Depth-based human activity recognition via multi-level fused features and fast broad learning system

Huang Yao, Mengting Yang, Tiantian Chen, Yantao Wei and Yu Zhang

Abstract

Human activity recognition using depth videos remains a challenging problem while in some applications the available training samples is limited. In this article, we propose a new method for human activity recognition by crafting an integrated descriptor called multi-level fused features for depth sequences and devising a fast broad learning system based on matrix decomposition for classification. First, the surface normals are computed from original depth maps; the histogram of the surface normal orientations is obtained as a low-level feature by accumulating the contributions from normals, then a high-level feature is acquired by sparse coding and pooling on the aggregation of polynormals. After that, the principal component analysis is applied to the conjunction of the two-level features in order to obtain a low-dimensional and discriminative fused feature. At last, fast broad learning system based on matrix decomposition is proposed to accelerate the training process and enhance the classification results. The recognition results on three benchmark data sets show that our method outperforms the state-of-the-art methods in term of accuracy, especially when the number of training samples is small.

Keywords

Human activity recognition, broad learning system, multi-level fused features, principal component analysis

Date received: 19 November 2019; accepted: 22 January 2020

Handling Editor: Wei Wang

Introduction

Human activity recognition (HAR) is a research hot-spot in the field of computer vision and pattern recognition with wide applications such as intelligent video surveillance, human computer interaction, ambient assisted living, virtual reality, and so on. Early researches on HAR have mainly focused on recognizing activities from RGB videos and many successful approaches have been proposed. However, the effectiveness of RGB cameras deteriorates because of illumination changes, surrounding clutters and disorder. The inventions of the cost-effective depth sensors such as Microsoft Kinect and Asus Xtion Pro have triggered the imagination of many researchers about activity recognition. Abundant structure information captured by depth sensors is insensitive to illumination variations, robust to complex background, and valuable for obtaining geometric information. Many researches have been carried out with depth maps for HAR. However, when the depth training samples are limited because of the cost-intensive collection of depth data of human activity, most of these methods cannot achieve
required accuracy due to weak descriptors and rough classifiers.

Current researches have investigated a number of human body representations including skeleton joints, cloud points, local interest points, projected depth maps, and surface normals. In Luo et al., a skeleton-based discriminative dictionary learning approach is proposed through utilizing group sparsity and geometry constraints. Vemulapalli et al. took skeletons as points and actions as curves in a Lie group by using the three-dimensional (3D) relative geometry between body parts. However, skeletons are usually noisy due to the difficulty in localizing body parts, self-occlusions, and sensor range errors. In contrast to skeleton joints, cloud points are more robust to occlusions and noise. In Wang et al., local occupancy patterns (LOP) were designed to subdivide the local 3D subvolumes related with skeleton joints into a group of spatial grids, then the number of cloud points falling into each grid was calculated. Rahmani et al. designed the histogram of oriented principal components (HOPC) to capture the local geometric information around each point in 3D cloud point videos, which is robust to viewpoint, scale, and temporal variations. To effectively suppress noise in the depth sequences, local spatiotemporal interest points (STIPs) were extracted from depth videos by a delicate filter to find out task-related interest points in Xia and Aggarwal. In order to transform the depth data from 3D to two-dimensional (2D), Yang et al. proposed depth motion maps (DMM) generated by projecting the depth maps onto three orthogonal planes and thresholding the difference of consecutive depth frames for each projected view, then applied the histogram of oriented gradients (HOG) to each 2D projected view to extract the features. However, the DMM features employed in Chen et al. and Yang et al. cannot capture the temporal information and thus suffered from the temporal disordering.

Surface normal has already been proved that it can extract valuable shape and structure information from depth maps. In Oreifej and Liu, histogram of the surface normal orientation in four-dimensional (4D) space (HON4D) in terms of time, depth, and 2D viewing planar was designed to capture the complex joint shape-motion cues at pixel level. Although as a low-level feature, it can capture motion and geometry cues effectively while being robust to occlusion. Yang and Tian proposed a new and high-level representation called super normal vector (SNV) by aggregating the low-level polynormals and concatenating the feature vector extracted from each adaptive spatiotemporal grid to encode spatiotemporal information. SNV is robust to noise; it can not only capture spatial and temporal order but also provide more distinguished local motion and appearance information for complex activities. In this article, data fusion is employed to gain robust and discriminative features to take the advantages of the both above features.

In recent years, deep learning methods have been widely used to automatically learn features from raw data and make successful computer vision applications, especially in HAR. Nonetheless, deep learning methods usually need large-scale training set, which is difficult because of economic and technical limit. Some recent works exploited transfer learning to deal with the lack of training samples. Nevertheless, the chosen parameters and models of deep learning methods still remain a challenging problem.

Broad learning system (BLS) is proposed as an improvement of the random vector functional link neural network (RVFLNN). Compared with the deep schemes of neural network models, RVFLNN dramatically reduces the training time and provides comparable generalization ability through combination of random functions. In BLS, the mapping features generated from the input data form the feature nodes of the network, then they are enhanced as enhancement nodes (EN) by randomly generated weights. Finally, all mapped features and EN are directly connected to the output, the corresponding output coefficients can be derived from pseudoinverse. BLS has been successfully applied to some specific image classification tasks, which outperformed common classifiers with limited labeled samples, such as k-nearest neighbor (KNN), support vector machine (SVM), and extreme learning machine (ELM). However, most of the real problems relevant to regression and classification are complex and need very broad-scale feature nodes, leading to extremely long training time. Therefore, we proposed a fast BLS based on matrix decomposition (FBLS-MD) to resolve this problem.

To handle the HAR tasks with limited training samples, in this article, we develop a robust descriptor of depth sequences called multi-level fused features (MLFF). In order to fully explore the MLFF’s validity, a FBLS-MD is further proposed. MLFF are generated by concatenating two different features extracted from low and high levels respectively. Since the low-level feature can capture the statistical patterns of shape changes of human activities while the high-level feature gives out a more comprehensive representation of spatial and temporal variations, MLFF are strongly robust to noise and occlusion. Principal component analysis (PCA) is further adopted to obtain lower dimensional features. Finally, FBLS-MD is carried out to classify activities and relieve the problem of heavy computation. The main contributions of our work are listed as follows:
We propose a new descriptor which is more discriminative and effective due to the complementarity between low-level and high-level features.

It is the first trial that introduces BLS into HAR classification. The proposed FBLS-MD can relieve the time-consuming training process caused by the large number of nodes.

The rest of this article introduces the detailed framework of our proposed method in the “Proposed method” section. In the “Experiments” section, we conduct experiments on three well known data sets and analyze the results. Finally, the “Conclusion” section summarizes the article.

Proposed method

Our proposed method has two major steps. First, MLFF of depth sequences are acquired by concatenating HON4D and SNV; then we employ PCA for dimensionality reduction. Second, FBLS-MD algorithm is introduced for efficient training and classification. The overview of our method is shown in Figure 1.

MLFF

All features in our work are calculated in 4D space (i.e. 2D images \((x, y)\), depth \((z)\), time \((t)\)). The steps to obtain the MLFF are illustrated in Figure 2.

Feature extraction. First, we extract HON4D in the same way as in the study by Oreifej and Liu.\(^\text{10}\) Depth sequences of human activities can be viewed as a hypersurface in 4D space \(R_3 \rightarrow R_1: z = f(x, y, t)\), where the set of points \((x, y, t, z)\) constitute a surface satisfying \(S(x, y, t, z) = f(x, y, t) - z = 0\); therefore, the normal of surface \(S\) is

\[
\mathbf{n} = \nabla S = \left( \frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, \frac{\partial S}{\partial t}, -1 \right)^T \tag{1}
\]
\( \hat{n} \) is computed by normalizing \( n \). Next, the 4D space is quantized by the 120 vertices of the 600-cell polytope, which divided it regularly. Then, the HON4D descriptor is calculated by accumulating the projection of unit normals \( \hat{n} \) onto the 120 vertices. As to further use the spatial-temporal information of a depth sequence, it is divided into \( 4 \times 3 \times 3 \) spatial-temporal subvolumes; the final descriptor is a concatenation of the HON4Ds acquired from all subvolumes.

Second, we extract the aggregated spatial-temporal features based on an improved spatial-temporal pyramid as in the study by Yang and Tian.\(^{37}\) The method generated the polynomial by clustering normals from a local spatiotemporal neighborhood to form the high-level features. \( N \) normals in the local neighborhood \( \Sigma \) of each cloud point are concatenated by a polynomial \( p \), which is denoted as

\[
p = (n_{1}^{T}, n_{2}^{T}, \ldots, n_{N}^{T}), n_{1}, \ldots, n_{N} \in \Sigma
\]

(2)

The neighborhood \( \Sigma \) is a spatiotemporal depth subvolume determined by two parameters \( r \) and \( t \), where \( r \) denotes the number of neighboring points in space and \( t \) indicates the number of neighboring depth maps in time series.

Then sparse coding\(^{38}\) is utilized to find a set of dictionary vectors encoding polynomials. And then the average pooling is applied spatially to aggregate the coefficient-weighted differences

\[
u_{k}(t) = \frac{1}{|N_{i}|} \sum_{i \in N_{i}} \alpha_{k,i}(p_{i} - d_{k})
\]

(3)

where \( u_{k}(t) \) represents the pooled difference vector of the \( k \)th visual word \( d_{k} \) of the \( t \)th frame, \( \alpha_{k,i} \) is the coefficient of sparse decomposition. Then the max pooling is used in temporal subsequence to aggregate the vectors from \( T \) frames and obtain \( u_{k} \), which represents the \( k \)th visual word in the whole volume. The final vector \( U \) is the concatenation of the \( u_{k} \) vectors from the \( K \) visual words, which is \( K \times M \) dimensions.

In order to exercise energy and characterize movement changes accurately, the \( t \)th frame \( M_{t} \) is projected onto three orthogonal planes to acquire the projected maps \( M_{t}^{v}, v \in 1, 2, 3 \). Then, the difference between two consecutive maps projected on the three planes is binarized with a specified threshold. We calculate the motion energy by accumulating the sum of the non-zero elements of the 2D graph as

\[
E(i) = \sum_{v=1}^{3} \sum_{j=1}^{i-1} \text{sum}(|M_{v}^{i+1} - M_{v}^{i}| > c_{v})
\]

(4)

where \( E(i) \) is the motion energy of the \( i \)th frame; \( c_{v} \) is the threshold of the \( v \)th projected map; \( \text{sum}(\cdot) \) returns the total number of non-zero elements in a binary map.

The motion energy of the \( i \)th frame is the energy superposition of the first frame to the \( i \)th frame.

In order to obtain information in the spatial dimensions, each frame is divided into \( h \times w \) blocks, and the entire activity is divided into \( T \) levels to obtain the information of the time dimension. Eventually each activity sequence is divided into \( h \times w \times (2^{T} - 1) \) blocks. This article uses a three-level pyramid in the time dimension: \( \{t_{0}, t_{4}, t_{12}, t_{16}\}, \{t_{0}, t_{4}, t_{12}, t_{16}, t_{20}, t_{24}, t_{28}, t_{32}\} \), which is shown in Figure 3.

Finally, the feature \( U \) extracted from each grid in the improved spatiotemporal pyramid are concatenated to form the high-level features as SNV.\(^{37}\)

**Feature fusion.** Feature fusion has been demonstrated to be an effective method to boost the performance in HAR system.\(^{39-42}\) And it is usually conducted through feature normalization and feature selection or transformation due to the highly correlated feature set and the curse of dimensionality.\(^{43}\)

In our method, HON4D\(^{10}\) of one cell has been extracted and the final features are denoted as

\[
H = \{H_{1}, H_{2}, \ldots, H_{N}\}
\]

(5)

where \( N \) represents the number of spatiotemporal cells. The final high-level descriptors \( V \) can be written as

\[
V = \{U(1), U(2), \ldots, U(c)\}
\]

(6)

where \( c \) indicates the number of space-time grids from the spatiotemporal pyramid.

We mark the normalized \( H \) and \( V \) as \( \tilde{H} \) and \( \tilde{V} \), then we concatenate the above two features and obtain \( F = [\tilde{V}, \tilde{H}] \).
Feature fusion usually produce representations in a higher dimensional space. Although pooling can eliminate data redundancy, its dimensionality reduction usually is a by-product rather than a direct goal. PCA44 is useful for dimensionality reduction, increasing interpretability and as the same time minimizing information loss; it can maximize variance by creating new uncorrelated variables. PCA has become an adaptive data analysis technique. Therefore, we employ PCA to reduce the dimension of the features thus improving the efficiency of the algorithm in this article. Finally, we can get the MLFF as the representation of a depth sequence.

The MLFF descriptors have the following obvious advantages. (1) Our descriptors are more robust and discriminative than previous representations. (2) With a lower dimension, MLFF can greatly improve the running speed of the algorithm while also increase the recognition rate.

Classification

In order to perform classification with the designed features, we feed the MLFF to FBLS-MD classifier, which can accelerate the training speed by making use of block matrix inversion lemma to decompose the large matrix inversion process. Next, the details of the algorithm will be introduced.

Given the input data set \(X\), which contains \(N\) samples, each with \(M\) dimensions, the output matrix \(Y\) belongs to \(R^N \times C\). The \(i\)th mapped features \(Z_i\) can be obtained according to

\[
Z_i = \phi(XW_{e_i} + \beta_{e_i}), \quad i = 1, 2, \ldots, n
\]

where \(W_{e_i}\) is the random weights with the proper dimensions, \(W_{e_i}\) and \(\beta_{e_i}\) are randomly generated. Note that different function \(\phi\) can be chosen for different groups of the mapped nodes (MN). In addition, all the MN are expressed as the set of \(Z = [Z_1, \ldots, Z_n]\), which is represented as a connection of the first \(i\) groups of mapping features. Furthermore, the \(m\)th group of EN can be written as

\[
H_m = \xi(Z^nW_{h_m} + \beta_{h_m})
\]

Finally, the broad learning model can be defined as follows

\[
Y = [Z_1, \ldots, Z_n; \xi(Z^nW_{h_m} + \beta_{h_m}), \ldots, \xi(Z^nW_{h_m} + \beta_{h_m})]W^m
\]

\[
= [Z_1, \ldots, Z_n; H_1, \ldots, H_m]W^n
\]

\[
= [Z^n; H^m]W^m
\]

where \(W^m = [Z^n; H^m]^{-1}Y\) is the connecting weights for the broad structure, and it was computed through the ridge regression approximation of \([Z^n; H^m]^{-1}\) in original BLS which was written as

\[
G^+ = \lim_{\lambda \to 0} (\lambda I + GG^T)^{-1}G^T
\]

where \(G = [Z^n; H^m]\), and \(\lambda\) is the regular \(l_2\)-norm regularization.

In FBLS-MD algorithm, the connecting weights \(W^m\) are decomposed into two parts

\[
W^m = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}
\]

where \(W_1 \in R^{L_1 \times m}\), \(W_2 \in R^{L_2 \times m}\), and \(L_1 + L_2 = L\), \(L\) represents the total numbers of the mapped features and EN.

Hence, the coefficient matrix \(G\) is decomposed into two small matrices as follows

\[
G = \begin{bmatrix} G_1 & G_2 \end{bmatrix}
\]

where \(G_1 \in R^{N \times L_1}\) and \(G_2 \in R^{N \times L_2}\). Based on BLS algorithm, the weights \(W\) can be obtained in the following way

\[
W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = (G^T G)^{-1}G^TY
\]

\[
= \left(\begin{bmatrix} G_1 & G_2 \end{bmatrix}^T [G_1, G_2]\right)^{-1} [G_1, G_2]^T Y
\]

\[
= \begin{bmatrix} G_1^T & G_2^T \end{bmatrix} [G_1, G_2]^{-1} [G_1, G_2]^T Y
\]

\[
= \begin{bmatrix} G_1^T G_1 & G_1^T G_2 \\ G_2^T G_1 & G_2^T G_2 \end{bmatrix}^{-1} G_1^T Y
\]

where

\[
U = \begin{bmatrix} G_1^T G_1 & G_1^T G_2 \\ G_2^T G_1 & G_2^T G_2 \end{bmatrix}^{-1}
\]

Through block matrix inversion Lemma, we can computer formula (14), then the connecting weights \(W\) can be written as follows

\[
\begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = \begin{bmatrix} G_1^T [I - G_2 (G_2^T)^{-1} G_2] G_1^{-1} \\ (G_1^T G_2)^{-1} G_1^T Y \\ (G_2^T G_2)^{-1} G_2^T G_1 (G_2^T G_2)^{-1} G_2^T Y \\ V^{-1} G_2^T (Y - G_2 K) \end{bmatrix} \begin{bmatrix} G_1^{-1} \\ G_1^{-1} G_2 \\ G_2^{-1} G_2 \\ K - \lambda^{-1} G_2 G_1 \end{bmatrix}
\]
where

\[ E = G_2^T G_1 \in R^{L_2 \times L_1} \quad (16) \]
\[ K = E^{-1} G_2^T Y \in R^{L_1 \times C} \quad (17) \]
\[ Q = I - G_2 E^{-1} G_2^T \in R^{N \times N} \quad (18) \]
\[ V = G_1^T Q G_1 \in R^{L_1 \times L_1} \quad (19) \]

where \( I \) represents an unit matrix. Therefore, The FBLS-MD is structured as Figure 4.

In summary, the training steps of FBLS-MD algorithm are shown in Table 1.

### Experiments

#### Experimental setup

The proposed method is extensively evaluated on three benchmark data sets, including MSR Action 3D data set, MSR Hand Gesture 3D data set, and 3D Action Pairs data set. For each activity, we extract its MLFF descriptors.

In the experiments, each video sequence is divided into space-time grids, which are \( 4 \times 3 \times 7 \) in width, height, and frame numbers respectively. The number of the visual words in the process of obtaining high-level descriptors is set to 100, and the size of local neighborhood is \( 3 \times 3 \times 3 \).

We evaluate the performance of our proposed method comparing with the state-of-the-art methods using the same experimental settings in Yang and Tian.\(^3\) For three data sets, \( s (s = 1, 2, 3, 4, 5) \) randomly chosen actors’ activities are used for training, while the remaining samples are used for testing. The selections of \( s \) are conducted randomly five times in each case to get the average results as the final recognition rate. In addition, the performance of FBLS-MD is compared with original BLS in term of the training time when the MN and EN increase gradually, which is verified on whole data set. It is worth noting that the experimental results about BLS and FBLS-MD algorithms are acquired by taking the average of 10 results. Our experiments are all performed using MATLAB on a computer with a 3.60 GHz Intel Core i7-4790 CPU and 16 GB RAM.

#### Experimental results and analysis

**MSR Action 3D data set.** MSR Action 3D\(^4\) is one of most classical data sets for HAR as recorded in related research literatures. It includes 20 different actions. Each action is performed by 10 actors for two or three times. Inevitably, there are some missing and wrong depth sequences. It is a challenging data set for HAR due to the similar actions. The specific actions in this data set are shown in Figure 5. With the same experiment setup as in Wang et al.\(^4\) (first five actors for training, and the rest for testing), we compare our results with the state-of-the-art methods on this data set and present the results in Table 2. This setting is much more challenging than which has

---

**Figure 4.** The structure of FBLS-MD.

**Figure 5.** The specific actions in MSR Action 3D data set.

**Table 1.** The training steps of the proposed FBLS-MD algorithm.

| Algorithm: FBLS-MD |
|---------------------|
| **Input:** training sample \( X \). | **Output:** the weights \( W \) of model. |
| 1: \( i = 0 \); \( i \leq n \) do | 3: Calculate \( Z_i = f_i(XW_i + b_i) \) |
| 2: Randomly generate \( W_i \), \( b_i \) | 4: end |
| 5: Obtain the feature mapping group \( Z_i = [Z_1, \ldots, Z_n] \). | 6: \( j = 1 \); \( j \leq m \) do |
| 7: Randomly produce \( W_h \), \( b_h \) | 8: Calculate \( H_j = g_j(ZW_h + b_h) \) |
| 9: end | 10: Get the enhancement nodes group \( H_m = [H_1, \ldots, H_m] \). |
| 11: Calculate \( W_m \) according to equations (11)–(19). | 12: Output the connecting weights. |
| 13: Obtain the model of FBLS-MD: \( Y = [Z][H][W] \). |

FBLS-MD: fast broad learning system based on matrix decomposition.
been used in Li et al.,48 because evaluation on whole action set increases the chance of confusion which often occurs in recognizing similar actions. As the result shows, our method is superior to other classical methods. The confusion matrix is demonstrated in Figure 6. It can be observed that our method has significant improvement on recognizing “hand catch” and “forward punch” actions by comparing with the results of Yang and Tian.37

Table 2. Performance comparison of the proposed method with the state-of-the-art methods on MSR Action 3D data set.

| Method                              | Accuracy (%) |
|-------------------------------------|--------------|
| HPM + TM47                          | 65.70        |
| Bag of 3D Points48                  | 74.70        |
| HOJ3D49                             | 79.00        |
| STOP50                              | 84.80        |
| ROP51                               | 86.50        |
| DMM19                               | 88.73        |
| HON4D10                             | 88.89        |
| DSTIP14                             | 89.30        |
| DMMs-based GLAC52                   | 90.48        |
| JSG + JSGK53                        | 92.20        |
| DMM-LBP-DF54                        | 93.00        |
| SNV37                               | 93.09        |
| MLFF + BLS                          | 93.86        |
| MLFF + FBLS-MD                      | 94.18        |

HFM: human pose representation model; TM: temporal modeling; HOJ3D: histograms of 3D joint locations; STOP: space-time occupancy patterns; ROP: random occupancy patterns; DMM: depth motion maps; HON4D: histogram of the surface normal orientation in four-dimensional space; DSTIP: spatial-temporal interest points from depth video; GLAC: gradient local auto-correlations; JSG: joint spatial graph; JSGK: joint spatial graph top-K; LBP-DF: local binary patterns-decision level fusion approach; SNV: super normal vector; MLFF: multi-level fused features; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition.

The highest two classification accuracies are marked in bold respectively.

Then, we evaluate the performance of our proposed descriptors. Table 3 compares MLFF descriptors with the single-feature methods on this data set. The results show our descriptors have a more powerful representation than HON4D or SNV. When the training samples are fewer, the improvements of the recognition rate are more observable. MLFF show a significant gain in classification accuracy by nearly 10% when \( s \) is 1 or 2. In addition, we find that the standard deviations of our method are smaller than other methods, which means MLFF are more robust.

Next, we verify the validity of FBLS-MD classifier. In the third experiment on this data set, we compare the performances of FBLS-MD classifier with the other four classifiers. The results are shown in Table 4 and Figure 7. From the experimental results with the same feature set and varied classifiers, we can see that the BLS and FBLS-MD turn out to be remarkably good at distinguishing activities. Obviously, only the result of our method exceeds 90% when \( s \) is 5.

At the final stage, the performance comparison of BLS and FBLS-MD on MSR Action 3D data set is shown in Table 5 (where \( \text{Ratio} \) refers to the time reduction proportion). When the feature nodes are small, the reduction of the training time is not obvious. But as the number of mapping nodes increases, the training time shrinks greatly while maintaining the recognition rate. As can be seen, when the feature nodes reach a certain number, the action recognition rate will reach the highest value and then slightly decrease.

MSR Hand Gesture 3D data set. MSR Hand Gesture 3D\textsuperscript{24} consists of 12 dynamic American Sign Language (ASL) gestures captured by a Kinect device. The whole data set contains 333 depth sequences and bears self-occlusions. Each gesture is performed for two or three times.
Table 3. Performance comparison for SNV, HON4D, and MLFF features on MSR Action 3D data set.

| Method       | s = 1       | s = 2       | s = 3       | s = 4       | s = 5       |
|--------------|-------------|-------------|-------------|-------------|-------------|
| SNV + SVM    | 60.78% ± 3.45% | 69.55% ± 4.65% | 77.66% ± 3.74% | 81.24% ± 5.23% | 88.01% ± 3.59% |
| SNV + BLS    | 63.01% ± 3.16% | 70.75% ± 3.55% | 80.51% ± 2.90% | 81.97% ± 5.43% | 89.26% ± 3.43% |
| HON4D + SVM  | 62.35% ± 4.73% | 74.45% ± 2.34% | 75.16% ± 4.28% | 79.81% ± 4.02% | 87.87% ± 4.08% |
| HON4D + BLS  | 66.26% ± 2.45% | 75.51% ± 2.73% | 79.40% ± 3.72% | 81.85% ± 4.70% | 88.53% ± 3.15% |
| MLFF + BLS-MD| 71.18% ± 1.78% | 79.62% ± 0.95% | 85.38% ± 1.52% | 85.40% ± 4.95% | 90.85% ± 1.98% |

SNV: super normal vector; HON4D: histogram of the surface normal orientation in four-dimensional space; MLFF: multi-level fused features; SVM: support vector machine; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition.

Values are expressed as means ± SD (Standard Deviation), while the best ones of each column are marked in bold respectively.

Table 4. Performance comparison under different classifiers on MSR Action 3D data set.

| Method       | s = 1       | s = 2       | s = 3       | s = 4       | s = 5       |
|--------------|-------------|-------------|-------------|-------------|-------------|
| MLFF + KNN   | 59.75% ± 2.21% | 69.29% ± 4.49% | 74.10% ± 1.89% | 74.51% ± 3.29% | 81.37% ± 4.53% |
| MLFF + Softmax| 69.17% ± 2.02% | 77.33% ± 1.82% | 83.19% ± 1.24% | 83.42% ± 4.29% | 89.63% ± 2.65% |
| MLFF + SVM   | 68.31% ± 3.10% | 78.66% ± 2.11% | 83.19% ± 1.24% | 84.38% ± 4.84% | 89.99% ± 2.89% |
| MLFF + ELM   | 70.99% ± 1.11% | 79.11% ± 1.55% | 83.70% ± 1.41% | 84.62% ± 4.90% | 88.68% ± 2.66% |
| MLFF + BLS   | 71.26% ± 1.71% | 79.73% ± 1.12% | 84.11% ± 2.55% | 85.40% ± 5.46% | 90.80% ± 2.42% |
| MLFF + BLS-MD| 71.18% ± 1.78% | 79.62% ± 0.95% | 85.38% ± 1.52% | 85.40% ± 4.95% | 90.85% ± 1.98% |

MLFF: multi-level fused features; KNN: k-nearest neighbor; SVM: support vector machine; ELM: extreme learning machine; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition.

Values are expressed as means ± SD, while the best ones of each column are marked in bold respectively.

Figure 7. Recognition rates of different methods with different numbers of training sample on MSR Action 3D data set.

by 10 actors; the depth map size of each gesture is varied. Some samples are shown in Figure 8.

In the experiments, we conducted the leave-one-subject-out cross-validation (LOO-CV) as in the study by Wang et al.\cite{Wang2013} to evaluate the performance of our algorithm. Table 6 shows the comparison of our proposed method with the state-of-the-art methods on this data set. We can see that our method has improved 4.49% and 2.19% compared with the single-feature methods in terms of HON4D and SNV, respectively. Moreover, our method outperforms all compared approaches.

Furthermore, our method achieves a high recognition rate of 96.05%. The confusion matrix is showed in Figure 9 with the experimental setup as in the study by Wang et al.,\cite{Wang2013} which refers to the first five actors for training and the rest for testing. Figure 10 gives the confusion matrix constructed through the method in Yang and Tian\cite{Yang2017} under the same experimental setup. From the two confusion matrices, it is clear that “blue,” “finish,” “green,” “hungry,” “milk,” “j,” and “z” gestures are more precisely identified, and the overall recognition rate was increased by nearly 7%.

We compare our proposed method with single-feature methods. We can find that our method is better than the single-feature methods presented in Table 7 with a large margin in small samples. Comparing with the best of the other methods, our method achieves a 2.59% recognition rate improvement when \( s = 4 \) and more than 4% improvement in other cases.

Table 8 and Figure 11 show the performance comparison of the five classifiers. Although the result of ELM is slightly higher than that of BLS and FBLS-MD when \( s \) equals 3, the overall results show that FBLS-MD significantly outperforms the other classifiers. The performance comparison of BLS and FBLS-MD algorithms on MSR Hand Gesture 3D data set is given in Table 9, which has also shown that FBLS-MD is useful for elevating the training speed.
The actions in 3D MSR Action Pair data set are the paired-activities captured by a depth camera. This data set contains 12 activities which were performed by 10 actors with each actor performing three times. Part of them are shown in Figure 12. Every couple of activities has similar movements and shapes. The challenge of this data set is that some activities differ only on sequences’ order, such as picking and dropping. Therefore, the temporal order of frames is one of the most important factor in the activity recognition of this data set.

As shown in Table 10, our proposed method outperforms the state-of-the-art methods on this data set. Table 11 indicates that the single-feature methods are still inferior to our method on the third data set. When the training samples are smaller, our method achieves a great improvement. The accuracies of our method are 5.25%, 2.36%, 0.95%, 0.64%, and 0.89% higher than the single-feature method in the best cases when $s$ is 1, 2, 3, 4, and 5, respectively. From Table 11, we can find that HON4D feature is more suitable than SNV feature on 3D Action Pairs data set. Accordingly, this also proves the complementarity of this two features for different data sets.

It is obvious that our method is superior to the other five classifiers as presented in Table 12 and Figure 13.

Table 5. Performance comparison of BLS and FBLS-MD algorithms on MSR Action 3D data set.

| Number of MN | Number of EN | BLS       | FBLS-MD    | Ratio (%) |
|--------------|--------------|-----------|------------|-----------|
|              |              | Accuracy (%) | Training time (s) | Accuracy (%) | Training time (s) |          |
| 100          | 5000         | 91.18     | 0.9356     | 91.27     | 0.7473     | 20.13    |
| 200          | 6000         | 92.00     | 1.6189     | 92.35     | 1.3020     | 19.58    |
| 400          | 6000         | 93.24     | 1.9666     | 93.09     | 1.5166     | 22.88    |
| 800          | 8000         | 94.06     | 4.8800     | 93.77     | 3.7002     | 24.18    |
| 1000         | 10,000       | 93.52     | 7.8780     | 93.86     | 6.0910     | 22.68    |
| 1500         | 10,000       | 92.36     | 10.546     | 92.33     | 8.9729     | 14.92    |
| 2000         | 10,000       | 92.40     | 15.9614    | 92.47     | 13.4259    | 15.89    |
| 3000         | 11,000       | 92.36     | 35.2919    | 92.73     | 31.5492    | 10.6     |
| 3500         | 11,000       | 92.58     | 47.5928    | 92.65     | 43.816     | 7.94     |
| 4000         | 10,000       | 92.65     | 59.5227    | 92.62     | 56.2620    | 5.48     |
| 4500         | 11,000       | 92.65     | 82.2300    | 92.73     | 77.2088    | 6.11     |
| 5000         | 10,000       | 92.69     | 105.5370   | 92.73     | 98.6860    | 6.49     |
| 5500         | 11,000       | 92.69     | 140.7690   | 92.73     | 132.7351   | 5.71     |
| 6000         | 11,000       | 92.73     | 172.8102   | 92.73     | 164.6193   | 4.74     |

BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition; MN: mapped nodes; EN: enhancement nodes.

Table 6. Performance comparison of the proposed method with the state-of-the-art methods on MSR Hand Gesture 3D data set.

| Method                                      | Accuracy (%) |
|---------------------------------------------|--------------|
| Action Graph on Occupancy$^{55}$            | 80.50        |
| Action Graph on Silhouette$^{55}$           | 87.70        |
| ROP$^{51}$                                  | 88.50        |
| DMM$^{19}$                                  | 89.20        |
| HON4D$^{10}$                                | 92.45        |
| DMM-LBP-FF$^{54}$                           | 93.40        |
| Histograms of Depth Gradients$^{56}$         | 93.60        |
| HOG3D + LLC$^{57}$                          | 94.10        |
| HPF + TM$^{47}$                             | 94.58        |
| DMM-LBP-DF$^{54}$                           | 94.60        |
| SNV$^{37}$                                  | 94.75        |
| MLFF + BLS                                  | 96.94        |
| MLFF + FBLS-MD                               | 96.67        |

HON4D: histogram of the surface normal orientation in four-dimensional space; DMM: depth motion maps; SNV: super normal vector; MLFF: multi-level fused features; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition; LBP-FF: local binary patterns—feature level fusion approach; HOG3D: histogram of oriented 3D; LLC: locality-constrained linear coding. The two highest classification accuracies are acquired with our methods, which are marked in bold.

3D Action Pairs data set. The actions in 3D MSR Action Pair data set are the paired-activities captured by a depth camera. This data set contains 12 activities which were performed by 10 actors with each actor performing three times. Part of them are shown in Figure 12. Every couple of activities has similar movements and shapes. The challenge of this data set is that some activities differ only on sequences’ order, such as picking and dropping. Therefore, the temporal order of frames is one of the most important factor in the activity recognition of this data set.

As shown in Table 10, our proposed method outperforms the state-of-the-art methods on this data set. Table 11 indicates that the single-feature methods are still inferior to our method on the third data set. When the training samples are smaller, our method achieves a great improvement. The accuracies of our method are 5.25%, 2.36%, 0.95%, 0.64%, and 0.89% higher than the single-feature method in the best cases when $s$ is 1, 2, 3, 4, and 5, respectively. From Table 11, we can find that HON4D feature is more suitable than SNV feature on 3D Action Pairs data set. Accordingly, this also proves the complementarity of this two features for different data sets.

It is obvious that our method is superior to the other five classifiers as presented in Table 12 and Figure 13.
However, with the third data set, Softmax, SVM, ELM, BLS, and FBLS-MD classifiers have slight differences in classification performance. The differences of the recognition rate are within 3%.

Table 13 shows the comparison of BLS and FBLS-MD algorithms' performances on 3D Action Pairs data set. On this data set, as the feature nodes continue to increase, the training time has decreased significantly, simultaneously the recognition rate has reached maximum and no longer drops. It also proves FBLS-MD's value when large-scale feature nodes are required to train models.
Discussion

Our proposed method obtains the highest performance while the training samples are small. It may be attributed to two factors. First, our fused features are

Table 8. Performance comparison under different classifiers on MSR Hand Gesture 3D data set.

| Method       | $s = 1$          | $s = 2$          | $s = 3$          | $s = 4$          | $s = 5$          |
|--------------|------------------|------------------|------------------|------------------|------------------|
| MLFF + KNN   | 67.41% ± 2.24%   | 73.39% ± 5.39%   | 78.65% ± 4.05%   | 82.38% ± 2.18%   | 87.07% ± 3.34%   |
| MLFF + Softmax| 70.03% ± 3.63%   | 81.39% ± 3.36%   | 85.48% ± 2.81%   | 88.96% ± 2.68%   | 92.59% ± 1.67%   |
| MLFF + SVM   | 67.41% ± 2.24%   | 81.89% ± 2.74%   | 86.44% ± 3.56%   | 90.67% ± 2.03%   | 93.74% ± 1.16%   |
| MLFF + ELM   | 64.11% ± 3.60%   | 81.48% ± 3.22%   | 87.61% ± 1.69%   | 90.04% ± 3.44%   | 93.51% ± 1.60%   |
| MLFF + BLS   | 71.18% ± 3.4%    | 83.65% ± 2.68%   | 87.37% ± 2.96%   | 90.90% ± 3.22%   | 94.68% ± 1.40%   |
| MLFF + FBLS-MD| 71.01% ± 3.22%   | 83.70% ± 2.38%   | 87.22% ± 2.89%   | 91.04% ± 2.59%   | 94.70% ± 1.17%   |

MLFF: multi-level fused features; KNN: k-nearest neighbor; SVM: support vector machine; ELM: extreme learning machine; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition.

Values are expressed as means ± SD, while the best ones of each column are marked in bold respectively.

Table 9. Performance comparison of BLS and FBLS-MD algorithms on MSR Hand Gesture 3D data set.

| Number of MN | Number of EN | BLS         | FBLS-MD       | Ratio (%) |
|--------------|--------------|-------------|---------------|-----------|
|              |              | Accuracy (%)| Training time (s) | Accuracy (%)| Training time (s) |
| 100          | 1000         | 93.22       | 0.0655        | 93.33      | 0.0611        | 6.72        |
| 200          | 2000         | 94.35       | 0.2004        | 94.12      | 0.1955        | 2.45        |
| 400          | 3000         | 96.05       | 1.6124        | 95.68      | 1.3096        | 18.78       |
| 600          | 5000         | 95.80       | 4.5436        | 96.05      | 3.4459        | 24.16       |
| 800          | 8000         | 95.48       | 15.4136       | 96.05      | 13.4471       | 12.76       |
| 1000         | 10,000       | 95.08       | 34.4396       | 94.28      | 5.8594        | 22.18       |
| 2000         | 20,000       | 94.08       | 81.8490       | 94.05      | 76.7648       | 6.21        |
| 3000         | 30,000       | 94.97       | 107.3245      | 94.16      | 100.2453      | 6.60        |
| 4000         | 40,000       | 94.97       | 144.8436      | 94.50      | 135.4084      | 6.51        |
| 5000         | 50,000       | 94.92       | 174.8452      | 94.11      | 164.6830      | 5.81        |

BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition; MN: mapped nodes; EN: enhancement nodes.

Figure 11. Recognition rates of different methods with different numbers of training sample on MSR Hand Gesture 3D data set.

Figure 12. The specific actions in 3D Action Pairs data set.
complementary to HON4D and SNV, thus increasing recognition accuracies. Second, the comparison experiments show that FBLS-MD performs very favorably against other commonly used classifiers. In addition, FBLS-MD is demonstrated to effectively shorten the training time in case that the computation burden is increased by a large number of feature nodes. With increasing feature nodes, the computation burden of the inverse of the growing matrix will be huger and matrix decomposition will play a vital role. There are large standard deviations for some random experimental results, which is attributed to the large individual differences in the data sets and the missing data in the first two data sets.

The adjustable parameters in FBLS-MD include the following: the feature nodes per window, number of windows of the feature nodes, number of EN, the $l_2$ regularization parameter, and the shrinkage scale of the EN. In the article, we have undergone extensive experiments by making specified some of the parameters unchanged and adjusting the number of MN and EN constantly to get the best recognition results.

Table 10. Performance comparison of the proposed method with the state-of-the-art methods on 3D Action Pairs data set.

| Method                  | s = 1       | s = 2       | s = 3       | s = 4       | s = 5       |
|-------------------------|-------------|-------------|-------------|-------------|-------------|
| Skeleton + LOP$^{13}$    | 63.33%      | 74.65%      | 82.94%      | 89.86%      | 90.56%      |
| DMM$^{19}$               | 66.11%      | 74.87%      | 83.89%      | 90.97%      | 91.67%      |
| Skeleton + LOP + Pyramid$^{13}$ | 70.99%     | 88.47%      | 92.54%      | 95.65%      | 95.67%      |
| HON4D$^{10}$             | 96.67%      | 92.13%      | 95.16%      | 97.04%      | 97.11%      |
| HPM + TM$^{17}$          | 98.22%      | 91.39%      | 92.29%      | 95.16%      | 97.04%      |
| SNV$^{37}$               | 98.89%      | 92.09%      | 95.16%      | 97.04%      | 97.11%      |
| MLFF + BLS               | 99.44%      | 92.09%      | 95.16%      | 97.04%      | 97.11%      |
| MLFF + FBLS-MD           | 99.44%      | 92.09%      | 95.16%      | 97.04%      | 97.11%      |

LOP: local occupancy patterns; DMM: depth motion maps; HON4D: histogram of the surface normal orientation in four-dimensional space; SNV: super normal vector; MLFF: multi-level fused features; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition.

Table 11. Performance comparison for SNV, HON4D, and MLFF on 3D Action Pairs data set.

| Method                  | s = 1       | s = 2       | s = 3       | s = 4       | s = 5       |
|-------------------------|-------------|-------------|-------------|-------------|-------------|
| SNV + SVM $^{37}$       | 63.33% ± 6.47% | 74.65% ± 7.72% | 82.94% ± 6.94% | 89.86% ± 3.46% | 90.56% ± 6.51% |
| SNV + BLS               | 65.18% ± 5.87% | 74.87% ± 7.64% | 83.89% ± 6.15% | 90.97% ± 3.06% | 91.67% ± 5.97% |
| HON4D + SVM             | 70.99% ± 6.04% | 88.47% ± 2.07% | 92.54% ± 2.50% | 95.65% ± 1.16% | 95.67% ± 0.73% |
| HON4D + BLS             | 80.37% ± 3.45% | 91.39% ± 2.29% | 95.16% ± 2.16% | 97.04% ± 0.62% | 97.11% ± 1.20% |
| MLFF + FBLS-MD          | 85.62% ± 4.64% | 93.75% ± 2.24% | 96.11% ± 1.95% | 97.68% ± 0.66% | 98.00% ± 0.93% |

SNV: super normal vector; HON4D: histogram of the surface normal orientation in four-dimensional space; MLFF: multi-level fused features; SVM: support vector machine; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition.

Table 12. Performance comparison under different classifiers on 3D Action Pairs data set.

| Method                  | s = 1       | s = 2       | s = 3       | s = 4       | s = 5       |
|-------------------------|-------------|-------------|-------------|-------------|-------------|
| MLFF + KNN              | 77.65% ± 5.31% | 82.71% ± 4.40% | 89.21% ± 1.91% | 90.09% ± 2.46% | 90.11% ± 2.30% |
| MLFF + Softmax          | 84.94% ± 5.04% | 91.08% ± 2.59% | 94.67% ± 2.01% | 97.22% ± 1.30% | 97.44% ± 1.22% |
| MLFF + SVM              | 85.06% ± 4.76% | 92.43% ± 2.48% | 94.52% ± 1.85% | 97.13% ± 0.83% | 97.44% ± 1.22% |
| MLFF + ELM              | 85.37% ± 3.92% | 92.78% ± 3.10% | 96.03% ± 1.40% | 97.41% ± 0.90% | 97.00% ± 1.34% |
| MLFF + BLS              | 85.62% ± 4.59% | 94.03% ± 1.50% | 96.11% ± 1.95% | 97.59% ± 0.76% | 97.89% ± 1.07% |
| MLFF + FBLS-MD          | 85.62% ± 4.64% | 93.75% ± 2.24% | 96.11% ± 1.95% | 97.68% ± 0.66% | 98.00% ± 0.93% |

MLFF: multi-level fused features; KNN: k-nearest neighbor; SVM: support vector machine; ELM: extreme learning machine; BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition.
The relationships between recognition rate and MN or EN of FBLS-MD on the three data sets are shown in Figure 14, where the first two data sets are verified with the same experimental setting as in the study by Wang et al.46 and the third data set with the set as in the study by Oreifej and Liu.10 The result demonstrated that

(a) As MN and EN increase, we can see that the recognition results also boost. However, when the number of nodes reach a certain value, the recognition rate will reach a maximum, then it will gradually decline.
(b) It will lead to low recognition rate with limited EN and MN. Meanwhile, the excessively abundant EN and MN will lead to additional computation. Therefore, we set MN and EN as 800–6000, 400–3000, and 400–6000 for the three data sets respectively. In addition, the $l_2$ regularization parameter and the shrinkage scale of the EN are 0.05 and 0.9.

**Conclusion**

In this article, we have presented a new method for HAR with depth videos. It consists of our proposed features called MLFF and a FBLS-MD. MLFF descriptors are designed to describe spatiotemporal and motion information more abundantly. Moreover, it is robust to noise and occlusion. FBLS-MD is proposed to effectively reduce the training time and obtain satisfied classification results. Extensive experiments have been performed on three benchmarks data sets to verify the effectiveness of our method. Experimental results have shown that our method outperforms state-of-the-art methods. It has also been demonstrated that our method holds an advantage with a small training set.

### Table 13. Performance comparison of BLS and FBLS-MD algorithms on 3D Action Pairs data set.

| Number of MN | Number of EN | BLS Accuracy (%) | BLS Training time (s) | FBLS-MD Accuracy (%) | FBLS-MD Training time (s) | Ratio (%) |
|--------------|--------------|------------------|-----------------------|-----------------------|--------------------------|-----------|
| 50           | 800          | 97.22            | 0.0411                | 97.22                 | 0.0395                   | 3.89      |
| 100          | 1000         | 98.33            | 0.0642                | 98.33                 | 0.0616                   | 4.05      |
| 200          | 2000         | 99.44            | 0.2055                | 99.44                 | 0.1913                   | 6.91      |
| 400          | 4000         | 99.44            | 0.8118                | 99.44                 | 0.7130                   | 12.17     |
| 600          | 8000         | 99.44            | 3.8409                | 99.44                 | 3.0007                   | 21.88     |
| 600          | 10,000       | 99.44            | 6.4703                | 99.44                 | 4.7522                   | 26.55     |
| 1500         | 10,000       | 99.44            | 10.3901               | 99.44                 | 9.0050                   | 13.33     |
| 2000         | 10,000       | 99.44            | 15.4797               | 99.44                 | 13.2392                  | 14.47     |
| 3000         | 10,000       | 99.44            | 34.7353               | 99.44                 | 31.6053                  | 9.01      |
| 4000         | 10,000       | 99.44            | 62.8090               | 99.44                 | 59.4614                  | 5.33      |
| 4500         | 11,000       | 99.44            | 82.8254               | 99.44                 | 77.4562                  | 6.48      |
| 5000         | 11,000       | 99.44            | 106.8392              | 99.44                 | 100.4823                 | 5.95      |
| 5500         | 12,000       | 99.44            | 140.1583              | 99.44                 | 100.4823                 | 5.95      |
| 6000         | 12,000       | 99.44            | 172.0283              | 99.44                 | 160.6063                 | 6.64      |

BLS: broad learning system; FBLS-MD: fast broad learning system based on matrix decomposition; MN: mapped nodes; EN: enhancement nodes.

**Figure 14.** The relationship between recognition rate, MN and EN. (a) MSR Action 3D data set, (b) MSR Hand Gesture 3D data set, and (c) on 3D Action Pairs data set.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work has been financially supported by the National Natural Science Foundation of China under Grant 61502195, the Natural Science Foundation of Hubei Province under Grant 2018CFB691, the Fundamental Research Funds for the Central Universities under Grants CCNU19QN023 and CCNU18QN020, and the Humanities and Social Sciences Foundation of the Ministry of Education under Grant 19YJJC880079.

ORCID iDs
Huang Yao https://orcid.org/0000-0001-5055-4106
Yantao Wei https://orcid.org/0000-0001-7225-1958

References
1. Gaikwad K and Patil MD. Human activity detection and recognition algorithm from video surveillance. Int J Technol Explor Learn 2013; 2(5): 2319–2315.
2. Lazar J, Feng JH and Hochheiser H. Research methods in human-computer interaction. Amsterdam: Morgan Kaufmann, 2017.
3. Chaaraoui AA, Climent-Pérez P and Flórez-Revuelta F. A review on vision techniques applied to human behaviour analysis for ambient-assisted living. Expert Syst Appl 2012; 39(12): 10873–10888.
4. Kyan M, Sun G, Li H, et al. An approach to ballet dance training through MS Kinect and visualization in a cave virtual reality environment. ACM T Intel Syst Tec 2015; 6(2): 23.
5. Laptev I, Marszalek M, Schmid C, et al. Learning realistic human actions from movies. In: Proceedings of the 2008 IEEE conference on computer vision and pattern recognition (CVPR 2008), Anchorage, AK, 23–28 June 2008, pp.1–8. New York: IEEE.
6. Sun J, Wu X, Yan S, et al. Hierarchical spatio-temporal context modeling for action recognition. In: Proceedings of the 2009 IEEE conference on computer vision and pattern recognition (CVPR 2009), Miami, FL, 20–25 June 2009 pp.2004–2011. New York: IEEE.
7. Turaga P, Chellappa R, Subrahmanian VS, et al. Machine recognition of human activities: a survey. IEEE T Circ Syst Vid 2008; 18(11): 1473–1488.
8. Chen C, Jafari R and Kehtarnavaz N. Improving human action recognition using fusion of depth camera and inertial sensors. IEEE T Hum: Mach Syst 2015; 45(1): 51–61.
9. Xia L, Chen C-C and Aggarwal JK. Human detection using depth information by Kinect. In: Proceedings of the 2011 IEEE Computer Society conference on computer vision and pattern recognition workshops (CVPRW), Colorado Springs, CO, 20–25 June 2011, pp.15–22. New York: IEEE.
10. Oreifej O and Liu Z. HON4D: histogram of oriented 4D normals for activity recognition from depth sequences. In: Proceedings of the 2013 IEEE conference on computer vision and pattern recognition (CVPR), Portland, OR, 23–28 June 2013, pp.716–723. New York: IEEE.
11. Chen C, Jafari R and Kehtarnavaz N. A survey of depth and inertial sensor fusion for human action recognition. Multimed Tools Appl 2017; 76(3): 4405–4425.
12. Vemulapalli R, Arrate F and Chellappa R. Human action recognition by representing 3D skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, 23–28 June 2014, pp.588–595. New York: IEEE.
13. Wang J, Liu Z, Wu Y, et al. Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of the 2012 IEEE conference on computer vision and pattern recognition (CVPR), Portland, OR, 23–28 June 2013, pp.2834–2841. New York: IEEE.
14. Xia L and Aggarwal JK. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: Proceedings of the 2013 IEEE conference on computer vision and pattern recognition (CVPR), Providence, RI, 16–21 June 2012, pp.1290–1297. New York: IEEE.
15. Chen C, Liu K and Kehtarnavaz N. Real-time human action recognition based on depth motion maps. J Real-Time Image Pr 2016; 12(1): 155–163.
16. Luo J, Wang W and Qi H. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In: Proceedings of the IEEE international conference on computer vision, Sydney, NSW, Australia, 1–8 December 2013, pp.1809–1816. New York: IEEE.
17. Koniusz P, Cherian A and Porikli F. Tensor representations via kernel linearization for action recognition from 3D skeletons. In: Proceedings of the European conference on computer vision, Amsterdam, 8–16 October 2016, pp.37–53. Cham: Springer.
18. Rahmani H, Mahmood A, Huynh D, et al. Histogram of oriented principal components for cross-view action recognition. IEEE T Pattern Anal 2016; 38(12): 2430–2443.
19. Yang X, Zhang C and Tian YL. Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on multimedia, Nara Japan, 27–31 October 2012, pp.1057–1060. New York: ACM.
20. Tang S, Wang X, Lv X, et al. Histogram of oriented normal vectors for object recognition with a depth sensor. In: Proceedings of the Asian conference on computer vision, Daejeon, Korea, 5–9 November 2012, pp.525–538. Berlin: Springer.
21. Yang X and Tian YL. Super normal vector for human activity recognition with depth cameras. IEEE T Pattern Anal 2017; 39(5): 1028–1039.
22. Ma J, Yu W, Liang P, et al. FusionGAN: a generative adversarial network for infrared and visible image fusion. Inform Fusion 2019; 48: 11–26.
23. Ma J, Wang X and Jiang J. Image super-resolution via dense discriminative network. IEEE T Ind Electron. Epub ahead of print 14 August 2019. DOI: 10.1109/TIE.2019.2934071.
24. Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, 23–28 June 2014, pp.1725–1732. New York: IEEE.
25. Simonyan K and Zisserman A. Two-stream convolutional networks for action recognition in videos. In: Ghahramani Z, Welling M, Cortes C, et al. (eds) Advances in neural information processing systems. San Diego, CA: Neural Information Processing Systems, 2014, pp.568–576.
26. Tran D, Bourdev L, Fergus R, et al. Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 7–13 December 2015, pp.4489–4497. New York: IEEE.
27. Oquab M, Bottou L, Laptev I, et al. Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 23–28 June 2014, pp.1717–1724. New York: IEEE.
28. Liu L, Shao L, Li X, et al. Learning spatio-temporal representations for action recognition: a genetic programming approach. IEEE T Cybernetics 2016; 46(1): 158–170.
29. Wang P, Li W, Gao Z, et al. Deep convolutional neural networks for action recognition using depth map sequences, 2015, https://arxiv.org/pdf/1501.04686.pdf
30. Philip Chen CL and Liu Z. Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. IEEE T Neur Net Learn 2018; 29(1): 10–24.
31. Dehuri S and Cho S-B. A comprehensive survey on functional link neural networks and an adaptive PSO-BP learning for CFLNN. Neural Comput Appl 2010; 19(2): 187–205.
32. Pao Y. Adaptive pattern recognition and neural networks. Boston, MA: Addison-Wesley, 1989.
33. Kong Y, Wang X, Cheng Y, et al. Hyperspectral imagery classification based on semi-supervised broad learning system. Remote Sens 2018; 10(5): 685.
34. Cunningham P and Delany SJ. k-nearest neighbour classifiers. Multi Classif Syst 2007; 34: 1–17.
35. Schultd C, Laptev I and Caputo B. Recognizing human actions: a local SVM approach. In: Proceedings of the 17th international conference on pattern recognition, 2004 (ICPR 2004), vol. 3, Cambridge, 26–26 August 2004, pp.32–36. New York: IEEE.
36. Huang G-B, Zhu Q-Y and Siew C-K. Extreme learning machine: theory and applications. Neurocomputing 2006; 70(1–3): 489–501.
37. Yang X and Tian YL. Super normal vector for activity recognition using depth sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, 23–28 June 2014, pp.804–811. New York: IEEE.
38. Lee H, Battle A, Raina R, et al. Efficient sparse coding algorithms. In: Schölkopf B, Platt JC and Hoffman T (eds) Advances in neural information processing systems. San Diego, CA: Neural Information Processing Systems, 2007, pp.801–808.
39. Gehler P and Nowozin S. On feature combination for multiclass object classification. In: Proceedings of the 2009 IEEE 12th international conference on computer vision, Kyoto, Japan, 29 September–2 October 2009, pp.221–228. New York: IEEE.
40. Cai Z, Wang L, Peng X, et al. Multi-view super vector for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, 23–28 June 2014, pp.596–603. New York: IEEE.
41. Wang H and Schmid C. Action recognition with improved trajectories. In: Proceedings of the 2013 IEEE international conference on computer vision (ICCV), Sydney, NSW, Australia, 8–12 April 2013, pp.3551–3558. New York: IEEE.
42. Ma J, Ma Y and Li C. Infrared and visible image fusion methods and applications: a survey. Inform Fusion 2019; 45: 153–178.
43. Ross A and Jain A. Information fusion in biometrics. Pattern Recogn Lett 2003; 24(13): 2115–2125.
44. Moore B. Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans Automat Contr 1981; 26(1): 17–32.
45. Li J, Hua C, Tang Y, et al. A fast training algorithm for extreme learning machine based on matrix decomposition. Neurocomputing 2016; 173: 1951–1958.
46. Wang J, Liu Z, Wu Y, et al. Learning actionlet ensemble for 3D human action recognition. IEEE T Pattern Anal 2014; 36(5): 914–927.
47. Rahmani H and Mian A. 3D action recognition from novel viewpoints. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, 27–30 June 2016, pp.1506–1515. New York: IEEE.
48. Li W, Zhang Z and Liu Z. Action recognition based on a bag of 3D points. In: Proceedings of the 2010 IEEE Computer Society conference on computer vision and pattern recognition workshops (CVPRW), San Francisco, CA, 13–18 June 2010, pp.9–14. New York: IEEE.
49. Xia L, Chen C-C and Aggarwal JK. View invariant human action recognition using histograms of 3D joints. In: Proceedings of the 2012 IEEE Computer Society conference on computer vision and pattern recognition workshops (CVPRW), Providence, RI, 16–21 June 2012, pp.20–27. New York: IEEE.
50. Vieira AW, Nascimento ER, Oliveira GL, et al. STOP: space-time occupancy patterns for 3D action recognition from depth map sequences. In: Proceedings of the Iberoamerican congress on pattern recognition, Buenos Aires, Argentina, 3–6 September 2012, pp.252–259. Berlin: Springer.
51. Wang J, Liu Z, Chorowski J, et al. Robust 3D action recognition with random occupancy patterns. In: Proceedings of the European conference on computer vision (ECCV 2012), Florence, 7–13 October 2012, pp.872–885. Berlin: Springer.
52. Chen C, Hou Z, Zhang B, et al. Gradient local auto-correlations and extreme learning machine for depth-based activity recognition. In: Proceedings of the
international symposium on visual computing, Las Vegas, NV, 14–16 December 2015, pp.613–623. Cham: Springer.

53. Li M and Leung H. Graph-based approach for 3D human skeletal action recognition. Pattern Recogn Lett 2017; 87: 195–202.

54. Chen C, Jafari R and Kehtarnavaz N. Action recognition from depth sequences using depth motion maps-based local binary patterns. In: Proceedings of the 2015 IEEE winter conference on applications of computer vision (WACV), Waikoloa, HI, 5–9 January 2015, pp.1092–1099. New York: IEEE.

55. Kurakin A, Zhang Z and Liu Z. A real time system for dynamic hand gesture recognition with a depth sensor. In: Proceedings of the 2012 20th European signal processing conference (EUSIPCO), Bucharest, 27–31 August 2012, pp.1975–1979. New York: IEEE.

56. Rahmani H, Mahmood A, Huyhn DQ, et al. Real time action recognition using histograms of depth gradients and random decision forests. In: Proceedings of the 2014 IEEE winter conference on applications of computer vision (WACV), Steamboat Springs, CO, 24–26 March 2014, pp.626–633. New York: IEEE.

57. Rahmani H, Huyhn DQ, Mahmood A, et al. Discriminative human action classification using locality-constrained linear coding. Pattern Recogn Lett 2016; 72: 62–71.