Multimed Tools Appl
DOI 10.1007/s11042-016-3284-7

Action recognition from depth sequences using weighted fusion of 2D and 3D auto-correlation of gradients features

Chen Chen[1] · **Baochang Zhang**[2] · **Zhenjie Hou**[3] · **Junjun Jiang**[4] · **Mengyuan Liu**[5] · **Yun Yang**[2]

Received: 3 November 2015 / Revised: 26 December 2015 / Accepted: 17 January 2016

Abstract This paper presents a new framework for human action recognition from depth sequences. An effective depth feature representation is developed based on the fusion of 2D and 3D auto-correlation of gradients features. Specifically, depth motion maps (DMMs) are first employed to transform a depth sequence into three images capturing shape and motion cues. A feature extraction method utilizing spatial and orientational auto-correlations of image local gradients is introduced to extract features from DMMs. Space-time auto-correlation of gradients features are also extracted from depth sequences as complementary features to cope with the temporal information loss in the DMMs generation. Each set of features is used as

✉ Baochang Zhang
bczhang@buaa.edu.cn

✉ Zhenjie Hou
houzj@cczu.edu.cn

Chen Chen
chenchen870713@gmail.com

Junjun Jiang
junjun0595@163.com

Mengyuan Liu
liumengyuan@pku.edu.cn

Yun Yang
bhu_yunyang@163.com

[1] Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080, USA

[2] School of Automation Science and Electrical Engineering, Beihang University, Beijing, China

[3] School of Information Science & Engineering, Changzhou University, Changzhou, China

[4] School of Computer Science, China University of Geosciences, Wuhan, China

[5] Engineering Laboratory on Intelligent Perception for Internet of Things (ELIP), Peking University, Shenzhen Graduate School, Shenzhen, China

Published online: 26 January 2016

input to two extreme learning machine classifiers to generate probability outputs. A weighted fusion strategy is proposed to assign different weights to the classifier probability outputs associated with different features, thereby providing more flexibility in the final decision making. The proposed method is evaluated on two depth action datasets (MSR Action 3D and MSR Gesture 3D) and obtains the state-of-the-art recognition performance (94.87 % for the MSR Action 3D and 98.50 % for the MSR Gesture 3D).

Keywords Action recognition · Depth data · Depth motion maps · Gradient local auto-correlations · Space-time auto-correlation of gradients · Extreme learning machine · Weighted fusion

1 Introduction

Human action recognition is an active research area benefitting many applications. Example applications include surveillance systems, video analytics, physical rehabilitation, robotics, and human computer interaction [3,5,6,14,15,23,37]. Research on human action recognition has made significant progress in the last decade. Earlier attempts at action recognition have mainly focused on learning and recognizing activities from conventional RGB cameras. As noted in [1], there are limitations associated with the utilization of RGB cameras for action recognition. In practice, one requires to have a considerable amount of hardware resources in order to run computationally intensive image processing and computer vision algorithms and also one needs to deal with a lack of 3D action data in conventional images.

Recent emergence of cost-effective depth sensors (in particular, Microsoft Kinect and Asus Xtion Pro) has led to their widespread utilization for human action recognition. Compared to conventional RGB images captured by video cameras, depth images generated by depth cameras are shown to be insensitive to lighting changes and provide body shape and structure information for action recognition. In addition, depth images can significantly simplify tasks such as background subtraction and segmentation. The human skeleton information can also be obtained from depth images [30].

Prior works Research on human action recognition from depth images has explored various representations include a bag of 3D points [33], projected depth maps [4,7,36], spatio-temporal depth cuboid [24], occupancy patterns [32], surface normals [26,35], and skeleton joints [12,29,34]. Here we review some developed major feature representation techniques for human action recognition based on depth sequences.

In [33], a bag of 3D points was sampled from depth images to describe the 3D shapes of salient postures and an action graph model was employed to model the dynamics of the actions. In [17], a local occupancy patterns (LOP) feature computes the local occupancy information based on the 3D point cloud around a particular joint was proposed for action recognition from depth sequences. The temporal dynamics of the occupancy patterns can roughly discriminate different types of interactions. To transform the action recognition problem in 3D to 2D, depth images in a depth video sequences were projected onto three orthogonal planes and differences between projected depth maps were accumulated to form three 2D depth motion maps (DMMs) [36]. Histogram of oriented gradients (HOG) [11] features were then extracted from DMMs as global representations of a depth video. In [7], the procedure of generating DMMs was modified to reduce the computational complexity in order to achieve real-time action recognition. Later in [4], local binary pattern [25,38,39] operator was applied to the overlapped blocks in DMMs to enhance the discriminative power for action recognition. A filtering method to extract spatio-temporal interest points (STIPs) from depth videos (called DSTIP) was introduced in [24] to localize activity related interest points by effectively suppressing the noise in the depth videos. Depth cuboid similarity feature (DCSF) built around the DSTIPs was proposed to describe the local 3D depth cuboid. In [26], histogram of the surface normal orientation in the 4D space of time, depth, and spatial coordinates (HON4D) was developed to capture the complex joint shape-motion cues at pixel-level of depth images. Due to the effectiveness of the surface normals characterizing the local motion and shape information simultaneously, the polynormal feature descriptor was introduced in [35] by clustering hypersurface normals in a depth sequence. An adaptive spatio-temporal pyramid was also proposed to globally capture the spatial and temporal orders of the polynormals.

Skeleton information which can be considered as high level features extracted from depth images has also been explored for action recognition. In [34], a skeleton feature descriptor named EigenJoints was developed based on differences of skeleton joints. The EigenJoints feature can effectively combine action information including static posture, motion property, and overall dynamics. In [12], a local skeleton descriptor that encodes the relative position of joint quadruples was presented. The similarity normalization transform in the coding strategy makes the descriptor scale, viewpoint and body-orientation invariant. In [29], a new skeletal representation that explicitly models the 3D geometric relationships between various body parts using rotations and translations in 3D space was proposed. The relative geometry between a pair of body parts was mathematically presented as a point in a special Euclidean group. Therefore, the entire human skeleton was modeled as a point in a Lie group. The action recognition was then performed by classifying these curves.

Motivation and contributions In our previous work [2], we introduced the gradient local auto-correlations (GLAC) [18] descriptor and applied GLAC to the three DMMs of a depth sequence to generate the feature (i.e., DMMs-based GLAC feature) for action recognition. The GLAC descriptor utilizes spatial and orientational auto-collections (i.e., second order statistics) of local gradients to capture richer information from images than the histogram-based methods (e.g., HOG) which use first order statistics (i.e., histograms). Although DMMs obtained using all depth frames in a depth sequence can describe the shape and motion cues of a depth action sequence, they may not be able to capture the detailed (or local) temporal motion in a subset of depth images. Old motion history may get overwritten when a more recent action occurs at the same point. Therefore, in the paper we further incorporate another feature extraction method which exploits the local relationships (co-occurrence) among space-time gradients in the space and time domain. The resulting space-time motion features are named space-time auto-correlation of gradients (STACOG) [19]. The STACOG feature is an extension of GLAC in 3D space (i.e., space and time) and was designed for RGB video sequences in [19]. A weighted fusion framework based on extreme learning machine (ELM) [16] is proposed to effectively combine the GLAC features from DMMs and STACOG features for action recognition. We extensively evaluate our method on standard depth sequence datasets, MSR Action 3D [33] and MSR Gesture 3D [21], and achieve superior performance over state-of-the-art methods.

The remainder of the paper is organized as follows. Section 2 presents the feature extraction methods including DMMs-based GLAC and STACOG. Section 3 describes the proposed weighted fusion scheme based on ELM. Section 4 reports action recognition results on two benchmark datasets, comparing to a variety of other state-of-the-art methods. Finally, Section 5 concludes the paper.

2 Feature extraction

2.1 Depth motion maps-based gradient local auto-correlations

GLAC [18] descriptor is an effective tool for extracting shift-invariant image features. Let I be an image region and $\mathbf{r}=(x,y)^t$ be a position vector in I. The magnitude and the orientation angle of the image gradient at each pixel can be represented by $z=\sqrt{\frac{\partial I^2}{\partial x}+\frac{\partial I^2}{\partial y}}$ and $\theta=\arctan\left(\frac{\partial I}{\partial x},\frac{\partial I}{\partial y}\right)$, respectively. The orientation θ is then coded into D orientation bins by voting weights to the nearest bins to form a gradient orientation vector $\mathbf{f}\in\mathbb{R}^D$. With the gradient orientation vector f and the gradient magnitude z, the N^{th} order auto-correlation function of local gradients can be expressed as follows:

$$R\left(d_0,\ldots,d_{N,}\mathbf{a}_1,\ldots,\mathbf{a}_N\right)=\int_I \omega[z(\mathbf{r}),z(\mathbf{r}+\mathbf{a}_1),\ldots,z(\mathbf{r}+\mathbf{a}_N)]f_{d_0}(\mathbf{r})f_{d_1}(\mathbf{r}+\mathbf{a}_1)\cdots f_{d_N}(\mathbf{r}+\mathbf{a}_N)d\mathbf{r} \tag{1}$$

where $\mathbf{a}_i$ are displacement vectors from the reference point r, f_d is the d^{th} element of f, and $\omega(\cdot)$ indicates a weighting function. $N\in\{0,1\}$, $a_{1x,y}\in\{\pm\Delta r,0\}$, and $\omega(\cdot)\equiv\min(\cdot)$ were considered as suggested in [18], where Δr represents the displacement interval in both horizontal and vertical directions. For $N\in\{0,1\}$, the formulation of GLAC is given by

$$\mathbf{F}_0 : R_{N=0}(d_0)=\sum_{\mathbf{r}\in I} z(\mathbf{r})f_{d_0}(\mathbf{r}) \tag{2}$$

$$\mathbf{F}_1 : R_{N=1}(d_0,d_1,\mathbf{a}_1)=\sum_{\mathbf{r}\in I} \min[z(\mathbf{r}),z(\mathbf{r}+\mathbf{a}_1)]f_{d_0}(\mathbf{r})f_{d_1}(\mathbf{r}+\mathbf{a}_1) \tag{3}$$

The spatial auto-correlation patterns of $(\mathbf{r},\mathbf{r}+\mathbf{a}_1)$ are shown in Fig. 1.

The dimensionality of the above GLAC features (F_0 and F_1) is $D+4D^2$. Although the dimensionality of the GLAC features is high, the computational cost is low due to the sparseness of f. It is worth noting that the computational cost is invariant to the number of bins, D, since the sparseness of f doesn't depend on D.

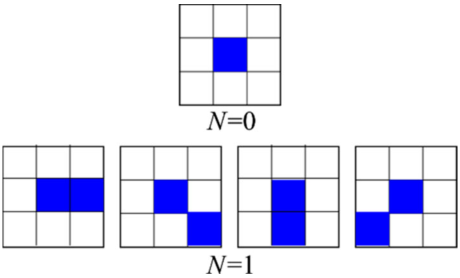


Fig. 1 Configuration patterns of $(\mathbf{r},\mathbf{r}+\mathbf{a}_1)$

Since GLAC is used to extract features from 2D images, we first utilize the method discussed in [7] to generate three DMMs due to its computational efficiency. Each 3D depth image in a depth video sequence is first projected onto three orthogonal Cartesian planes to generate three 2D projected maps corresponding to front, side, and top views, denoted by map_f, map_s, and map_t, respectively. Under each projection view, the absolute difference between two consecutive projected maps is accumulated through an entire depth video sequence forming a DMM. For a depth video sequence with N' frames, the DMMs are obtained as follows:

$$DMM_{\{f,s,t\}} = \sum_{i=2}^{N'} \left| map^{i}_{\{f,s,t\}} - map^{i-1}_{\{f,s,t\}} \right| \tag{4}$$

where i represents frame index. A bounding box is considered to extract the foreground in each DMM. The procedure of generating DMMs is illustrated in Fig. 2.

After DMMs generation, the method described in [4] is adopted to extract GLAC features from DMMs. Specifically, DMMs are first divided into several overlapped blocks and the GLAC descriptor is applied to each block to compute the GLAC features (i.e., $\mathbf{F}_0$ and $\mathbf{F}_1$). The GLAC features of all the blocks from three DMMs are concatenated to form a single composite feature vector $\mathbf{F}_G$.

2.2 Space-time auto-correlation of gradients

As stated earlier, STACOG was proposed in [19] to extract local relationships among the space-time gradients from RGB video sequences. Here, this approach is applied to depth video sequences in order to capture the geometric characteristics of a motion shape.

Let $I'(x, y, t)$ denote a depth image volume. The 3D space-time gradient vector (see Fig. 3a) can be obtained from the derivatives (I'_x, I'_y, I'_t) at each space-time point in a depth video sequence. The magnitudes of the gradient vectors are given by $m = \sqrt{I'^{2}_{x} + I'^{2}_{y} + I'^{2}_{t}}$. The spatial orientation in a depth image and the temporal elevation along the time axis can be represented by $\theta' = \arctan(I'_x, I'_y)$ and $\phi = \arcsin(I'_t/m)$, respectively. The space-time orientation of the gradient represented by θ' and ϕ is then placed into B orientation bins on a unit sphere by voting weights to the nearest bins to form a B -dimensional vector $\mathbf{v}$ (see Fig. 3b).

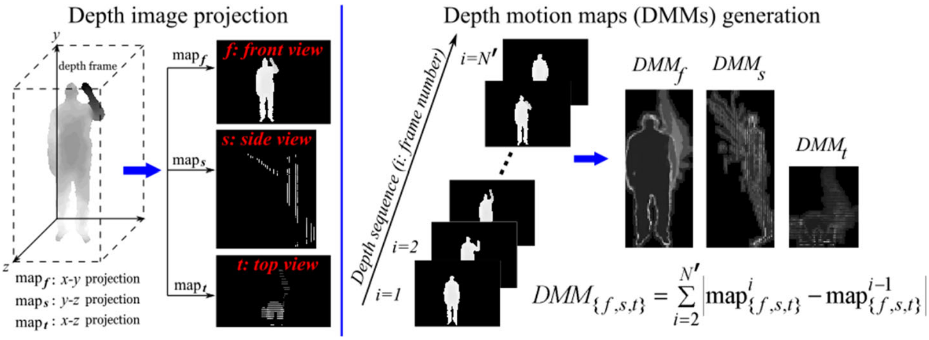


Fig. 2 Three DMMs (DMM_f, DMM_s *and* DMM_t) generated from a depth video sequence for the action *high throw*

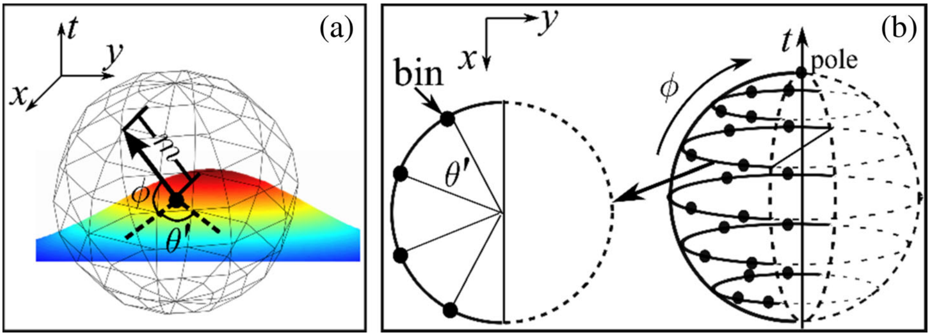


Fig. 3 **a** Space-time gradient **b** Orientation bins along latitude and longitude on a hemisphere (opposite directions along the longitude not used)

The N^{th} order auto-correlation function for the space-time gradients can be expressed as follows:

$$R'_N\left(\mathbf{a}'_1,\ldots,\mathbf{a}'_N\right)=\int g\left[m\left(\mathbf{r}'\right),\ldots,m\left(\mathbf{r}'+\mathbf{a}'_N\right)\right]\mathbf{v}\left(\mathbf{r}'\right)\otimes\cdots\otimes\mathbf{v}\left(\mathbf{r}'+\mathbf{a}'_N\right)d_{\mathbf{r}'} \tag{5}$$

where $[\mathbf{a}'_1,\ldots,\mathbf{a}'_N]$ are displacement vectors from the reference point $\mathbf{r}'=(x,y,t)$, g indicates a weighting function, and $\otimes$ denotes tensor product. In the experiments reported later, $N\in\{0,1\}$, $a'_{1x,y}\in\{\pm\Delta r',0\}$, $a'_{1t}\in\{\pm\Delta t',0\}$ and $g(\cdot)\equiv\min(\cdot)$ were considered as suggested in [19], where $\Delta r'$ and $\Delta t'$ represent the displacement interval along the spatial and temporal axis, respectively. For $N\in\{0,1\}$, the STACOG features can be written as follows:

$$\mathbf{F}'_0=\sum_{\mathbf{r}'}m\left(\mathbf{r}'\right)\mathbf{v}\left(\mathbf{r}'\right) \tag{6}$$

$$\mathbf{F}'_1\left(\mathbf{a}'_1\right)=\sum_{\mathbf{r}'}\min\left[m\left(\mathbf{r}'\right),m\left(\mathbf{r}'+\mathbf{a}'_1\right)\right]\mathbf{v}\left(\mathbf{r}'\right)\mathbf{v}\left(\mathbf{r}'+\mathbf{a}'_1\right)^T \tag{7}$$

Since there are 13 different configuration patterns of $(\mathbf{r}',\mathbf{r}'+\mathbf{a}'_1)$ as illustrated in Fig. 4, the dimensionality of the above STACOG features ($\mathbf{F}'_0$ and $\mathbf{F}'_1$) becomes $B+13B^2$. It is worth noting that $\Delta r'/\Delta t'$ is closely connected to the velocity of motion. The frame-based STACOG features in Eqs. (6) and (7) are extracted by summing up over the full space-time region within an entire depth sequence. We denote the STACOG feature vector for a depth sequence by $\mathbf{F}_S$. A detailed description of the STACOG features is provided in [19].

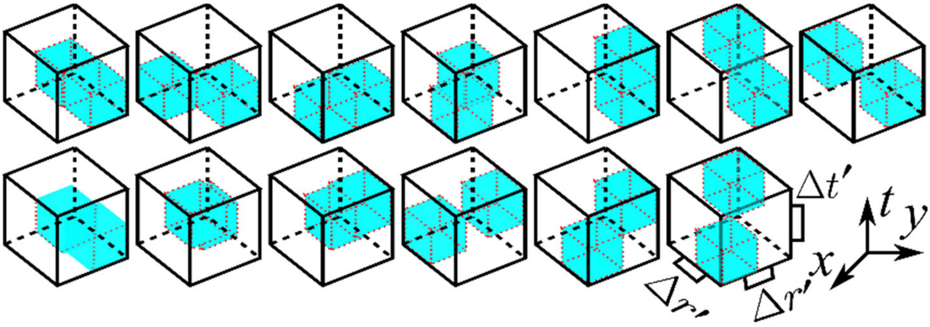


Fig. 4 Configuration patterns of $(\mathbf{r}',\mathbf{r}'+\mathbf{a}'_1)$

3 Decision-level fusion based on ELM

In this section, we present a weighted decision-level fusion scheme based on ELM to effectively combine the 2D (DMMs-based GLAC) and 3D (STACOG) auto-correlation of gradients features for action recognition.

3.1 Extreme learning machine

ELM [16] is an efficient learning algorithm for single hidden layer feed-forward neural networks (SLFNs) and has been applied in various applications (e.g., [8,9,22]). Let $\mathbf{y} = [y_1, \ldots, y_k, \ldots, y_C]^T \in \mathbb{R}^C$ be the class to which a sample belongs, where $y_k \in \{1, -1\}$ $(1 \leq k \leq C)$ and C is the number of classes. Given n training samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^M$ and $\mathbf{y}_i \in \mathbb{R}^C$, a single hidden layer neural network having L hidden nodes can be expressed as

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

where $h(\cdot)$ is a nonlinear activation function, $\boldsymbol{\beta}_j \in \mathbb{R}^C$ denotes the weight vector connecting the j^{th} hidden node to the output nodes, $\mathbf{w}_j \in \mathbb{R}^M$ 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. Eq. (8) can be written compactly as:

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

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T; \ldots; \boldsymbol{\beta}_L^T] \in \mathbb{R}^{L \times C}, \mathbf{Y} = [\mathbf{y}_1^T; \ldots; \mathbf{y}_n^T] \in \mathbb{R}^{n \times C}$,, and $\mathbf{H}$ is the hidden layer output matrix. A least-squares solution $\hat{\boldsymbol{\beta}}$ to Eq. (9) is

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

where $\mathbf{H}^\dagger$ is the *Moore-Penrose inverse* of $\mathbf{H}$. 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.

In ELM, a feature mapping $\mathbf{h}(\mathbf{x}_i)$ is usually known to users. If a feature mapping is unknown to users, a kernel matrix for ELM can be defined as follows:

$$\Omega_{ELM} = \mathbf{H}\mathbf{H}^T : \Omega_{ELM_{i,j}} = \mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}\left(\mathbf{x}_j\right) = K\left(\mathbf{x}_i, \mathbf{x}_j\right) \tag{12}$$

Thus, the output function of kernel-based ELM (KELM) can be written as

$$\mathbf{f}_L(\mathbf{x}_i) = \begin{bmatrix} K(\mathbf{x}_i, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}_i, \mathbf{x}_P) \end{bmatrix}^T \left(\frac{\mathbf{I}}{\rho} + \Omega_{ELM}\right)^{-1} \mathbf{Y} \tag{13}$$

The label of a test sample $\mathbf{x}_l$ is determined by the index of the output node with the largest value, i.e.,

$$y_l = \underset{k=1,\ldots,C}{\operatorname{argmax}} f_L(x_l)_k \tag{14}$$

where $f_L(x_l)_k$ denotes the k^{th} output of $\mathbf{f}_L(\mathbf{x}_l) = [f_L(x_l)_1, f_L(x_l)_2, \ldots, f_L(x_l)_C]^T$. In our experiments, we use KELM with a radial basis function (RBF) kernel.

3.2 Weighted fusion of 2D and 3D auto-correlation of gradients features

Two sets of features are generated from a depth sequence including the DMMs-based GLAC features ($\mathbf{F}_G$) and the STACOG features ($\mathbf{F}_S$). In this paper, we use decision-level fusion [4] to merge results from a classifier ensemble of two sets of features. Specifically, $\mathbf{F}_G$ and $\mathbf{F}_S$ are used individually as input to two ELM classifiers. The probability outputs of each individual classifier are merged to generate the final outcome. The posterior probabilities are estimated using the decision function of ELM (i.e., $\mathbf{f}_L$ in Eq. (11)) since it estimates the accuracy of the output label. $\mathbf{f}_L$ is normalized to $[0, 1]$ and Platt's empirical analysis [13] using a Sigmoid function is utilized to approximate the posterior probabilities,

$$p(y_k|\mathbf{x}) = \frac{1}{1 + \exp\left(Af_L(\mathbf{x})_k + B\right)} \tag{15}$$

where $f_L(\mathbf{x})_k$ is the k^{th} output of the decision function $\mathbf{f}_L(\mathbf{x})$. In our experiments, we set $A = -1$ and $B = 0$. Logarithmic opinion pool (LOGP) [22] is employed to estimate a global membership function:

$$\log P(y_k|\mathbf{x}) = \sum_{q=1}^{Q} \alpha_q p_q(y_k|\mathbf{x}) \tag{16}$$

where Q is the number of classifiers and $\{\alpha_q\}_{q=1}^{Q}$ indicate classifier weights. In our proposed method, Q is set to 2 since there are two ELM classifiers for two sets of features ($\mathbf{F}_G$ and $\mathbf{F}_S$). Therefore, Eq. (16) becomes

$$\log P(y_k|\mathbf{x}) = \alpha_1 p_1(y_k|\mathbf{F}_G) + \alpha_2 p_2(y_k|\mathbf{F}_S) \tag{17}$$

For the classifier weights, the uniformly distributed weights (i.e., $\alpha_q = 1/Q$) are usually utilized (e.g., [4,22]). However, it may not be reasonable to assign equal weights to different features because they have different importance in decision making. A larger weight means the corresponding feature is more important in decision making. Therefore, we propose to use different classifier weights to provide more freedom to different features, thereby improving the flexibility of the fusion approach. In the weighted fusion strategy, we impose the weights with non-negativity and sum-to-one constraints. The fused probability output can be further written as

$$\log P(y_k|\mathbf{x}) = \mu p_1(y_k|\mathbf{F}_G) + (1-\mu) p_2(y_k|\mathbf{F}_S) \tag{18}$$

where $\mu(\mu \geq 0)$ is the weight assigned to the classifier using $\mathbf{F}_G$ as the feature input. The final class label y^* is determined according to

$$y^* = \underset{k=1,\ldots,C}{\operatorname{argmax}} P(y_k|\mathbf{x}) \tag{19}$$

Figure 5 summarizes the overall framework of the proposed action recognition method.

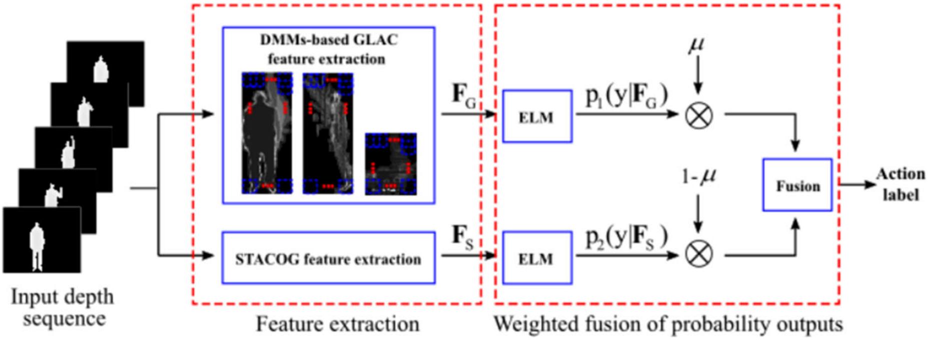


Fig. 5 The framework of the proposed action recognition method

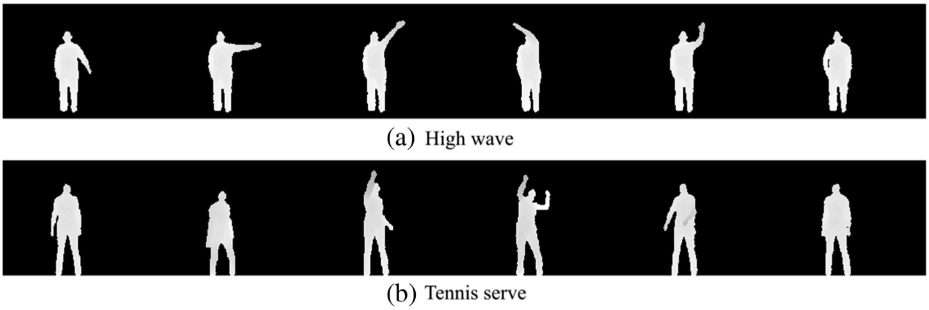
4 Experiments

In this section, the performance of the proposed action recognition algorithm on two benchmark depth action datasets (MSR Action 3D [33] and MSR Gesture 3D [21]) is investigated. The action recognition results are compared with other state-of-the-art methods.

4.1 Experimental data and setup

The MSR Action 3D dataset [33] includes 20 actions performed by 10 subjects. The 20 actions are: *high wave*, *horizontal wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side boxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, and *pickup throw*. Each subject performed each action 2 or 3 times. This dataset includes 557 action sequences with a resolution of 320 × 240 pixels. It is a challenging dataset due to similarity of actions, e.g., *draw x* and *draw tick*. Some example depth images from the dataset are shown in Fig. 6. The same experimental setup in [33] 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.

The second dataset used for evaluation is the MSR Gesture 3D dataset [21]. It is a hand gesture dataset of depth sequences captured by a depth camera. This dataset contains a subset of gestures defined by American Sign Language (ASL). There are 12 gestures in the dataset: *bathroom*, *blue*, *finish*, *green*, *hungry*, *milk*, *past*, *pig*, *store*, *where*, *j*, and *z*. The dataset contains 333 depth sequences, and is considered challenging because of self-occlusions. Some


(a) High wave

(b) Tennis serve

Fig. 6 Examples of depth sequences from the MSR Action 3D dataset

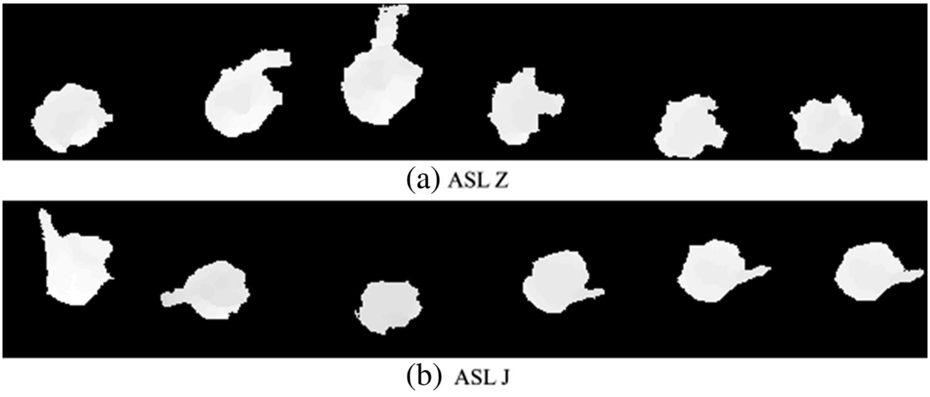
example depth images from this dataset are presented in Fig. 7. For this dataset, the leave-one-subject-out cross-validation test [32] is performed.

4.2 Parameters setting

For our proposed action recognition framework, appropriate values for several parameters need to be set first. To compute the DMMs for the depth sequences, the same parameters (sizes of DMMs and blocks) reported in [4] are used for the two datasets. The same parameter setting for the GLAC descriptor is adopt according to [2], that is $(D, \Delta r) = (10, 8)$ for the MSR Action 3D dataset and $(D, \Delta r) = (12, 1)$ for the MSR Gesture 3D dataset.

For the STACOG feature descriptor, 4 orientation bins in x-y plane (2D space) and 6 orientation bin-layers (one layer is located at the pole, see Fig. 3b) are chosen as noted in [19], making the total number of bins to be 21 (i.e., $B = 21$). In the experiments, $\Delta t'$ (the displacement interval along the temporal axis) is set to 1 as suggested in [19] to cope with fast motion. Different $\Delta r'$ (spatial interval) values are examined using the 3-fold cross validation strategy based on available training samples. Figure 8 illustrates the recognition accuracy versus $\Delta r'$ for the MSR Action 3D dataset. Hence, $\Delta r' = 3$ for the STACOG features extraction is used.

The classifier weight (i.e., μ in Eq. (18)), which balances the probability contributions of using DMMs-based STACOG features ($\mathbf{F}_S$) and using the STACOG features ($\mathbf{F}_S$), varies from 0 to 1 with a step size of 0.1. When $\mu = 0$, it is equal to use $\mathbf{F}_S$ only for action recognition, and $\mu = 1$, it is equal to use $\mathbf{F}_G$ only. To estimate the optimal weight parameter μ, the 3-fold cross validation strategy based on the training samples is employed. Figure 9 illustrates the recognition accuracy with various values of μ on the MSR Action 3D dataset. As can be seen from this figure, the best action recognition performance corresponds to $\mu = 0.6$. It indicates the probability output of the classifier using $\mathbf{F}_G$ is given more importance than that of the classifier using $\mathbf{F}_S$ for the final decision making. This is because a recognition accuracy of 90 % is achieved by using $\mathbf{F}_G$ only ($\mu = 0$), which is about 12 % higher than when using $\mathbf{F}_S$ only ($\mu = 0$). Therefore, it is reasonable to assign larger weight to $\mathbf{F}_G$ since it exhibits more discriminative power than $\mathbf{F}_S$.


(a) ASL Z

(b) ASL J

Fig. 7 Examples of depth sequences from the MSR Gesture 3D dataset

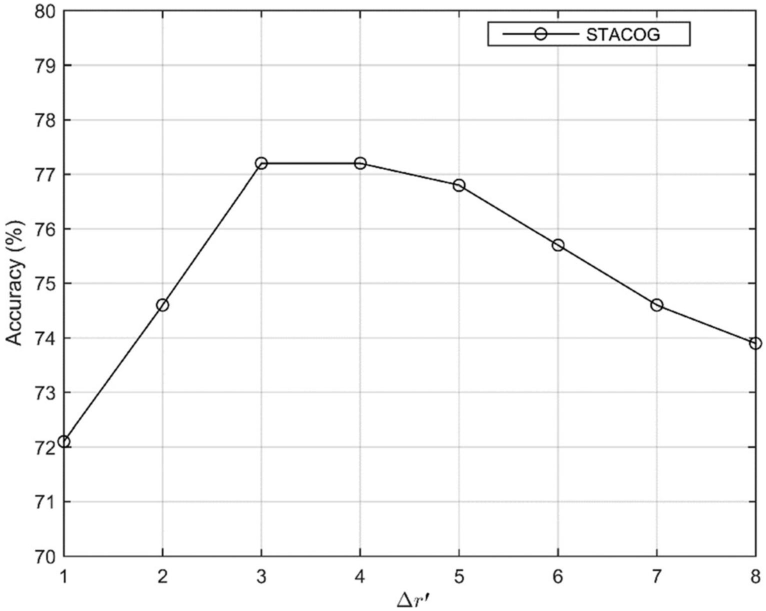


Fig. 8 Action recognition accuracy (%) using the STACOG features versus $\varDelta r'$ for the MSR Action 3D dataset with the 3-fold cross validation test on the training samples

4.3 Results

To prove the effectiveness of the proposed action recognition algorithm, we compare its action recognition performance on the MSR Action 3D and MSR Gesture 3D datasets with the state-of-the-art performance reported in the literatures, under the same experimental setup described in Section 4.1 for those two datasets. The outcomes of the comparison are listed in Tables 1 and 2 for the MSR Action 3D dataset and the MSR Gesture 3D dataset, respectively. The

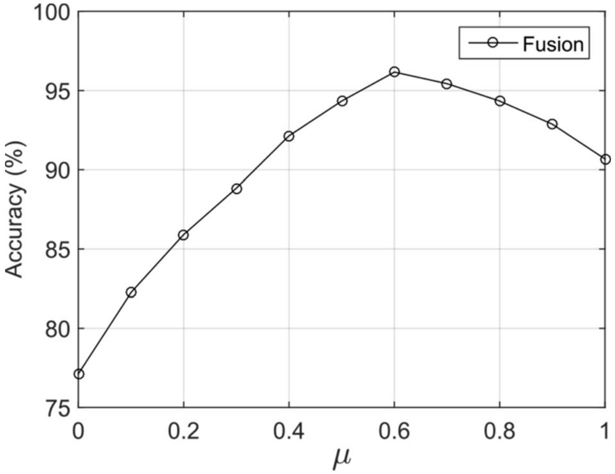


Fig. 9 Action recognition accuracy (%) versus μ for the MSR Action 3D dataset with the 3-fold cross validation test on the training samples

Table 1 Comparison of recognition accuracy on the MSRAction3D dataset

Method	Accuracy
Bag of 3D points [33]	74.70 %
EigenJoints [34]	82.30 %
STOP [31]	84.80 %
Random Occupancy Pattern [32]	86.50 %
Actionlet Ensemble [17]	88.20 %
DMM-HOG [36]	88.73 %
Histograms of Depth Gradients [28]	88.80 %
HON4D [26]	88.89 %
DSTIP [24]	89.30 %
Skeletons in a Lie group [29]	92.46 %
DMM-LBP [4]	93.00 %
SNV [35]	93.09 %
Hierarchical 3D Kernel Descriptors [20]	92.73 %
HOG3D + locality-constrained linear coding (LLC) [27]	90.90 %
DMM-GLAC	89.38 %
STACOG	75.82 %
Proposed	94.87 %

performances of using the DMMs-based GLAC features only (denoted by DMM-GLAC) and the STACOG features only (denoted by STACOG) are also reported. It is easy to see that our proposed method obtains the state-of-the-art accuracies of 94.87 % for the MSR Action 3D dataset and 98.5 % for the MSR Gesture 3D dataset. Especially for the MSR Gesture 3D dataset, our method outperforms the comparison methods considerably, leading to almost 3 % improvement over the second best result (95.66 % in [20]).

To further show the recognition performance, the confusion matrices of our method for the MSRAction3D dataset and the MSR Gesture 3D dataset are shown in Figs. 10 and 11, respectively. For the MSR Action 3D dataset, the most confusion occurs in recognizing similar

Table 2 Comparison of recognition accuracy on the MSRGesture3D dataset

Method	Accuracy
Random Occupancy Pattern [32]	88.50 %
DMM-HOG [36]	89.20 %
Histograms of Depth Gradients [28]	93.60 %
HON4D [26]	92.45 %
Action Graph on Silhouett [21]	87.70 %
Edge Enhanced DMM [10]	90.50 %
DMM-LBP [4]	94.60 %
SNV [35]	94.74 %
Hierarchical 3D Kernel Descriptors [20]	95.66 %
HOG3D + Locality-constrained linear coding (LLC) [27]	94.10 %
DMM-GLAC	95.30 %
STACOG	92.60 %
Proposed	98.50 %

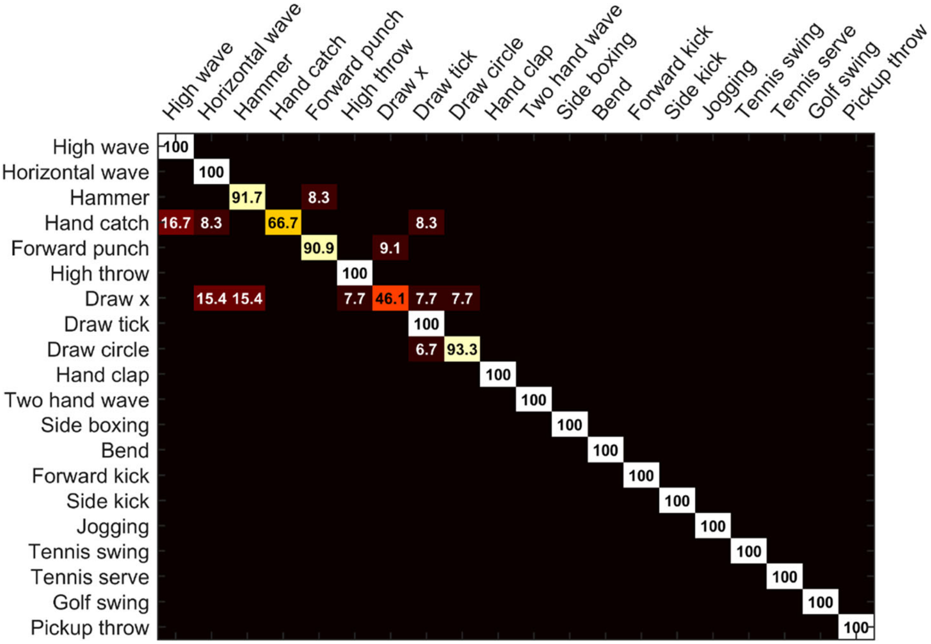


Fig. 10 Confusion matrix of our method on the MSR Action 3D dataset

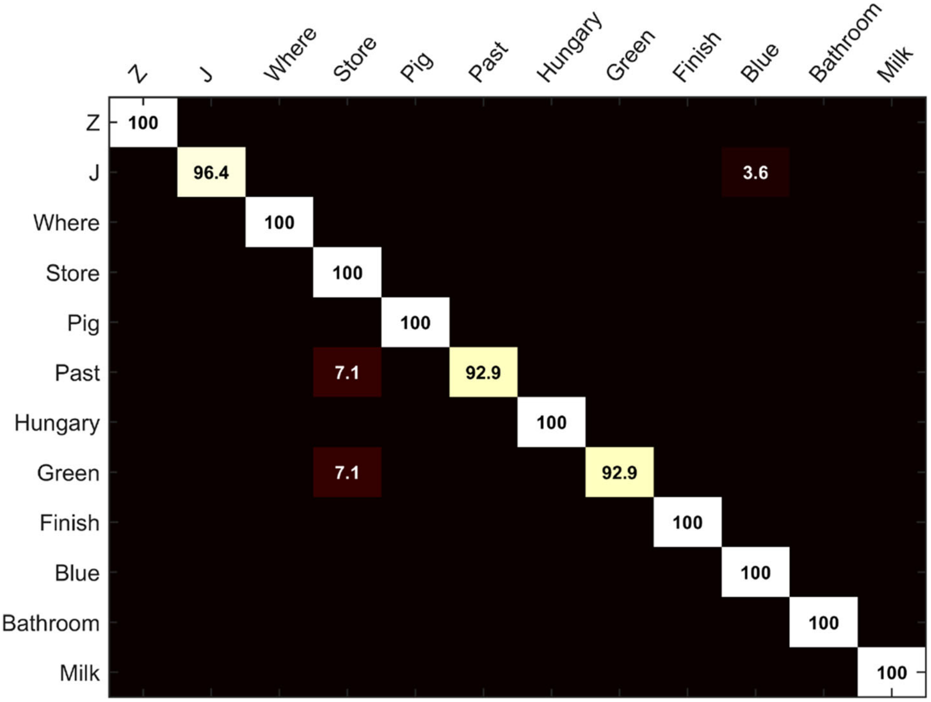


Fig. 11 Confusion matrix of our method on the MSR Gesture 3D dataset

actions, for example, *hand catch* and *high wave*, draw x and horizontal wave, and *hammer* and *forward punch*. The DMMs-based GLAC features may not be discriminative enough to distinguish these actions with similar motion due to the similarities of their DMMs. In addition, since the STACOG descriptor characterizes the space-time motion shape of an action sequence, it is also a challenge to accurately distinguish similar actions based on the STACOG features. For the MSR Gesture 3D dataset, only gestures *J*, *past* and *store* didn't reach 100 % accuracy. The misclassifications mostly occurred among *store*, *past* and *green*.

It is also important to observe that, by combining the DMMs-based GLAC features and the STACOG features, the overall recognition accuracy is improved considerably over the situations when using the DMMs-based GLAC features alone or the STACOG features alone. For example, the proposed method has over 5 % higher accuracy than DMM-GLAC and over 19 % higher accuracy than STACOG on the MSR Action 3D dataset. It clearly demonstrate the advantage of fusion the 2D and 3D auto-correlation of gradients features for improving the action recognition performance. To further examine the improvement, we compare the class-specific recognition accuracy (i.e., recognition accuracy per action class) associated with the proposed method, DMM-GLAC and STACOG for the MSR Action 3D dataset in Fig. 12. As evident from this figure, the proposed fusion method is able to improve the classification performance for most of the action classes, e.g., *hammer*, *hand catch*, *draw tick*, and *draw circle*.

5 Conclusion

In this paper, we proposed an effective feature representation method by combine two sets of powerful features based on auto-correlation of gradients. The DMMs-based GLAC features are used to capture the rich texture information from the DMMs of a depth sequence. The STACOG descriptor, which is a 3D extension of the GLAC descriptor, characterizes the space-

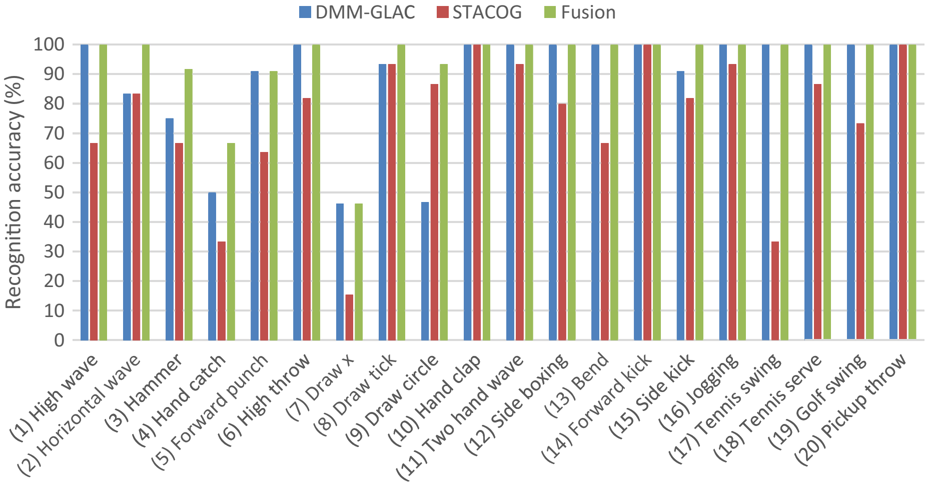


Fig. 12 Class-specific recognition performances of the proposed method (denoted by Fusion), DMM-GLAC and STACOG for the MSR Action 3D dataset

time motion shape of a depth sequence. It also helps bringing more temporal information of a depth sequence that is lost in the DMMs. A weighted fusion scheme based on ELM was proposed to provide more flexibility in combining the two sets of features. The proposed action recognition approach was extensively evaluated on two depth action datasets. The experimental results demonstrated that the proposed method consistently outperformed the state-of-the-art action recognition algorithms. However, there are several problems need to be investigated in the future. The classifier weights developed in this paper were determined based on the training samples and were fixed for all the testing samples. An adaptive classifier weights assignment strategy according to the feature characteristic of each testing sample will be an open question.

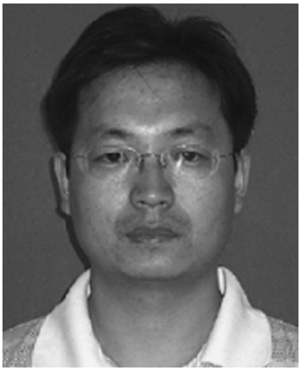
Acknowledgments This work was supported in the part by Natural Science Foundation of China, under Contracts 61272052 and 61473086, and by the Program for New Century Excellent Talents University of Ministry of Education of China. Thanks for the support of PAPD and CICAEET. Baochang Zhang is the correspondence.

References

1. Aggarwal JK, Lu X (2014) Human activity recognition from 3d data: a review. Pattern Recogn Lett 48:70–80
2. Chen C, Hou Z, Zhang B, Jiang J, Yang Y (2015) Gradient local auto-correlations and extreme learning machine for depth-based activity recognition. In: 11th international symposium on Visual Computing (ISVC'15), Las Vegas, December 14–16, pp 613–623
3. Chen C, Jafari R, Kehtarnavaz N (2015) Improving human action recognition using fusion of depth camera and inertial sensors. IEEE Trans Human-Mach Syst 45(1):51–61
4. Chen C, Jafari R, Kehtarnavaz N (2015) Action recognition from depth sequences using depth motion maps-based local binary patterns. In: 2015 IEEE winter conference on Applications of Computer Vision (WACV). IEEE
5. Chen C, Kehtarnavaz N, Jafari R (2014) A medication adherence monitoring system for pill bottles based on a wearable inertial sensor. In: EMBC, p 4983–4986
6. Chen C, Liu K, Jafari R, Kehtarnavaz N (2014) Home-based senior fitness test measurement system using collaborative inertial and depth sensors. In: EMBC, p 4135–4138
7. Chen C, Liu K, Kehtarnavaz N (2013) Real-time human action recognition based on depth motion maps. Journal of real-time image processing, p 1–9
8. Chen C, Zhang B, Su H, Li W, Wang L, (2015) Land-use scene classification using multi-scale completed local binary patterns. Signal, image and video processing
9. Chen C, Zhou L, Guo J, Li W, Su H, Guo F (2015) Gabor-filtering-based completed local binary patterns for land-use scene classification. In: 2015 IEEE international conference on multimedia big data, p 324–329
10. Chenyang Z, Tian Y (2013) Edge enhanced depth motion map for dynamic hand gesture recognition. In: 2013 IEEE conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE
11. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: CVPR, p 886–893
12. Georgios E, Singh G, Horaud R (2014) Skeletal quads: Human action recognition using joint quadruples. In: 2014 22nd international conference on Pattern Recognition (ICPR). IEEE
13. Gu B, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. IEEE Trans Neural Netw Learn Syst 26(7):1403–1416
14. Han J, Pauwels EJ, de Zeeuw PM, de With PHN (2012) Employing a RGB-D sensor for real-time tracking of humans across multiple re-entries in a smart environment. Transactions on Consumer Electronics 58(2):255–263
15. Han J, Shao L, Xu D, Shotton J (2013) Enhanced computer vision with microsoft kinect sensor: a review. IEEE Transactions on Cybernetics 43(5):1318–1334

16. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
17. Jiang W, et al (2012) Mining actionlet ensemble for action recognition with depth cameras. In: 2012 IEEE conference on Computer Vision and Pattern Recognition (CVPR). IEEE
18. Kobayashi T, Otsu N (2008) Image feature extraction using gradient local auto-correlations. In: ECCV, p 346–358
19. Kobayashi T, Otsu N (2012) Motion recognition using local auto-correlation of space-time gradients. Pattern Recogn Lett 33(9):1188–1195
20. Kong Y, Sattar B, Fu Y (2015) Hierarchical 3D Kernel Descriptors for Action Recognition Using Depth Sequences. IEEE international conference on Automatic Face and Gesture Recognition (FG)
21. Kurakin A, Zhang Z, Liu Z (2012) A real time system for dynamic hand gesture recognition with a depth sensor. In: EUSIPCO, p 1975–1979
22. Li W, Chen C, Su H, Du Q (2015) Local binary patterns and extreme learning machine for hyperspectral imagery classification. IEEE Trans Geosci Remote Sens 53(7):3681–3693
23. Liu L, Shao L (2013, August) Learning discriminative representations from RGB-D video data. In: Proceedings of the twenty-third international joint conference on artificial intelligence, pp. 1493–1500. AAAI Press
24. Lu X, Aggarwal JK (2013) Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: 2013 IEEE conference on Computer Vision and Pattern Recognition (CVPR). IEEE
25. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
26. Omar O, Liu Z (2013) Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: 2013 IEEE conference on Computer Vision and Pattern Recognition (CVPR). IEEE
27. Rahmani H, Huynh DQ, Mahmood A, Mian A (2015) Discriminative human action classification using locality-constrained linear coding, Pattern recognition letters
28. Rahmani H, Mahmood A, Huynh DQ, Mian A (2014) Real time action recognition using histograms of depth gradients and random decision forests. In: WACV, p 626–633
29. Raviteja V, Arrate F, Chellappa R (2014) Human action recognition by representing 3d skeletons as points in a lie group. In: 2014 IEEE conference on Computer Vision and Pattern Recognition (CVPR). IEEE
30. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: CVPR, p 1297–1304
31. Vieira AW, Nascimento ER, Oliveira GL, Liu Z, Campos MF (2012) Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: CIARP, p 252–259
32. Wang J, Liu Z, Chorowski J, Chen Z, Wu, Y (2012) Robust 3d action recognition with random occupancy patterns. In: ECCV, p 872–885
33. Wanqing L, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3d points. In: 2010 IEEE computer society conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE
34. Xiaodong Y, Tian YL (2012) Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: 2012 IEEE computer society conference on Computer Vision and Pattern Recognition Workshops. IEEE
35. Xiaodong Y, Tian YL (2014) Super normal vector for activity recognition using depth sequences. 2014 IEEE conference on Computer Vision and Pattern Recognition (CVPR). IEEE
36. Xiaodong Y, Zhang C, Tian YL (2012) Recognizing actions using depth motion maps-based histograms of oriented gradients. Proceedings of the 20th ACM international conference on multimedia. ACM
37. Yu M, Liu L, Shao L (2015) Structure-preserving binary representations for RGB-D action recognition. In: IEEE Transactions on pattern analysis and machine intelligence. doi:10.1109/TPAMI.2015.2491925
38. Zhang B, Gao Y, Zhao S, Liu J (2010) Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. IEEE Trans Image Process 19(2):533–544
39. Zhang B, Shan S, Chen X, Gao W (2007) Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition. IEEE Trans Image Process 16(1):57–68

Chen Chen received the BE 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. He is a PhD candidate in the Department of Electrical Engineering at the University of Texas at Dallas, Richardson, TX. His research interests include compressed sensing, signal and image processing, pattern recognition and computer vision.



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 also holds 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.

Zhenjie Hou received the PhD degree in mechanical engineering from Inner Mongolia Agricultural University, in 2005. From 1998 to 2010, he was a professor in the computer science department of Inner Mongolia Agricultural University. In Aug. of 2010, he joined Changzhou University. His research interests include signal and image processing, pattern recognition and computer vision.

Junjun Jiang received the B.S. degree from School of Mathematical Sciences, Huaqiao University, Quanzhou, China, in 2009, and the Ph.D. degree from School of Computer, Wuhan University, Wuhan, China, in 2014. He is currently an Associate Professor with the School of Computer Science, China University of Geosciences. His research interests include image processing, pattern recognition, hyperspectral remote sensing and high-resolution remote sensing.

Mengyuan Liu received the B.E. degree in intelligence science and technology in 2012, and is working toward the Doctor degree in the School of EE&CS, Peking University (PKU), China. His research interests include Action Recognition and Localization. He has published articles in IEEE International Conference Robotics and Biomimetics (ROBIO), IEEE International Conference on Image Processing (ICIP), IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Neurocomputing.

Yun Yang received the BE degree in Automation science and electrical engineering department of Beihang University, Beijing, China. Now he is a master student working at Machine Perception Lab in the same department. His research focuses on human action recognition, deep learning and pedestrian re-identification.