Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 5861435, 11 pages
https://doi.org/10.1155/2017/5861435

Research Article

Mining Key Skeleton Poses with Latent SVM for Action Recognition

Xiaoqiang Li,[1] Yi Zhang,[1] and Dong Liao[2]

[1]*School of Computer Engineering and Science, Shanghai University, Shanghai, China*
[2]*School of Mathematic and Statistics, Nanyang Normal University, Nanyang, China*

Correspondence should be addressed to Xiaoqiang Li; xqli@i.shu.edu.cn and Dong Liao; liaodong@nynu.edu.cn

Received 23 August 2016; Revised 8 November 2016; Accepted 15 December 2016; Published 23 January 2017

Academic Editor: Lei Zhang

Human action recognition based on 3D skeleton has become an active research field in recent years with the recently developed commodity depth sensors. Most published methods analyze an entire 3D depth data, construct mid-level part representations, or use trajectory descriptor of spatial-temporal interest point for recognizing human activities. Unlike previous work, a novel and simple action representation is proposed in this paper which models the action as a sequence of inconsecutive and discriminative skeleton poses, named as key skeleton poses. The pairwise relative positions of skeleton joints are used as feature of the skeleton poses which are mined with the aid of the latent support vector machine (latent SVM). The advantage of our method is resisting against intraclass variation such as noise and large nonlinear temporal deformation of human action. We evaluate the proposed approach on three benchmark action datasets captured by Kinect devices: MSR Action 3D dataset, UTKinect Action dataset, and Florence 3D Action dataset. The detailed experimental results demonstrate that the proposed approach achieves superior performance to the state-of-the-art skeleton-based action recognition methods.

1. Introduction

The task of automatic human action recognition has been studied over the last few decades as an important area of computer vision research. It has many applications including video surveillance, human computer interfaces, sports video analysis, and video retrieval. Despite remarkable research efforts and many encouraging advances in the past decade, accurate recognition of the human actions is still a quite challenging task [1].

In traditional RGB videos, human action recognition mainly focuses on analyzing spatiotemporal volumes and representation of spatiotemporal volumes. According to the variety of visual spatiotemporal descriptors, human action recognition work can be classified into three categories. The first category is local spatiotemporal descriptors. An action recognition method first detects interesting points (e.g., STIPs [2] or trajectories [3]) and then computes descriptors (e.g., HOG/HOF [2] and HOG3D [4]) based on the detected local motion volumes. These local features are then combined (e.g., bag-of-words) to represent actions. The second category is global spatiotemporal templates that represent the entire action. A variety of image measurements have been proposed to populate such templates, including optical flow and spatiotemporal orientations [5, 6] descriptors. Except the local and holistic representational method, the third category is mid-level part representations which model moderate portions of the action. Here, parts have been proposed which capture a neighborhood of spacetime [7, 8] or a spatial key frame [9]. These representations attempt to balance the trade-off between generality exhibited by small patches, for example, visual words, and the specificity by large ones, for example, holistic templates. In addition, with the advent of inexpensive RGB-depth sensors such as Microsoft Kinect [10], a lot of efforts have been made to extract features for action recognition in depth data and skeletons. Reference [11] represents each depth frame as a bag of 3D points along the human silhouette and utilizes HMM to model the temporal dynamics. Reference [12] learns semilocal features automatically from the data with an efficient random sampling approach. Reference [13] selects most informative joints based on the

Figure 1: Two athletes perform the same action (diving water) in different way.

discriminative measures of each joint. Inspired by [14], Seidenari et al. model the movements of the human body using kinematic chains and perform action recognition by Nearest-Neighbor classifier [15]. In [16], skeleton sequences are represented as trajectories in an n-dimensional space; then these trajectories are then interpreted in a Riemannian manifold (shape space). Recognition is finally performed using kNN classification on this manifold. Reference [17] extracts a sparse set of active joint coordinates and maps these coordinates to lower-dimensional linear manifold before training an SVM classifier. The methods above generally extract the spatial-temporal representation of the skeleton sequences with well-designed handcrafted features. Recently, with the developing of deep learning, several Recurrent Neural Networks (RNN) models have been proposed for action recognition. In order to recognize actions according to the relative motion between limbs and the trunk, [18] uses an end-to-end hierarchical RNN for skeleton-based action recognition. Reference [19] uses skeleton sequences to regularize the learning of Long Short Term Memory (LSTM), which is grounded via deep Convolutional Neural Network (DCNN) onto the video for action recognition.

Most of the above methods relied on entire video sequences (RGB or RGBD) to perform action recognition, in which spatiotemporal volumes were always selected as representative feature of action. These methods will suffer from sensitivity to intraclass variation such as temporal scale or partial occlusions. For example, Figure 1 shows that two athletes perform some different poses when diving water, which makes the spatiotemporal volumes different. Motivated by this case, the question we seek to answer in this paper is whether a few inconsecutive key skeleton poses are enough to perform action recognition. As far as we know, this is an unresolved issue, which has not yet been systematically investigated. In our early work [20], it has been proven that some human actions could be recognized with only a few inconsecutive and discriminative frames for RGB video sequences. Related to our work, very short snippets [9] and discriminative action-specific patches [21] are proposed as representation of specific action. However, in contrast to our method, these two methods focused on consecutive frame.

In this paper, a novel framework is proposed for action recognition in which key skeleton poses are selected as representation of action in RGBD video sequences. In order to make our method more robust to translation, rotation, and scaling, Procrustes analysis [22] is conducted on 3D skeleton joint data. Then, the pairwise relative positions of the 3D skeleton joints are computed as discriminative features to represent the human movement. Finally, key skeleton poses, defined as the most representative skeleton model of the action, are mined from the 3D skeleton videos with the help of latent support vector machine (latent SVM) [23]. In early exploration experiments, we noticed that the number of the inconsecutive key skeleton poses is no smaller than 4. During testing, the temporal position and similarity of each of the key poses are compared with the model of the action. The proposed approach has been evaluated on three benchmark datasets: MSR Action 3D [24] dataset, UTKinect Action dataset [25], and Florence 3D Action dataset [26]; all are captured with Kinect devices. Experimental results demonstrate that the proposed approach achieves better recognition accuracy than a few existing methods. The remainder of this paper is organized as follows. The proposed approach is elaborated in Section 2 including the feature extracting, key poses mining, and action recognizing. Experimental results are shown and analyzed in Section 3. Finally, we conclude this paper in Section 4.

2. Proposed Approach

Due to the large performance variation of an action, the appearance, temporal structure, and motion cues exhibit large intraclass variability. So selecting the inconsecutive and discriminative key poses is a promising method to represent the action. In this section, we answer the question of what are and how to find the discriminative key poses.

2.1. Definition of the Key Poses and Model Structure. The structure of the proposed approach is shown in Figure 2. Each action model is composed of a few key poses, and each key pose in the model will be represented by three parts: (1) a linear classifier $g_i(x)$ which can discriminate the key

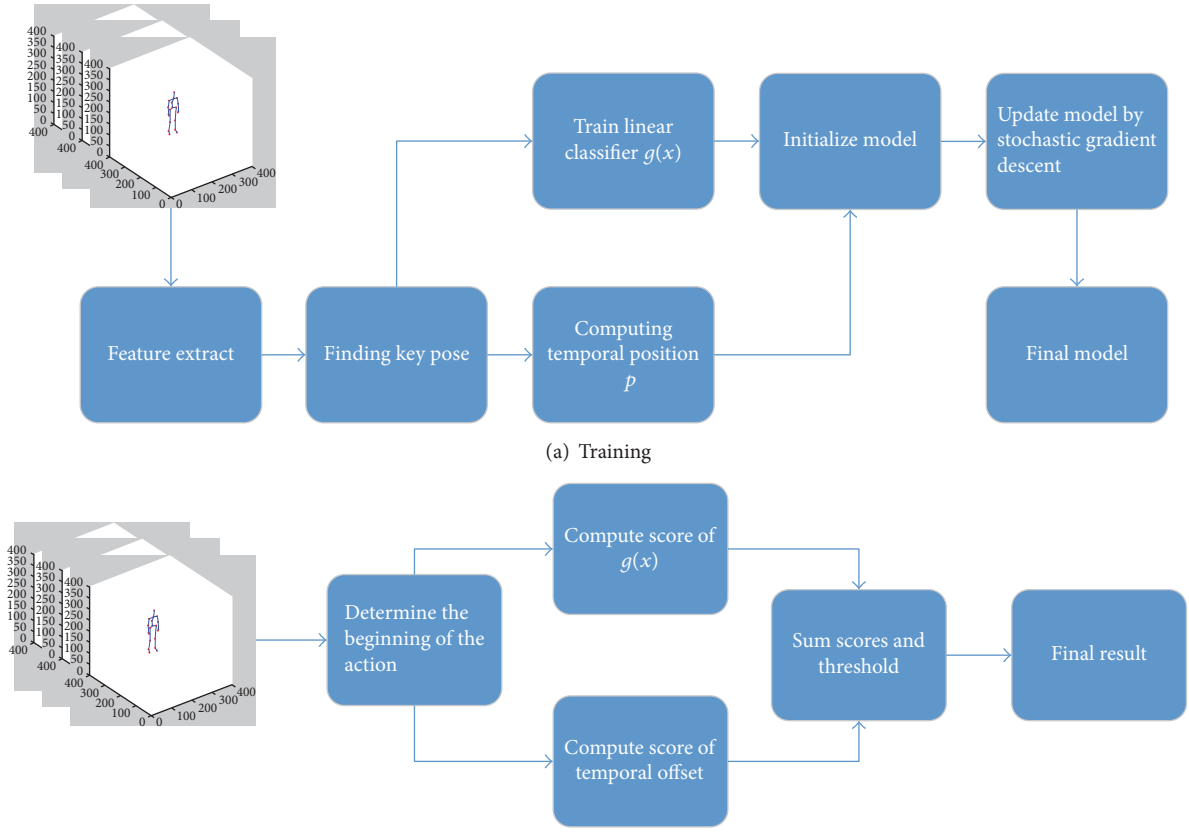

FIGURE 2: Structure of our model.

pose from the others, (2) the temporal position p_i and offset o_i, where the key poses i are most likely to appear in the neighborhood of p_i with radius o_i, and (3) the weight of linear classifier w_{g_i} and weight of the temporal information w_{p_i}.

Given is a video that contains m frames $X = \{x_1, \ldots, x_m\}$, where x_i is the i-th frame of the video. The score will be computed as follows:

$$f\left(X_{T^n}\right) = \max_{t \in T^n} \sum_{i=1}^{n} \left(w_{g_i} \times g\left(x_{t_i}\right) + w_{p_i} \times \Delta t_i\right), \quad (1)$$

in which X_{T^n} is the set of key poses of video X, $T^n = \{t \mid t = (t_1, \ldots, t_n),\ 1 \le t_i \le m\}$, and $x_{t_i} \in X_{T^n}$. For example, T^n is $\{1, 9, 10, 28\}$ in Figure 3(a). n is the total number of key poses in the action model; in our following experiment, n is ranging from 1 to 20. t_i is the serial number of the key pose in the sequence of frames of video. And Δt_i is defined as follows:

$$\Delta t_i = \frac{1}{2\pi o_i} \exp\left(\frac{-\left(t_i - t_0 - p_i\right)^2}{2{o_i}^2}\right), \quad (2)$$

in which t_0 is the frame at which action begins. Δt is a Gaussian function and reaches peak when $t_i - t_0 = p_i$. t_0 has been manually labeled on the training set. The method of finding t_0 in a testing will be discussed in Section 2.4.

2.2. Feature Extracting and Linear Classifier. With the help of real-time skeleton estimation algorithm, the 3D joint positions are employed to characterize the motion of the human body. Following the methods [1], we also represent the human movement as the pairwise relative positions of the joints.

For a human skeleton, joint positions are tracked by the skeleton estimation algorithm and each joint j has 3 coordinates at each frame. The coordinates are normalized based on Procrustes analysis [22], so that the motion is invariant to the initial body orientation and the body size. For a given frame $x = \{j_{1_x}, j_{1_y}, j_{1_z}, \ldots, j_{n_x}, j_{n_y}, j_{n_z}\}$, n is the number of joints. The feature of this frame $\varphi(x)$ is

$$\begin{aligned} \varphi(x) &= \left\{j_{a,b} \mid j_{a,b} = j_a - j_b,\ 1 \le a < b \le n\right\} \\ &\quad + \left\{j_{1_x}, j_{1_y}, j_{1_z}, \ldots, j_{n_x}, j_{n_y}, j_{n_z}\right\} \\ j_a - j_b &= \left\{j_{a_x} - j_{b_x}, j_{a_y} - j_{b_y}, j_{a_z} - j_{b_z}\right\}. \end{aligned} \quad (3)$$

And the feature is a 630-dimension (570 pairwise relative positions of the joint and 60 joint position coordinates) vector for MSR Action 3D and UTKinect Action dataset. AS for Florence 3D Action dataset, it is a 360-dimension vector. (The

(a) Key poses of drink are 1, 9, 10, and 28 (subject 1, action drink, video 4, total 32 frames)

(b) Key poses of stand up are 1, 3, 10, and 11 (subject 1, action stand up, video 21, total 29 frames)

(c) Key poses of wave are 4, 7, 8, and 12 (subject 2, action wave, video 1, total 13 frames)

(d) Key poses of drink are 5, 7, 9, and 11 (subject 2, action drink, video 3, total 14 frames)

Figure 3: Key poses for different action in Florence 3D Actions dataset.

selection of alternative feature representations will be discussed in Experiment Result.) Then, we train a linear classifier for each key pose according to the following equation:

$$g(x) = w \cdot \varphi(x). \tag{4}$$

The question of which frame should be used for training $g(x)$ will be discussed in Section 2.3.

2.3. Latent Key Poses Mining. It is not easy to decide which frames contain the key poses, because key poses' space T^n is too large to enumerate all the possible poses. Enlightened by [23], since the key pose positions are not observable in the training data, we formulate the learning problem as a latent structural SVM, regarding the key pose positions as the latent variable.

Rewrite (1) as follows:

$$\begin{aligned} f(X) &= \max_{t \in T^n} W \cdot \Phi(X, t) \\ W &= \left(w_{g_1}, w_{p_1}, \ldots, w_{g_n}, w_{p_n}\right) \\ \Phi(X, t) &= \left(g\left(x_{t_1}\right), \Delta t_1, \ldots, g\left(x_{t_n}\right), \Delta t_n\right), \end{aligned} \tag{5}$$

in which $t = (t_1, \ldots, t_n)$ is treated as the latent variable. Given a labeled set $D = \{\langle X_1, Y_1\rangle, \ldots, \langle X_i, Y_i\rangle, \ldots\}$, where $X_i = \{x_1, \ldots, x_m\}$ and $Y_i \in \{-1, +1\}$, the objective is to minimize the objective function:

$$L_D(W) = \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{n} \max\left(0, 1 - Y_i f(X_i)\right), \tag{6}$$

Require:
D_p, D_n, N;
$pos = 1$, $neg\_pose = \{x_i \mid i = random(),\ x_i \in X,\ X \in D_n\}$;
for $i = 1 \ldots N$ **do**
$o_i = 5$
$\varphi^{exp} = \varphi(x_{pos})$
where x_{pos} is the pos-th frame of the first video in D_p
for $X \in D_p$ **do**

$$pos\_pose = \left\{ pos\_pose,\ \arg\min_{x_j} \left(Euclidean\left(\varphi^{exp}, \varphi\left(x_j\right)\right)\right)\right\}$$

where $pos - o_i < j < pos + o_i,\ x_j \in X$
end for
Train $g_i(x)$ with pos_pose and neg_pose
$p_i = average\ \{j \mid x_j \in pos\_pose\}$
for $X \in D_p$ **do**
For each frame $x_j \in X$, $s[j] = s[j] + g_i(x_j)$
end for
$pos = \arg\min_j(s[j])$
end for
Training w_{g_i} and w_{p_i} with linear SVM

Algorithm 1

in which C is the penalty parameter. Following [23], the model is first initialized: D_p and D_n are the positive and negative subsets of D, and the model is initialized with N key frames as shown in Algorithm 1. In Algorithm 1, *pos_pose* and *neg_pose* are the positive frame set and the negative frame set, respectively. They are used to train the linear classifier $g(x)$. In order to initialize our model, we firstly compute $\varphi(x_{pos})$, the feature of the pos-th frame which belongs to the first video sample in D_p. Then the Euclidean distance between $\varphi(x_{pos})$ and the feature of the frames in other samples in the neighborhood of temporal position pos with radius o_i in D_p is computed. The frame which has the minimum Euclidean distance from $\varphi(x_{pos})$ in each sample is added in *pos_pose*. Then *pos_pose* is used to train the linear classifier $g_i(x)$ and choose p_i as the average of frame number in *pos_pose*. To select the next key pose, pos chose j with the minimum score based on $g_i(x)$ for next loop; in other words, the j-th frame which is most different from previous key pose is selected in the next loop. Finally, all w_{g_i} and w_{p_i} are trained with the linear SVM when Algorithm 1 is completed.

Once the initialization is finished, the model will be iteratively trained as follows. First, to find the optimal value t subjected to $t^{\text{opt}} \in T^n$ where $t^{\text{opt}} = \arg\max_t(W \cdot \Phi(X, t))$ for each positive video example and update p with the average value of all t^{opt}, the new linear classifier $g(x)$ is trained with modified p for each key pose. Second, (6) is optimized over W, where $f(x) = W \cdot \Phi(X, t^{\text{opt}})$ with stochastic gradient descent. Thus, the models are modified to better capture skeleton characteristics for each action.

2.4. Action Recognition with Key Poses. The key technical issue in action recognition in real-world video is that we do not know where the action starts, and searching start position in all possible places takes a lot of time. Fortunately, the score of each possible start position can be computed, respectively. So a parallel tool such as OpenMP or CUDA might be helpful.

Given a test video X with m frames, first, the skeleton feature score $g(x)$ of each frame has been computed in advance so we could reuse them later. Then for each possible action start position t_0, we compute the score of each key pose x_{t_i} according to the following equation:

$$\text{score} = \max_{t_i \geq t_0} \left(w_{g_i} \times g\left(x_{t_i}\right) + w_{p_i} \times \Delta t_i \right). \tag{7}$$

These scores are summed together as the final score of t_0. If the final score is bigger than the threshold, then an action beginning at t_0 has been detected and recognized. Figure 3 shows key poses for different actions in Florence 3D Action dataset.

3. Experiment Result

This section presents all experimental results. First, trying to eliminate the noise generated by translation, scale, and rotation changes of skeleton poses, we preprocess the dataset with Procrustes analysis [22]. And we conduct the experiment for action recognition with or without Procrustes analysis on UTKinect dataset to demonstrate effectiveness of Procrustes analysis. Second, the appropriate feature extraction was selected from four existing feature extraction methods according to experimental result on Florence 3D Action dataset. Third, quantitative experiment is conducted to select the number of inconsecutive key poses. Last, we evaluate our model and compare it with some state-of-the-art method on three benchmark datasets: MSR Action 3D dataset, UTKinect Action dataset, and Florence 3D Action dataset.

3.1. Datasets

(1) Florence 3D Action Dataset. Florence 3D Action dataset [26] was collected at the University of Florence during 2012 and captured using a Kinect camera. It includes 9 activities; 10 subjects were asked to perform the above actions for two or three times. This resulted in a total of 215 activity samples. And each frame contains 15 skeleton joints.

(2) MSR Action 3D Dataset. MSR Action 3D dataset [11] consists of the skeleton data obtained by depth sensor similar to the Microsoft Kinect. The data was captured at a frame rate of 15 frames per second. Each action was performed by 10 subjects in an unconstrained way for two or three times. The set of actions included *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two-hand wave*, *side boxing*, *forward kick*, *side kick*, *jogging*, *tennis swing*, and *tennis serve*.

(3) UTKinect Action Dataset. UTKinect Action dataset [24] was captured using a single stationary Kinect and contains 10 actions. Each action is performed twice by 10 subjects in indoor setting. Three synchronized channels (RGB, depth, and skeleton) are recorded with a frame rate of 30 frames per second. The 10 actions are *walk*, *sit down*, *stand up*, *pick up*, *carry*, *throw*, *push*, *pull*, *wave hands*, and *clap hands*. It is a challenging dataset due to the huge variations in view point and high intraclass variations. So, this dataset is used to validate the effectiveness of Procrustes analysis [22].

3.2. Data Preprocessing with Procrustes Analysis.

Skeleton data in each frame of a given video usually consists of a fixed number of predefined joints. The position of joint is determined by three coordinates (x, y, z). Figure 4 shows the skeleton definition in MSR Action 3D dataset. It contains 20 joints which could be represented by their coordinates. Regarding raw human skeleton in the video as the features is not a good choice in consideration of the nature of skeleton—rotation, scaling, and translation. So, before the experiment, we should normalize the datasets by Procrustes analysis.

In statistics, Procrustes analysis is a form of statistical shape analysis used to analyze the distribution of a set of shapes and is widely applied to the field of computer vision such as face detection. In this paper, it is used to align the skeleton joints and eliminate the noise owed to rotation, scaling, or translation. Details of Procrustes analysis will be depicted next.

Given a skeleton data with k joints $((x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_k, y_k, z_k))$, the first step is to process the joints with translation transformation. We compute the mean coordinates $(\overline{x}, \overline{y}, \overline{z})$ of all joints and put them on the origin of coordinates. The translation is completed after each joint coordinate subtracting the mean coordinate, denoted as equation $(x_i, y_i, z_i) = (x_i - \overline{x}, y_i - \overline{y}, z_i - \overline{z})$. The purpose of scaling is making mean square root of all joint coordinates equivalent to 1. For the skeleton joints, we compute s according to the following equation:

$$s = \sqrt{\frac{x_1^2 + y_1^2 + z_1^2 + \cdots + x_k^2 + y_k^2 + z_k^2}{k}}. \quad (8)$$

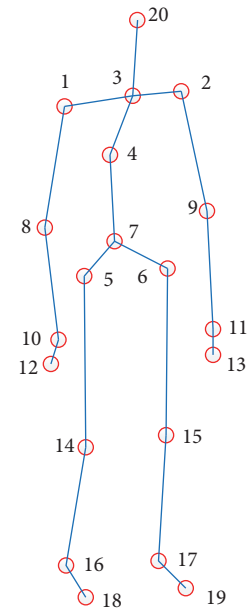

Figure 4: Skeleton of MSR Action 3D.

And the scaling result is calculated as follows: $(x_i, y_i, z_i) = (x_i/s, y_i/s, z_i/s)$. The rotation of skeleton is the last step of Procrustes analysis. Removing the rotation is more complex, as standard reference orientation is not always available. Given is a group of standard skeleton joint points $A = ((u_1, v_1, w_1), (u_2, v_2, w_2), \ldots, (u_k, v_k, w_k))$, which represent an action *stand* facing positive direction of x-coordinate axis. The mean coordinate of A is put on the origin of coordinate and the mean square root of coordinate is 1. Then we compute the rotation matrix R for skeleton $B = ((x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_k, y_k, z_k))$ which has been scaled and transformed as aforementioned method by (9), in which M is $3 * 3$ matrix. $U\Sigma V^T$ is the singular value decomposition with orthogonal U and V and diagonal Σ. And the rotation matrix R is equal to matrix V multiplied by the matrix transform of U. At last, skeleton joint points B can be aligned with A through computing R multiplied by B.

$$\begin{aligned} M &= B^T A \\ M &= U\Sigma V^T \\ R &= VU^T. \end{aligned} \quad (9)$$

We followed the cross-subject test setting of [30] on UTKinect dataset to test the validity of Procrustes analysis. Result is shown in Table 1. It is easy to see that the recognition rate of almost all actions is improved after preprocessing skeleton joint point with Procrustes analysis. In particular, the recognition rate of action *walk* is improved by 10%. It turned out that the translation, scaling, and rotation of human action skeleton in the video affect the recognition accuracy and Procrustes analysis is an effective method to eliminate the influence of geometry transformation.

Table 1: Results of action recognition with or without Procrustes analysis.

	Walk	Sit down	Stand up	Pick up	Carry	Throw	Push	Pull	Wave	Clap
with PA	**93%**	**86%**	**91%**	**97%**	**92%**	**88%**	87%	**91%**	**99%**	**91%**
without PA	83%	84%	85%	92%	89%	83%	87%	82%	93%	87%

3.3. Feature Extraction Method Selection. With the deep research on action recognition based on skeleton, there are many efficient feature representations. We select four of them (Pairwise [1], the most informative sequences of joint angles (MIJA) [31], histograms of 3D joints (HOJ3D) [24], and sequence of the most informative joints (SMIJ) [13]) as alternative feature representations.

Given is a skeleton $x = \{j_1, j_2, \ldots, j_n\}$, in which $j_i = (j_{x_i}, j_{y_i}, j_{z_i})$. The Pairwise representation is computed as follows: for each joint a, we extract the pairwise relative position features by taking the difference between the position of joint a and the position of another joint b: $j_{ab} = j_a - j_b$, so the feature of x is $\varphi(x) = \{j_{ab} \mid j_{ab} = j_a - j_b,\ 1 \le a < b \le n\}$. Due to the informativeness of the original joints, we made an improvement on this representation by concatenating $\varphi(x)$ and x. Then the new feature is $\varphi(x) = \{j_{a,b} \mid j_{a,b} = j_a - j_b,\ 1 \le a < b \le n\} + \{j_1, j_2, \ldots, j_n\}$.

The most informative sequences of joint angles (MIJA) representation regards joint angle as features. The shape of trajectories of joints encodes local motion patterns for each action. It chooses to use 11 out of the 20 joints capturing information for an action and center the skeleton, using the hip center joint as the origin $(0, 0, 0)$ of the coordinate system. From this origin, vectors to the 3D position of each joint are calculated. For each vector, it computes the angle θ_1 of its projection onto the x-z plane with the positive x-axis and the angle θ_2 between the vector and y-axis. The feature consists of the 2 angles of each joint.

Histograms of 3D joints (HOJ3D) representation chooses 12 discriminative joints of 20 skeletal joints. It takes the hip center as the center of the reference coordinate system and defines x-direction according to left and right hip. The remaining 8 joints are used to compute the 3D spatial histogram. The Spherical Coordinates space is partitioned to 84 bins. And for each joint location, a Gaussian weight function is used for the 3D bins. Counting the votes in each bin and concatenating them, we can get an 84-dimension feature vector.

Sequence of the most informative joints (SMIJ) representation also takes the joint angle as feature but it is different from MIJA. It partitions the joint angle time series of an action sequence into a number of congruent temporal segments and computes the variance of the joint angle time series of each joint over each temporal segment. The top 6 most variable joints in each temporal segment are selected to extract features with mapping function Φ. Here $\Phi(a) : \mathbb{R}^{|a|} \to \mathbb{R}$ is a function that maps a time series of scalar values to a single scalar value.

In order to find the optimal feature, we conduct an experiment on Florence 3D Action dataset, in which each video is short. And we estimate other 5 joints coordinates from original 15 joints of each frame in Florence dataset to make the same joints number of each frame as MSR Action 3D or UTKinect dataset. The experiment takes cross-subject test settings; one half of the dataset is used to train the key pose model and the other is used for testing. The model has 4 key poses and Procrustes analysis has been done before the feature extracting. Results are shown in Figure 5. The overall accuracy of Pairwise feature across 10 actions is better than SMIJ and MIJA. And it is observed that, for all actions except sit down and stand up, the Pairwise representation shows promising results. So, in following experiment, we select Pairwise feature to conduct action recognition experiment. The estimated joints coordinates generate more noise, so the accuracy is lower than the results on original Florence 3D Action dataset (shown in Table 6).

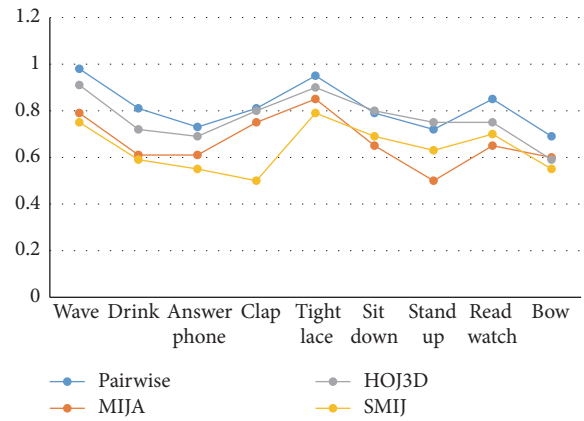

Figure 5: Features selection.

3.4. Selection of Key Pose Numbers. In this section, we implement some experiments to determine how many key poses are necessary for action recognition. The experimental results are shown in Figure 6; the horizontal axis denotes the number of key poses, and the vertical axis denotes recognition accuracy of the proposed approach. The number of key poses ranges from 1 to 20. We can see that the accuracy increases with the number of key poses when the number is less than 4. The accuracy almost achieves maximum values when the number of key poses equals 4, and the accuracy does not increase when the number of key poses is more than 4. To consider the accuracy and computation time, 4 is selected as the number of key poses for recognition action in our following experiment.

Table 2 only enumerates recognition accuracy for each action in UTKinect Action dataset when the number of key poses ranges from 4 to 8. It can be seen that the recognition accuracy varies with different key poses number for one action. However, the average recognition accuracy is nearly

Table 2: Recognition accuracy on different number of key poses.

Number	Carry	Clap	Pick	Pull	Push	Sit	Stand	Throw	Walk	Wave	Average
4	0.960	0.870	0.900	0.980	0.930	0.850	0.890	0.890	0.970	0.920	0.915
5	0.910	0.860	0.910	0.970	0.920	0.840	0.900	0.910	0.980	0.930	0.913
6	0.910	0.890	0.920	0.970	0.920	0.910	0.880	0.890	0.980	0.960	0.923
7	0.920	0.870	0.890	0.970	0.940	0.900	0.910	0.890	0.980	0.940	0.921
8	0.900	0.860	0.900	0.990	0.920	0.900	0.920	0.900	0.980	0.940	0.921

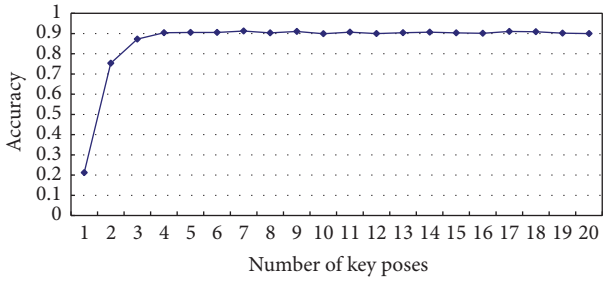

Figure 6: How many key poses does the model need?

Table 3: The three subsets of actions used in the experiments.

AS1	AS2	AS3
Bend	Draw circle	Forward kick
Forward punch	Draw tick	Golf swing
Hammer	Draw X	High throw
Hand clap	Forward kick	Jogging
High throw	Hand catch	Pick & throw
Horizontal arm wave	High arm wave	Side kick
Pickup & throw	Side boxing	Tennis serve
Tennis serve	Two-hand wave	Tennis swing

the same with different key poses number, so 4 is the high cost-effective choice.

3.5. Results on MSR Action 3D Dataset. According to the standard protocol provided by Li et al. [11], the dataset was divided into three subsets, shown in Table 3. AS1 and AS2 were intended to group actions with similar movement, while AS3 was intended to group complex actions together. For example, action *hammer* is likely to be confused with *forwardpunch* in AS1 and action *pickup* & *throw* in AS3 is a composition of *bend* and *high throw* in AS1.

We evaluate our method using a cross-subject test setting: videos of 5 subjects were used to train our model and videos of other 5 subjects were used for test procedure. Table 4 illustrates results for AS1, AS2, and AS3. We compare our performance with Li et al. [11], Xia et al. [24], and Yang and Tian [25]. We can see that our algorithm achieves considerably higher recognition rate than Li et al. [11] in all the testing setups on AS1, AS2, and AS3. For AS2, the accuracy rate of the proposed method is the highest. For AS1 or AS3, our recognition rate is only slightly lower than Xia et al. [24] or Yang and Tian [25], respectively. However, the average accuracy of our method on all three subsets is higher than the other methods.

Table 4: Comparison of our method with the others on AS1, AS2, and AS3.

Action subset	Li et al. [11]	Xia et al. [24]	Yang and Tian [25]	Ours
AS1	72.9%	**89.8**%	80.5%	89.1%
AS2	71.9%	85.5%	73.9%	**88.7**%
AS3	79.2%	63.5%	**95.5**%	94.9%
Average	74.7%	79.6%	83.3%	**90.9**%

Table 5: Comparison of our method with the others on MSR Action 3D.

MSR Action 3D	
Histogram of 3D joints [24]	78.97%
EigenJoints [25]	82.30%
Angle similarities [27]	83.53%
Actionlet [1]	88.20%
Spatial and temporal part-sets [28]	90.22%
Covariance descriptors [29]	90.53%
Our approach	**90.94**%

Table 5 shows the results on MSR Action 3D dataset. The average accuracy of the proposed method achieves 90.94%. It is easy to see that our method performs better than the other six methods.

3.6. Results on UTKinect Action Dataset. On UTKinect dataset, we followed the cross-subject test setting of [30], in which one half of the subjects is used for training our model and the other is used to evaluate the model. And we compare our model with Xia et al. [24] and Gan and Chen [30]. Figure 7 summarizes the results of our model along with competing approaches on UTKinect dataset. We can see that our method achieves the best performance on three actions such as pull, push, and throw. And the most important thing is that the average accuracy of our method achieves 91.5% and is better than the other two methods (90.9% and 91.1% for Xia et al. [24] and Gan and Chen [30], resp.). The accuracy of actions such as *clap hands* and *wave hands* is not so good; the reason may be the fact that the skeleton joint movement ranges of these actions are not large enough and the skeleton data contain more noise. So, it hinders our method from finding the optimal key poses and degrades the accuracy.

3.7. Result on Florence 3D Actions Dataset. We follow the leave-one-actor-out protocol which is suggested by dataset

Table 6: Results on Florence 3D Actions dataset.

Subject	1	2	3	4	5	6	7	8	9	10	Average
Wave	0.79	0.83	0.81	0.95	0.95	0.78	0.95	0.83	0.90	0.87	0.87
Drink	0.66	0.83	0.48	0.84	0.70	0.68	0.68	0.87	0.85	0.82	0.74
Answer	0.79	1.00	0.68	0.89	0.65	0.86	0.94	0.96	0.80	0.78	0.84
Clap	1.00	1.00	0.95	0.84	1.00	0.91	1.00	0.92	1.00	0.78	0.94
Tight	0.97	0.94	0.95	1.00	0.95	0.86	0.95	0.92	1.00	0.95	0.95
Sit down	0.72	0.89	0.90	0.90	0.76	0.86	0.79	1.00	0.80	0.91	0.85
Stand up	1.00	0.83	1.00	0.90	0.90	0.90	0.84	0.88	0.95	0.96	0.92
Read watch	0.59	0.89	0.90	0.84	0.75	0.82	0.68	0.75	0.85	0.73	0.78
Bow	0.86	0.89	0.86	1.00	0.85	1.00	1.00	0.96	0.90	0.74	0.91
Average	0.82	0.90	0.84	0.91	0.83	0.85	0.87	0.90	0.89	0.84	0.87

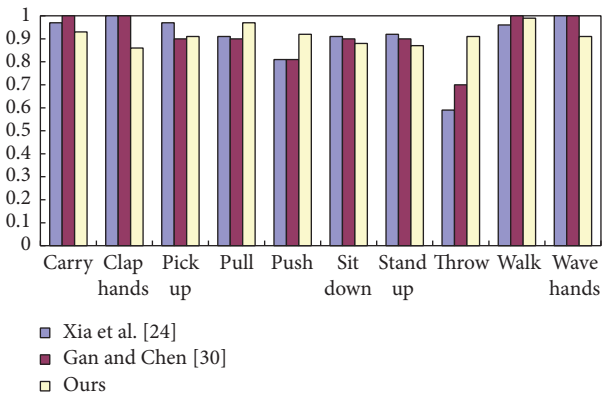

Figure 7: Results on UTKinect Action dataset.

collector on original Florence 3D Action dataset. All the sequences from 9 out of 10 subjects are used for training, while the remaining one is used for testing. For each subject, we repeat the procedure and average the 10 classification accuracy values at last. For comparison with other methods, average action recognition accuracy is also computed. The experimental results are shown in Table 6. In each column, the data represent each action's recognition accuracy, while the corresponding subject is used for testing. The challenges of this dataset are the human-object interaction and the different ways of performing the same action. By analyzing the experiment result of our method, we can notice that the proposed approach obtains high accuracies for most of the actions. Our method overcomes the difficulty of intraclass variation such as bow and clap. The proposed approach gets lower accuracies for the actions such as answer the phone and read watch; this can be explained by the fact that these actions are human-object interaction with small range of motion and the Pairwise feature could not well reflect the motion. Furthermore, results compared with other methods are listed in Table 7. It is clear that our average accuracy is better than Seidenari et al. [15] and is the same as Devanne et al. [16].

Table 7: Comparing of our method with the others on Florence 3D Actions dataset.

Florence 3D Actions		
Seidenari et al. [15]	Devanne et al. [16]	Our approach
82%	**87**%	**87**%

4. Conclusion

In this paper, we presented an approach for action recognition based on skeleton by mining the key skeleton poses with latent SVM. Experimental results demonstrated that human actions can be recognized by only a few frames with key skeleton pose; in other words, a few inconsecutive and representative skeleton poses can describe the video action. Starting from feature extraction using the pairwise relative positions of the joints, the positions of key poses are found with the help of latent SVM. Then the model is iteratively trained with positive and negative video examples. In test procedure, a simple method is given by computing the score of each start position to recognize the action.

We validated our model on three benchmark datasets: MSR Action 3D dataset, UTKinect Action dataset, and Florence 3D Action dataset. Experimental results demonstrated that our method outperforms all other methods. Because our method relies on extracting descriptors of simple relative positions of the joints, its performance degrades when the actions are little varied and uninformative, for instance, those actions that were performed only by forearm gestures such as *clap hands* in UTKinect Action dataset. In the future, we will explore the other local features reflecting minor motion for better understanding human action.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

[1] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 914–927, 2014.

[2] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

[3] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[4] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the 19th British Machine Vision Conference (BMVC '08)*, p. 275, British Machine Vision Association, 2008.

[5] K. G. Derpanis, M. Sizintsev, K. J. Cannons, and R. P. Wildes, "Action spotting and recognition based on a spatiotemporal orientation analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 527–540, 2013.

[6] S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1234–1241, IEEE, Providence, RI, USA, June 2012.

[7] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, June 2008.

[8] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, June 2012.

[9] K. Schindler and L. Van Gool, "Action snippets: how many frames does human action recognition require?" in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

[10] "kinect—australia," http://www.xbox.com/en-AU/Kinect.

[11] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '10)*, pp. 9–14, IEEE, San Francisco, Calif, USA, June 2010.

[12] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part II*, pp. 872–885, Springer, Berlin, Germany, 2012.

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

[14] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '06)*, pp. 137–146, Vienna, Austria, September 2006.

[15] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo, and P. Pala, "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '13)*, pp. 479–485, Portland, Ore, USA, June 2013.

[16] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "3-D human action recognition by shape analysis of motion trajectories on riemannian manifold," *IEEE Transactions on Cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015.

[17] T. Batabyal, T. Chattopadhyay, and D. P. Mukherjee, "Action recognition using joint coordinates of 3D skeleton data," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '15)*, pp. 4107–4111, IEEE, Québec, Canada, September 2015.

[18] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton-based action recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3010–3022, 2016.

[19] B. Mahasseni and S. Todorovic, "Regularizing long short term memory with 3D human-skeleton sequences for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 3054–3062, Las Vegas, NV, USA, June 2016.

[20] X. Li and Q. Yao, "Action detection based on latent key frame," in *Biometric Recognition*, pp. 659–668, Springer, Berlin, Germany, 2015.

[21] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: a strongly-supervised representation for detailed action understanding," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, Sydney, Australia, December 2013.

[22] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[24] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '12)*, pp. 20–27, June 2012.

[25] X. Yang and Y. Tian, "Effective 3D action recognition using EigenJoints," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 2–11, 2014.

[26] "Florence 3d actions dataset," http://www.micc.unifi.it/vim/datasets/3dactions/.

[27] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 915–922, IEEE, Portland, Ore, USA, June 2013.

[28] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 915–922, Portland, Ore, USA, June 2013.

[29] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*, pp. 2466–2472, Beijing, China, August 2013.

[30] L. Gan and F. Chen, "Human action recognition using APJ3D and random forests," *Journal of Software*, vol. 8, no. 9, pp. 2238–2245, 2013.

[31] H. Pazhoumand-Dar, C.-P. Lam, and M. Masek, "Joint movement similarities for robust 3D action recognition using skeletal data," *Journal of Visual Communication and Image Representation*, vol. 30, article no. 1493, pp. 10–21, 2015.