and meanwhile controlling the margin variance. In order words, it can be a sort of criterion to measure the proposed boosting algorithm. In our case, proving it is not easy, since we have assumed that samples from different classes might follow GMM but not a single Gaussian. As another motivation in [40], the boosting method can be used to solve various complex problems, but few researchers explain it from a theoretical aspect. We present a theorem to answer the question mentioned above. Based on Lemmas 1 and 2 in Appendix, we obtain new theoretical results for our boosting methods, and significantly extend the original one in [36]. Here we describe our algorithm as follows:

Algorithm 1: We solve our objective based on the MATLAB toolbox. Our method utilizes the information derived from depth motion maps and texture operators and improves the performance of the KELM base classifiers.

1. *Initialization: The parameters are initialized as m=4, r=1, n=3, and* $w^{t=0} = (\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N})$.

2. *Input: The input sequences (depth) is used to calculate SI based on Eq. (1), on which 3DHoT_S, 3DHoT_M, and 3DHoT_C features are extracted.*

3. *The decision outputs of three KELM classifiers are used to calculate* x_i $(C \times n)$ *as shown in section III, i = 1,...,N.*

4. *MBC is executed to combine KELMs into a strong classifier, in which we train* θ^1 *to* θ^C *in the* t^{th} *iteration:*

4.1. *Input* x_i *and target label* y_i.

4.2. *Solve the convex problem in Eq.(16) for each* θ^i *under the current weights.*

4.3. *Estimate the distribution of margin samples based on GMM (M=3). We first sort all the samples in decreasing order based on the decision output from current classifiers, and then half of samples are deployed to train GMM. For the* i^{th} *sample that satisfies GMM, we update* $w_i^{t+1} = w_i^t + 0.001$.

4.4. *Obtain* $\theta^{w^t} = (\theta^1, ..., \theta^C)$ *and calculate* $t = t + 1$. *Repeat step 4 until* $\| \theta^{w^t} - \theta^{w^{t-1}} \| \leq 0.01$ *or maximum iteration number reaches (i.e., 1000).*

5. *End*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

V. Experimental results

Our proposed system is implemented in MATLAB on an Intel i5 Quadcore 3.2 GHz desktop computer with 8GB of RAM. Separate algorithmic parts corresponding to our contributions as well as the entire action recognition system are evaluated and compared with state-of-the-art algorithms based on four public datasets including MSRAction3D [44], MSRGesture3D [44], MSRActivity3D [44] and UTD-MHAD [45]. Moreover, we conduct the experiments to investigate the effects of a few important parameters. For all the experiments, we fix $m = 4$ and $r = 1$ based on our empirical studies in [10], [14], and the region size is set to 4×2 with 15 histogram bins when extracting 3DHoTs.

A. Datasets

The *MSRAction3D* dataset [44] is a popular depth dataset for action recognition, containing 20 actions performed by 10 subjects. Each subject performs one action 2 or 3 times when facing the depth camera. The resolution of each depth image is 240×320. It is a challenging dataset due to the similarity of actions and large speed variations in actions.

The *MSRGesture3D* dataset [44] is a benchmark dataset for depth-based hand gesture recognition, consisting of 12 gestures defined by American Sign Language (ASL). Each action is performed 2 or 3 times by each subject, thereby resulting in 333 depth sequences.

The *MSRActivity3D* dataset [44] contains 16 daily activities acquired by a Microsoft Kinect device. In this dataset, there are 10 subjects, each being asked to perform the same action twice in standing position and sitting position, respectively. There are in total 320 samples with both depth maps and RGB sequences.

The *UTD-MHAD* dataset [45] employed four temporally synchronized data modalities for data acquisition. It provides RGB videos, depth videos, skeleton positions, and inertial signals (captured by a Kinect camera and a wearable inertial sensor) of a comprehensive set of 27 human actions. Some example frames of the datasets are shown in Fig. 4.

B. Contribution verification

We have claimed two contributions in Section I, which are a new multi-class boosting classifier and an improved feature descriptor. Here, we design an experiment to verify these two contributions simultaneously on the MSRAction3D dataset. More specifically, we have combined two different feature descriptors and four different classifier fusion methods for the action recognition. Feature descriptors include our 3DHoTs descriptor and the conventional DMM+LBP descriptor [9] while the four classifier fusion methods involve AdaBoost.M2 [36], LOGP [9], MCBoost [39] and our MBC. The idea is to feed two features into four classifiers respectively, and afterwards, the average recognition accuracy of each combination is calculated accordingly.

Table I shows the achieved results, for which we adopted the original settings suggested in [9]. If we look at each column vertically, we can find the accuracy comparisons when fixing the classifier but varying feature descriptors. As can be seen, our 3DHoTs feature is consistently better than the DMM+LBP feature over four classifiers, indicating that applying the CLBP descriptor on DMM maps indeed helps to represent the action. On the contrary, if we look at each row horizontally, we can find the results achieved by different classifiers when the input feature is constant. It is clear that our MBC classifier performs better than the other three, regardless of the input features. Compared with AdaBoost.M2 [36], MBC achieves a much better performance due to the fact that our framework focuses on the margin samples that can be more robust when the size of the sample set is not large, which is the case in this application.

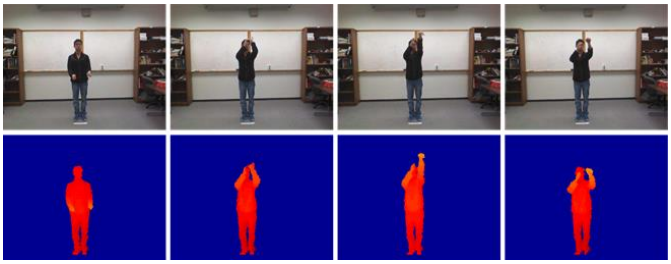

Fig. 4. An example of basketball-shoot action from UTD-MHAD dataset. The first row shows the color images, the second row shows the depth images.

TABLE I
Recognition Accuracy (%) of different feature and classifier combinations on MSRAction3D Dataset

	Adaboost.M2 [36]	LOGP [9]	MCBoost [39]	MBC
DMM+LBP [9]	87.55	93.04	94.51	94.51
3DHoTs	93.77	94.87	94.87	95.24

As is shown in Table II and Table III, our 3DHoTs feature outperforms DMM+LBP feature over four classifiers, which indicates that the CLBP descriptor on DMM maps make a contribution to recognizing different actions. Furthermore, in each row respectively, it is demonstrated that our MBC classifier achieves comparable results with other classifier combination methods.

In comparison with Adaboost.M2 and MCBoost, our MBC method performs better in both MSRGesture3D dataset and UTD-MHAD dataset. In fact, multiclass boosting method cannot be directly used in our problems. We addressed the issue by combining heterogeneous classification models, which is not a custom classification task. To compare with multi-class boosting methods, in a different way, we substituted our objective function with the loss function they defined for M-array classification.

TABLE II
Recognition Accuracy (%) of different feature and classifier combinations on MSRGesture3D Dataset

	Adaboost.M2 [36]	LOGP [9]	MCBoost [39]	MBC
DMM+LBP [9]	92.7	94.6	93.6	94.4
3DHoTs	93.6	94.7	94.0	94.7

TABLE III
Recognition Accuracy (%) of different feature and classifier combinations on UTD-MHAD Dataset

	Adaboost.M2 [36]	LOGP [9]	MCBoost [39]	MBC
DMM+LBP [9]	81.9	82.3	83.0	83.7
3DHoTs	83.0	83.3	83.7	84.4

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2718189, IEEE Transactions on Image Processing

TABLE V
COMPARISON OF RECOGNITION ACCURACIES (%) OF OUR METHOD AND EXISTING METHODS ON MSRACTION3D DATASET USING SETTING 1

Method	Test one				Test two				Cross subject			
	AS1	AS2	AS3	Average	AS1	AS2	AS3	Average	AS1	AS2	AS3	Average
Li *et al.* [11]	89.5	89.0	96.3	91.6	93.4	92.9	96.3	94.2	72.9	71.9	79.2	74.7
DMM-HOG [13]	97.3	92.2	98.0	95.8	98.7	94.	98.7	97.4	96.2	84.1	94.6	91.6
HOJ3D [47]	**98.5**	96.7	93.5	96.2	98.6	97.2	94.9	97.2	88.0	85.5	63.6	79.0
Chaaraoui *et al.* [55]	-	-	-	-	-	-	-	-	91.6	90.8	97.3	93.2
Vemulapalli *et al.* [52]	-	-	-	-	-	-	-	-	95.3	83.9	98.2	92.5
DMM-LBP-FF [9]	96.7	**100**	99.3	98.7	**100**	**100**	**100**	**100**	98.1	92.0	94.6	94.9
DMM-LBP-DF [9]	98.0	97.4	99.3	98.2	**100**	**100**	**100**	**100**	**99.1**	92.9	92.0	94.7
Occupancy Patterns [45]	98.2	94.8	97.4	96.8	99.1	97.0	98.7	98.3	91.7	72.2	**98.6**	87.5
3DHoT-MBC	98.0	98.7	**100**	**98.9**	**100**	**100**	**100**	**100**	**99.1**	**93.8**	98.2	**97.0**

C. System verification

1) Results on the MSRAction3D dataset

Similar to other publications, we establish two different experimental settings to evaluate our method.

Setting 1 - The experimental setting reported in [11] is adopted. Specifically, the actions are divided into three subsets as listed in Table IV. For each subset, three different tests are carried out. In the first test, 1/3 of the samples are used for training and the rest for testing; in the second test, 2/3 of the samples are used for training and the rest for testing; in the cross-subject test, one half of the subjects (1, 3, 5, 7, 9) are used for training and the rest for testing.

TABLE IV
THREE SUBSETS OF ACTIONS USED FOR MSRACTION3D DATASET

Action set 1 (AS1)	**Action set 2 (AS2)**	**Action set 3 (AS3)**
Horizontal wave (2)	High wave (1)	High throw (6)
Hammer (3)	Hand catch (4)	Forward kick (14)
Forward punch (5)	Draw x (7)	Side kick (15)
High throw (6)	Draw tick (8)	Jogging (16)
Hand clap (10)	Draw circle (9)	Tennis swing (17)
Bend (13)	Two hand wave (11)	Tennis serve (18)
Tennis serve (18)	Forward kick (14)	Golf swing (19)
Pickup throw (20)	Side boxing (12)	Pickup throw (20)

Setting 2 - The experimental setup suggested by [46] is used. A total of 20 actions are employed and one half of the subjects (1, 3, 5, 7, 9) are used for training and the remaining subjects are used for testing.

To facilitate a fair comparison, we set the same parameters of DMMs and blocks as noted in [9]. As illustrated in Table V, the results clearly validate the effectiveness of *MBC*. In the test one, our method achieves 100% recognition accuracy in AS3, and also comparable results in AS1 and AS2. In the second experiment, our method gets 100% recognition accuracy on all three subsets. In the cross-subject test, the *MBC* method again gets the highest average recognition accuracy, in this very challenging setting with large inter-class variations of different training and testing subjects.

The comparison results of setting 2 are illustrated in Table VI, showing that our approach performs the best in terms of the recognition accuracy. More specifically, the ensemble MBC classifier significantly improves the performance of single 3DHoT feature, i.e., 3DHoT_S, at least 3.3%. Compared to the state-of-the-art algorithm (DMM-LBP-DF) that is also based on the decision-level fusion scheme, we are 2% higher in terms of the accuracy rate. With respect to the feature extraction, we compare ours with most of existing descriptors, i.e., DMM [9], Cuboid [47], and our method consistently shows its advantages in the database. In terms of classifier, MBC achieves a much better performance than SVM [13] [48] and ELM [9]. Note that all compared results are cited from reference papers.

2) Results on the MSRGesture3D dataset

Table VII shows the recognition results of our method as well as comparative methods on the MSRGesture3D dataset. As shown in this Table, the proposed method achieves a much better performance than DMM-HOG with an increased rate of 5.5%. The accuracy of the decision level fusion approach (DMM-LBP-DF) is similar to ours, and both methods outperform the others. It should be noted that the AdaBoost.M2 [36] is not suitable for a small set of training samples, which are not used for the comparison in this experiment.

3) Results on the UTD-MHAD dataset

In the conducted experiments, we only utilize the depth data. Subsequently, the data from the subject numbers 1, 3, 5, 7 are used for training, and the data for the subject numbers 2, 4, 6, 8 are used for testing. Note that we slightly change the parameter m to 6 for 3DHoTs feature extraction due to the better performance on this dataset.

We have compared our method with the existing feature extraction methods [45] used for depth images and inertial sensors. It is remarkable that *MBC* obtains a much better performance than the combination of Kinect and Inertial as shown in Table VIII. Compared to the state-of-the-art DMM-HOG result, we obtain 2.9% higher recognition accuracy. The results clearly demonstrate the superior performance of our method. Compared to the traditional multi-class AdaBoost, we again achieve a much better performance, which further validates the effectiveness of MBC.

4) Results on the MSRActivity3D dataset

To further test the effectiveness of the proposed method, we consider a more complicated situation, i.e., human activity recognition. We conduct an experiment on the MSRActivity3D dataset, which is more challenging due to the large intra-class variations occurring in the dataset. Experiments performed on this dataset is based on a cross-subject test by following the same setting in [12], where 5 subjects are used for training, and the remaining 5 subjects are used for testing. The AdaBoost.M2

[36] is not used on this dataset, because the data set is not big enough to well train an ensemble classifier like it.

TABLE VI
RECOGNITION ACCURACY (%) COMPARED WITH EXISTING METHODS ON MSRACTION3D DATASET

Method	Accuracy (%)
DMM-HOG [13]	85.5
Random Occupancy Patterns [48]	86.5
DMM-LBP-FF [9]	91.9
DMM-LBP-DF [9]	93.0
HON4D [49]	88.9
Actionlet Ensemble [12]	88.2
Depth Cuboid [50]	89.3
Rahmani *et al.* [51]	88.8
Vemulapalli *et al*. [52]	89.5
3DHoT_S	91.9
3DHoT_M	88.3
3DHoT_C	86.4
3DHoT-MBC	**95.2**

TABLE VII
RECOGNITION ACCURACY (%) COMPARED WITH EXISTING METHODS ON MSRGESTURE3D DATASET

Method	Accuracy (%)
Random Occupancy Patterns [48]	88.5
HON4D [49]	92.5
Rahmani *et al.* [51]	93.6
DMM-HOG [13]	89.2
DMM-LBP-FF [9]	93.4
DMM-LBP-DF [9]	94.6
Edge Enhanced DMM [53]	90.5
Kurakin *et al.* [54]	87.7
3DHoT-MBC	**94.7**

TABLE VIII
RECOGNITION ACCURACY (%) COMPARED WITH EXISTING METHODS ON UTD-MHAD DATASET

Method	Accuracy (%)
Kinect [45]	66.1
Inertial [45]	67.2
Kinect&Inertial [45]	79.1
DMM-HOG [13]	81.5
Adaboost.M2 [36]	83.0
3DHOT-MBC	**84.4**

Seen from the results reported in Table IX, our algorithm outperforms all the prior arts including several recent ones except for [22]. It reveals that our MBC framework indeed works well even if feeding two different types of features. The major reason that our performance is worse than that of [22] lies in the fact that we are mainly based on the depth features extracted from the raw depth signal but the work in [22] employs more sophisticated skeleton-based features, which can better interpret the human actions when a challenging dataset is given. Though we have integrated the skeleton information here in order to verify whether our multi-class boosting framework can handle two different types of features, our skeleton features encoding only the joint position differences are very simple, in contrast to [22] that uses group sparsity and geometry constrained dictionary learning to further enhance the skeleton feature representation. According to their results, the classification performance benefits from generalizing vector quantization (e.g., Bag-of-Words representation) to sparse coding [22]. It is believed that our performance can be improved further if we could combine the sophisticated skeleton features.

TABLE IX
RECOGNITION ACCURACY (%) COMPARED WITH EXISTING METHODS ON MSRACTIVITY3D DATASET

Method	Accuracy (%)
Only Joint Position features [12]	68.0
RIA-SST [58]	70.0
Moving Pose [59]	73.8
HON4D [49]	80.0
Depth cobiod similarity feature [50]	83.6
Actionlet ensemble [12]	85.8
CoDe4D LST [57]	86.0
SNV [56]	86.25
DL-GSGC [22]	95.0
3DHoT+joint MBC	**89.4**

5) Comparison with deep learning based methods

The baseline methods mentioned above deploy the traditional handcrafted features. Differently, the deep learning models learn the feature representation from raw data and generate the high level semantic representation [26], [27] which represent the latest development in action recognition. Here, we compare our method with two deep models, in which one is SMF-BDL [26] and the other one is a DMM-Pyramid approach based on both traditional 2D *CNN* and 3D *CNN* for action recognition. Similar to *MBC*, the decision-level fusion method is used to combine different deep *CNN* models. To validate the proposed *3DHoT-MBC* method, we conduct the same experiment as those of the two methods. Note that the comparative results are all reported on their reference papers. The results in Table X and Table XI show that *3DHoT-MBC* is even superior to the two deep learning methods

TABLE X
RECOGNITION ACCURACIES (%) OF OUR METHOD AND DEEP LEARNING METHODS ON MSRACTION3D DATASET USING SETTING 1

Method	Test1(average)	Test2(average)	Test3(average)
MS [26]	93.6	94.3	86.3
SMF [26]	96.7	98.7	89.1
BDL [26]	94.1	95.6	87.6
SMF-BDL [26]	97.3	99.1	90.8
3DHoT-MBC	**98.9**	**100**	**97.0**

TABLE XI
RECOGNITION ACCURACIES (%) OF OUR METHOD AND DEEP LEARNING METHODS ON MSRACTION3D DATASET USING SETTING 2 AND MSRGESTURE3D DATASET

Method	MSRAction3D	MSRGesture3D
2D-CNN [27]	91.21	94.35
3D-CNN [27]	86.08	92.25
3DHoT-MBC	**95.2**	**94.7**

D. Comparison with other boosting methods

In this section, we create a large-scale action database by combining two action databases, MSR Action3D and UTD-MHAD, into a single one. We then compare

performances of different boosting algorithms for two kinds of features, i.e., DMM+LBP and 3DHoTs. The new combined Action-MHAD dataset has 38 distinct action categories (the same actions in both datasets are combined into one action) which consist of 1418 depth sequences. In experiments, odd subject numbers such as 1, 3, 5, 7 are used for training and the remaining subjects are used for testing. The experimental results, as shown in Table XII, demonstrate that our MBC is superior to other boosting methods.

TABLE XII
RECOGNITION ACCURACY (%) OF DIFFERENT FEATURE AND CLASSIFIER COMBINATIONS ON ACTION -MHAD DATASET

	Adaboost.M2 [36]	LOGP [9]	MCBoost [39]	Shen *et al.* [40]	Gentle Boost [35]	MBC
DMM+LBP [9]	84.09	86.47	87.47	86.04	83.66	88.90
3DHoTs	87.18	87.79	88.04	86.37	86.90	89.61

We also verify our algorithm on the DHA dataset [61]. DHA contains 23 action categories where the first 10 categories follow the same definitions in the Weizmann action dataset [65] and the 11th to 16th actions are extended categories. The 17th to 23rd are the categories of selected sport actions. Each of the 23 actions was performed by 21 different individuals (12 males and 9 females), resulting in 483 action samples. Table XIII shows the recognition results of our method against existing algorithms on the DHA dataset. Again, our method achieves the best recognition performance.

TABLE XIII
RECOGNITION ACCURACY (%) COMPARED WITH EXISTING METHODS ON DHA DATASET

Method	**Accuracy (%)**
D-STV/ASM [61]	86.80
SDM-BSM [62]	89.50
DMM-LBP-DF [9]	91.30
D-DMHI-PHOG [63]	92.40
DMPP-PHOG [63]	95.00
DMMs-FV [64]	95.44
3DHoT-MBC	**96.69**

E. Effects of parameters

Like other action recognition systems, our system also needs to tune a few parameters in both the 3DHoTs feature extraction stage and the MBC classification stage so as to obtain the best performance. Regarding feature extraction, the selections of m and r is critical, which determine the region size on DMM and also the number of the neighboring points involved in the descriptor. In our previous papers [9], [14], we accomplished an empirical study for these two parameters, which revealed m = 4 and r = 1 can obtain good results on most of the datasets.

With respect to our classification algorithm, there are two parts involving KELM base classifier and the MBC fusion algorithm. For the KELM, there is a regularization term ρ that is used to solve ill-posed problem. In Fig. 5, we plot the recognition accuracy changes of our method (training data cross validation) if we vary this parameter on the MSRAction3D dataset. Seen from the curve, it is very obvious that we could set this parameter to 1000 because the recognition rate reaches a peak point when adopting that value.

For the MBC, regularization coefficient λ is the only parameter required to be predefined. Here, we investigate how the algorithm will behave when varying λ. To do so, we change the value of λ and plot the corresponding recognition rates on two datasets, which are illustrated in Fig. 6. As shown on this figure, the MBC recognition accuracy is oscillating when λ is varying between 0 and 50. When λ exceeds 50, MBC results increase gradually and finally level off until λ reaches 100. We find more or less the same behavior on two different datasets, which makes the selection of this parameter feasible. In fact, the regularization term reflects our selected model complexity. When we set a small λ, we actually set a loose constraint of model complexity, which will easily lead to overfitting. On the other hand, a large λ ensures that we obtain a simple model. So, we set $\lambda = 100$ considering a tradeoff between algorithm performance and efficiency.

Finally, the execution time of our system is calculated, intending to reveal the feasibility of our system for a real-time application. To this end, we have set up a simulation platform using MATLAB on an Intel i5 Quadcore 3.2 GHz desktop computer with 8GB of RAM. It can be seen that the proposed method is able to process over *120* frames per second.

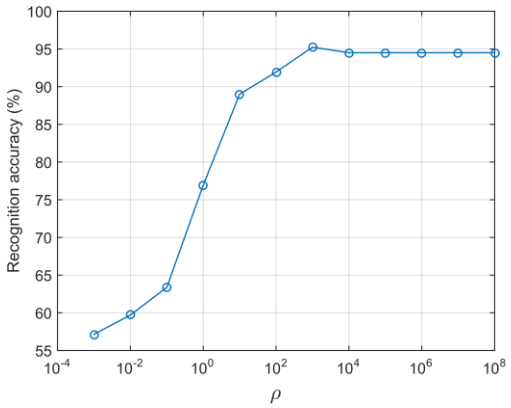

Fig. 5. KELM performance w.r.t. parameter ρ on the MSRAction3D dataset

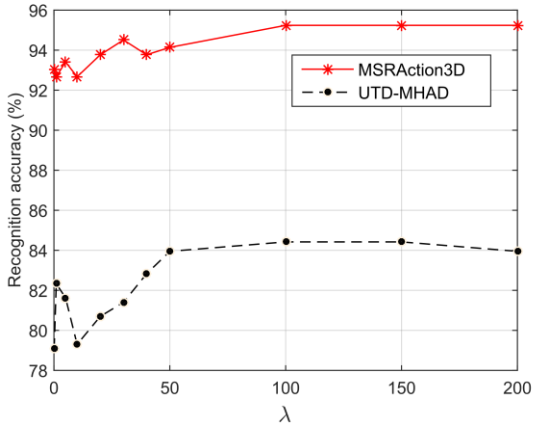

Fig. 6. System performance w.r.t. parameter λ on two datasets

VI. CONCLUSION

In this paper, we have proposed an effective feature descriptor and a novel decision-level fusion method for action recognition. This feature, called 3DHoTs, combines depth maps and texture description for an effective action representation of a depth video sequence. At the decision-level fusion, we have added the inequality constraints derived from a multi-class Support Vector Machine to modify the general AdaBoost optimization function, where Kernel-based extreme learning machine (KELM) classifiers serve as the base classifiers. The experimental results on four standard datasets demonstrate the superiority of our method. A future work is to extend this multi-class boosting framework to other relevant applications, such as object recognition [67] and image retrieval.

APPENDIX

Lemma 1: The GMM with 2 components is represented by $f(z,\mu_1,\sigma_1,\mu_2,\sigma_2)$ as:

$$f(z,\mu_1,\sigma_1,\mu_2,\sigma_2)=\omega_1 G_1(z,\mu_1,\sigma_1)+\omega_2 G_2(z,\mu_2,\sigma_2),$$

and we have:

$$f_{\sigma_z^2}(z,\mu_1,\sigma_1,\mu_2,\sigma_2)\le f_{\sigma^2}(z,0,\sigma_1,0,\sigma_2)+\varepsilon,$$

where ω_1, ω_2 are the mixture proportions, μ_1,μ_2 and σ_1,σ_2 are respectively the mean and variance of the Gaussian components, and ε is a constant. f_{σ^2} represents the variance of $f()$, with $0\le\mu_1,\mu_2\le 1, 0\le\varepsilon\le 1$.

Proof: Based on the definition of variance, we obtain:

$$\begin{aligned}f_{\sigma_z^2}&=\int_{-\infty}^{\infty}z^2(\omega_1G_1+\omega_2G_2)dz-\left(\int_{-\infty}^{\infty}z(\omega_1G_1+\omega_2G_2)dz\right)^2\\&=\omega_1\int_{-\infty}^{\infty}z^2G_1dz-\omega_1\left(\int_{-\infty}^{\infty}zG_1dz\right)^2+\omega_2\int_{-\infty}^{\infty}z^2G_2dz\\&-\omega_2\left(\int_{-\infty}^{\infty}zG_2dz\right)^2+\omega_1\left(\int_{-\infty}^{\infty}zG_1dz\right)^2-\omega_1^2\left(\int_{-\infty}^{\infty}zG_1dz\right)^2\\&+\omega_2\left(\int_{-\infty}^{\infty}zG_2dz\right)^2-\omega_2^2\left(\int_{-\infty}^{\infty}zG_2dz\right)^2-2\omega_1\omega_2u_1u_2\end{aligned}$$

As

$$\sigma_1^2=\int_{-\infty}^{\infty}z^2G_1dz-\left(\int_{-\infty}^{\infty}zG_1dz\right)^2$$

$$\sigma_2^2=\int_{-\infty}^{\infty}z^2G_2dz-\left(\int_{-\infty}^{\infty}zG_2dz\right)^2,$$

we obtain:

$$\begin{aligned}f_{\sigma_z^2}&=\int_{-\infty}^{\infty}z^2(\omega_1G_1+\omega_2G_2)dz-\left(\int_{-\infty}^{\infty}z(\omega_1G_1+\omega_2G_2)dz\right)^2\\&=\omega_1\sigma_1^2+\omega_2\sigma_2^2+\omega_1\omega_2\mu_1^2+\omega_1\omega_2\mu_2^2-2\omega_1\omega_2\mu_1\mu_2\end{aligned}$$

As $\omega_1+\omega_2=1$, we have:

$$\omega_1\omega_2\le 1/4,$$

and thus,

$$f_{\sigma_z^2}=\omega_1\sigma_1^2+\omega_2\sigma_2^2+\omega_1\omega_2(\mu_1-\mu_2)^2\le\omega_1\sigma_1^2+\omega_2\sigma_2^2+\frac{1}{4}(\mu_1-\mu_2)^2$$

and

$$f_{\sigma^2}(z,0,\sigma_1,0,\sigma_2)=\omega_1\sigma_1^2+\omega_2\sigma_2^2$$

As we constrain $0\le\mu_1,\mu_2\le 1$, and have: $0\le(\mu_1-\mu_2)^2\le 1$ and $\frac{1}{4}(\mu_1-\mu_2)^2\le\frac{1}{4}$. Thus, we obtain:

$$f_{\sigma_z^2}(z,u_1,\sigma_1,u_2,\sigma_2)\le f_{\sigma^2}(z,0,\sigma_1,0,\sigma_2)+\varepsilon$$

where ε is smaller than 0.25 in the case of $0\le\mu_1,\mu_2\le 1$.

Lemma 2: For GMM with *M* components, we have:

$f_{\sigma^2}(z,\mu_1,\sigma_1,\mu_2,\sigma_2,...)\le f_{\sigma^2}(z,0,\sigma_1,0,\sigma_2,...)+\varepsilon, 0\le\mu_1,\mu_2\le 1, 0\le\varepsilon\le 1$, when $M\le 4$.

Proof: We proven this Lemma from two different cases, when M is an even or odd number.

When M is an even number, based on Lemma 1, we have:

$$\begin{aligned}f_{\sigma_z^2}&=\int_{-\infty}^{\infty}z^2(\omega_1G_1+\omega_2G_2,...,+\omega_MG_M)dz-\left(\int_{-\infty}^{\infty}z(\omega_1G_1+\omega_2G_2,...,+\omega_MG_M)dz\right)^2\\&\le\omega_1\sigma_1^2+\omega_2\sigma_2^2,...,+\omega_M\sigma_M^2+\frac{1}{4}(\mu_1^2+\mu_2^2+...+\mu_{M-1}^2+\mu_M^2)\end{aligned}$$

As $0\le\mu_i\le 1, i=,...,M$, we have:

$$f_{\sigma_z^2}\le\omega_1\sigma_1^2+\omega_2\sigma_2^2,...,+\omega_M\sigma_M^2+\frac{\mathrm{M}}{4}.$$

We further prove Lemma 2 when *M* is an odd number, and have:

$$\begin{aligned}f_{\sigma_z^2}&=\int_{-\infty}^{\infty}z^2(\omega_1G_1+\omega_2G_2,...,+\omega_MG_M)dz-\left(\int_{-\infty}^{\infty}z(\omega_1G_1+\omega_2G_2,...,+\omega_MG_M)dz\right)^2\\&\le\omega_1\sigma_1^2+\omega_2\sigma_2^2,...,+\omega_M\sigma_M^2+\frac{1}{4}(\mu_1^2+\mu_2^2+...+\mu_{M-1}^2+\mu_M^2)\end{aligned}$$

and we obtain:

$$f_{\sigma_z^2}\le\omega_1\sigma_1^2+\omega_2\sigma_2^2,...,+\omega_M\sigma_M^2+\frac{\mathrm{M}}{4}$$

As

$$f_{\sigma^2}(z,0,\sigma_1,0,\sigma_2,...,)=\omega_1\sigma_1^2+\omega_2\sigma_2^2+,...,+\omega_M\sigma_{\mathrm{M}}^2$$

where $\varepsilon=\frac{M}{4}\le 1$. Lemma 2 is proved.

Theorem: Our objective (Eq. 16) maximizes the mean of margin, whilst minimizing the variance of margin, when the margin samples follow GMM ($M\le 4$).

Proof: We define $z_i'=\omega_i\exp(-y_iF(\theta^j,x_i))$. Here $0\le z_i'\le 1$ satisfying the conditions of Lemmas 1 and 2 is achieved by dividing a maximum value among z_i'. Minimizing $\sum_i z_i'$ leads to a similar result as that of Eq. (16), because $\log(.)$ (Eq. (16)) is a monotonically increasing function. Based on Lemma 2, if z' (margin) follows a GMM distribution, we have:

$$\sum_i(z_i'-u)^2\le\sum_i z_i'^2+\varepsilon,$$

where u is the mean. Using $0\le z_i'\le 1$ again, we have:

$$\sum_i z_i'^2 + \varepsilon \le \sum_i z_i' + \varepsilon ,$$

where ε is a given constant. $\sum_i z_i'$ (mean) is the upper bound of the variance $\sum_i (z_i' - u)^2$. Consequently, we conclude that our objective minimizes the variance of margin samples from a GMM distribution. In addition, $-y_i F(\theta^j, x_i^j)$ is defined based on [40] aiming to maximize the mean of margin, which is also propagated into our method. And so, the theorem is proved.

REFERENCES

[1] L. Zhao, X. Gao, D. Tao, and X. Li, "Tracking human pose using max-margin Markov models," *IEEE Trans. Image Proc.*, vol. 24, no. 12, pp. 5274–5287, 2015.

[2] C. Sun, I. Junejo, M. Tappen, and H. Foroosh, "Exploring sparseness and self-similarity for action recognition," *IEEE Trans. Image Proc.*, vol. 24, no. 8, pp. 2488–2501, 2015.

[3] Z. Zhang, and D. Tao, "Slow feature analysis for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 436–450, 2012.

[4] Y. Xu, D. Xu, S. Lin, T. Han, X. Cao, and X. Li, "Detection of sudden pedestrian crossings for driving assistance systems," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 3, pp. 729–739, Jun. 2012.

[5] A. Bobick, and J. Davis, "The recognition of human movement using temporal templates," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 3, pp. 257–267, 2001.

[6] A. Iosifidis, A. Tefas, and I. Pitas, "Multi-view action recognition based on action volumes, fuzzy distances and cluster discriminant analysis," Signal Processing, vol. 93, no. 6, pp. 1445–1457, 2013.

[7] I. Laptev, "On space-time interest points," Int. J. Comput. Vision, vol. 64, no. 2, pp. 107–123, 2005.

[8] A. A. Efros, E. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in Proc. IEEE Conf. Computer Vision, 2003, pp. 726–733.

[9] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," in Proc. IEEE Winter Conf. Applications of Computer Vision, 2015, pp. 1092–1099.

[10] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," IEEE Trans. Human-Machine Systems, vol. 45, no. 1, pp. 51–61, 2015.

[11] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshop, 2010, pp. 9–14.

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

[13] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in Proc. ACM Multimedia Conf., 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 Processing, vol. 12, no. 1, pp. 155–163, Aug. 2013.

[15] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," IEEE Trans. Image Proc., vol. 19, no. 6, pp. 1657–1663, 2010.

[16] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: a review," IEEE Trans. Cybern., vol. 43, no. 5, pp. 1318–1334, 2013.

[17] J. de Mesquita S. Junior, P. Cortez, and A. Backes, "Color Texture Classification Using Shortest Paths in Graphs," IEEE Trans. Image Proc., vol. 23, no. 9, pp. 3751–3761, 2014.

[18] A. Sharaf, M. Torki, M. E. Hussein, and M. El-Saban, "Real-time multi-scale action detection from 3D skeleton data," in Proc. IEEE Winter Conf. Applications of Computer Vision, 2015, pp. 998–1005.

[19] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," J. Visual Communication and Image Representation, vol. 25, no. 1, pp. 24–38, 2014.

[20] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, "Histogram of oriented displacements (hod): describing trajectories of human joints for action recognition," in Proc. Int. Joint Conf. Artificial Intell., 2013, pp. 1351–1357.

[21] H. Chen, Y. Chen, and S. Lee, "Human action recognition using star skeleton," in Proc. ACM Int. Workshop Video Surveillance and Sensor Networks, 2006, pp. 171–178.

[22] J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in Proc. IEEE Int. Conf. Computer Vision, 2013, pp. 1809–1816.

[23] S. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: real-time action recognition," *J. Machine Learning Research*, vol. 14, no. 1, pp. 2617–2640, 2013.

[24] T. Tran, and V. Nguyen, "How good is kernel descriptor on depth motion map for action recognition," in Proc. Int. Conf. Computer Vision Systems. 2015, pp. 137–146.

[25] C. Zhang, and Y. Tian, "Histogram of 3D facets: A depth descriptor for human action and hand gesture recognition," Computer Vision and Image Understanding, vol. 139, no. 1, pp. 29–39, 2015.

[26] L. Wang, B. Zhang, and W. Yang, "Boosting-like deep convolutional network for pedestrian detection", in *Proc. Chin. Conf. Biometric Recognition*, 2015, pp. 581–588.

[27] R. Yang, and R. Yang, "DMM-pyramid based deep architectures for action recognition with depth cameras," in *Proc. Asian Conf. Computer Vision*, 2014, pp. 37–49.

[28] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Trans. Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 1403–1416, 2015.

[29] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, 2012.

[30] T. Zhang, J. Liu, S. Liu, C. Xu, and H. Lu, "Boosted exemplar learning for action recognition and annotation," IEEE Trans. Circuits and Systems for Video Technology, vol. 21, no. 7, pp. 853–866, July. 2011.

[31] E. Allwein, R. Schapire, Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," J. Machine Learning Research, vol. 1, no. 12, pp. 113–141, 2000.

[32] C. Shen, Z. Hao, "A direct formulation for totally-corrective multi-class boosting," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2011, pp. 2585–2592.

[33] A. Fernandez-Baldera, L. Baumela, "Multi-class boosting with asymmetric binary weak-learners," Pattern Recognition, vol. 47, no. 5, pp. 2080–2090, 2014.

[34] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," Machine Learning, vol. 37, no. 3, pp. 297–336, Dec. 1999.

[35] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," The Annals of Statistics, vol. 28, no. 2, pp. 337–407, 2000.

[36] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[37] J. Zhu, H. Zou, S. Rosset and T. Hastie, "Multi-class Adaboost," Statistics and Its Interface, vol. 2, no. 1, pp. 349–360, 2009.

[38] A. Reiss, G. Hendeby, and D. Stricker, "A novel confidence-based multiclass boosting algorithm for mobile physical activity monitoring," Personal and Ubiquitous Computing, vol. 19, no. 1, pp. 105–121, Jan. 2015.

[39] M. Saberian, N. Vasconcelos, "Multiclass boosting: Theory and algorithms," in Advances in Neural Info. Processing Sys., 2011, pp. 2124–2132.

[40] C. Shen, and H. Li, "On the dual formulation of boosting algorithms," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 12, pp. 2216–2231, 2010.

[41] Y. Freund, and R. Schapire, "Experiments with a New Boosting Algorithm," in Proc. Int. Conf. Machine Learning, 1996, pp. 148–156.

[42] M. Collins, R. Schapire, and Y. Singer, "Logistic regression, AdaBoost and bregman distances," Machine Learning, vol. 48, no. 1, pp.253–285, 2002.

[43] K. Crammer, and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," J. Machine Learning Research, vol. 2, no. 2, pp. 265–292, 2001.

[44] Microsoft RGB-D datasets: http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/default.htm
[45] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in Proc. IEEE Int. Conf. Image Processing, 2015.
[46] A. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and M. Campos, "On the improvement of human action recognition from depth map sequences using space-time occupancy patterns," Pattern Recognition Letters, vol. 36, no. 1, pp. 221–227, 2014.
[47] L. Xia, C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshop, 2012, pp. 20–27.
[48] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in Proc. Eur. Conf. Computer Vision, 2012, pp. 872–885.
[49] O. Oreifej, and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2013, pp. 716–723.
[50] L. Xia, and J. Aggarwal. "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2013, pp. 2834–2841.
[51] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Real time action recognition using histograms of depth gradients and random decision forests," in Proc. IEEE Winter Conf. Applications of Computer Vision, 2014, pp. 626–633.
[52] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d human skeletons as points in a lie group," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2014, pp. 588–595.
[53] C. Zhang, and Y. Tian, "Edge enhanced depth motion map for dynamic hand gesture recognition," in Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshop, 2013, pp. 500–505.
[54] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in Proc. Eur. Signal Processing Conf., 2012, pp. 1975–1979.
[55] A. Chaaraoui, J. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta, "Evolutionary joint selection to improve human action recognition with RGB-D devices," Expert Systems with Applications, vol. 41, no. 3, pp. 786–794, 2014.
[56] X. Yang, and Y. Tian, "Super normal vector for activity recognition using depth sequences." in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2014, pp. 804–811.
[57] H. Zhang, and L. Parker, "CoDe4D: color-depth local spatio-temporal features for human activity recognition from RGB-D videos," IEEE Trans. Circuits and Systems for Video Technology, vol. 26, no. 3, pp. 541–555, 2016.
[58] B. Amor, J. Su, and A. Srivastava, "Action recognition using rate-invariant analysis of skeletal shape trajectories," IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 1, pp. 1–13, 2016.
[59] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in Proc. IEEE Int. Conf. Computer Vision, 2013, pp. 2752–2759.
[60] Y. Yang, B. Zhang, L. Yang, C. Chen, W. Yang. "Action recognition using completed local binary patterns and multiple-class boosting classifier" in Proc. Asian Conf. on Pattern Recognition (ACPR), 2015, pp. 336-340.
[61] 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 *ACM MM*, pp. 1053–1056, 2012.
[62] H. Liu, L. Tian, and M. Liu, "Sdm-bsm: A fusing depth scheme for human action recognition," in *ICIP*, pp. 4674–4678, 2015.
[63] 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, 2015
[64] 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. Int. Joint Conf. Artificial Intell., 2016, pp. 3331–3337.
[65] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, "Actions as space-time shapes," in Proceedings of the Tenth IEEE International Conference on Computer Vision, Beijing, 2005, pp. 1395-1402.
[66] B. Chen, J. Yang, B. Jeon, X. Zhang, "Kernel quaternion principal component analysis and its application in RGB-D object recognition," Neurocomputing, 10.1016/j.neucom.2017.05.047.
[67] B. Zhang, Y. Gao, S. Zhao and J. Liu, "Local Derivative Pattern versus Local Binary Pattern: Face Recognition with High-Order Local Pattern Descriptor", IEEE Transactions on Image Processing, Vol. 19, No. 2, pp. 533-544, 2010.

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, Australia. Currently, he is an associate professor with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. He held a senior postdoc position in PAVIS department, IIT, Italy. He was supported by the Program for New Century Excellent Talents in University of Ministry of Education of China. His current research interests include pattern recognition, machine learning, face recognition, and wavelets.

Yun Yang received the B.E. and M.S. degrees in Automation science and electrical engineering department of Beihang University, Beijing, China, in 2014 and 2017, respectively. Now he is a computer vision engineer working at Noah's Ark Lab in Huawei Technologies. His research focuses on human action recognition, face recognition and pedestrian re-identification.

Chen Chen received the B.E. degree in automation from Beijing Forestry University, Beijing, China, in 2009 and the MS degree in electrical engineering from Mississippi State University, Starkville, in 2012 and the PhD degree in the Department of Electrical Engineering at the University of Texas at Dallas, Richardson, TX in 2016. He is currently a Post-Doc in the Center for Research in Computer Vision (CRCV) at University of Central Florida (UCF). His research interests include compressed sensing, signal and image processing, pattern recognition and computer vision. He has published over 50 papers in refereed journals and conferences in these areas.

Linlin Yang received the B.S. and M.S. degrees in automation from Beihang University. His current research interests include signal and image processing, pattern recognition and computer vision.

Jungong Han is working in Lancaster University, UK. Previously, he was with the Northumbria University (2015-2017), UK, was with Philips CI (2012-2015), was with the Centre for Mathematics and Computer Science (2010-2012), was with the Technical University of Eindhoven (2005-2010) in Netherlands. Dr. Hans research interests include multimodality data fusion, computer vision, and artificial intelligence. He is an associate editor of Elsevier Neurocomputing and Springer Multimedia Tools and Applications.

Ling Shao (M09-SM10) is a professor with the School of Computing Sciences at the University of East Anglia, Norwich, UK. His research interests include computer vision, image/video processing and machine learning. He is an associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems and several other journals. He is a Fellow of the British Computer Society and the Institution of Engineering and Technology.