Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01403-5

ORIGINAL ARTICLE

An optimization method of human skeleton keyframes selection for action recognition

Hao Chen[1,2] · Yuekai Pan[2] · Chenwu Wang[3,4]

Received: 23 January 2022 / Accepted: 22 February 2024

Abstract

In the action recognition field based on the characteristics of human skeleton joint points, the selection of keyframes in the skeleton sequence is a significant issue, which directly affects the action recognition accuracy. In order to improve the effectiveness of keyframes selection, this paper proposes inflection point frames, and transforms keyframes selection into a multi-objective optimization problem based on it. First, the pose features are extracted from the input skeleton joint point data, which used to construct the pose feature vector of each frame in time sequence; then, the inflection point frames in the sequence are determined according to the flow of momentum of each body part. Next, the pose feature vectors are input into the keyframes multi-objective optimization model, with the fusion of domain information and the number of keyframes; finally, the output keyframes are input to the action classifier. To verify the effectiveness of the method, the MSR-Action3D, the UTKinect-Action and Florence3D-Action, and the 3 public datasets, are chosen for simulation experiments and the results show that the keyframes sequence obtained by this method can significantly improve the accuracy of multiple action classifiers, and the average recognition accuracy of the three data sets can reach 94.6%, 97.6% and 94.2% respectively. Besides, combining the optimized keyframes with deep learning classifier on the NTU RGB + D dataset can make the accuracies reaching 83.2% and 93.7%.

Keywords Human action recognition · Keyframes selection · Inflection point frame · Multi-objective optimization

Introduction

Human action recognition has broad application prospects and potential economic value in the fields of public security, human–computer interaction, sports and healthcare. In recent years, this direction has gradually become a research hotspot in the field of computer vision [1]. Traditional human action recognition methods are mainly specific to image data. With the development of sensor technology, the acquisition of high-precision skeleton joint point information has become convenient and feasible. Relatively speaking, skeleton posture has its inherent advantages in describing behaviour, which can describe human posture and motion state more accurately and is not affected by factors, such as background and illumination. On the other hand, the skeleton joint point data can be divided into two-dimensional (2D) skeleton and three-dimensional (3-D) skeleton. And the quantity of the data of 2D skeleton is smaller than that of 3-D skeleton, but it does not provide depth information and is more robust to action recognition under cross-view angles. Elias et al. [2] found that the accuracy of action recognition based on 3-D skeleton features can be increased by about 10% relative to 2D skeleton features. Therefore, 3-D skeleton features are more valuable [3]. Meanwhile, relevant research also showed that some keyframes were extracted from the skeleton frame sequence could further improve the accuracy of

✉ Hao Chen
chenhao@xupt.edu.cn

Yuekai Pan
kay535@foxmail.com

Chenwu Wang
wchenwu@xupt.edu.cn

[1] School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, Shaanxi, China

[2] Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an University of Posts and Telecommunications, Xi'an 710121, Shaanxi, China

[3] School of Modern Post, Xi'an University of Posts & Telecommunications, Xi'an 710061, China

[4] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China

Published online: 30 March 2024

action recognition [4]. However, the traditional keyframes selection methods, such as the clustering method, usually ignore the time sequence of action, and the number of categories in the clustering is not easy to be automatically processed. To solve this problem, this paper proposes a keyframes selection algorithm for action recognition. The main contributions of this paper are as follows:

(1) This work proposes to transform the keyframes selection problem into a multi-objective optimization problem, and introduces a multi-objective binary difference evolution algorithm for frame sequence coding and keyframes selection and optimization.

(2) This work proposes the concept of inflection point frames, uses the inflection point frames as the identification and redefines the population initialization rules, besides utilize the inflection point frames as the background knowledge to generate the random vector of the mutation, which can improve the global search ability of the algorithm.

(3) An evaluation model based on fusion domain information and the number of keyframes is proposed, which can adaptively adjust the number of keyframes according to the compression ratio as well as fully preserve the time sequence of action.

Related work

Traditional human action recognition method

There have been a number of image-based human action recognition methods in early years, and have achieved remarkable results. Davis et al. [5] utilized contour-based motion energy graph and dynamic history graph to express behavioural features, and used Mahalanobis Distance to classify actions. However, the model was limited by several critical factors in complex scenes; hence, it was not suitable for action recognition in complex situations. Laptev et al. [6] raised an action recognition method based on local traces of interest points, and combined the local feature detection method of Harris3D detection of space–time interest points with KLT tracker to obtain the movement trajectory of interest points. Wang et al. [7] presented an action recognition method on account of dense trajectory, in which a number of feature points were densely sampled in each frame and the optical flow field was applied to track. Subsequently, IDT algorithm [8, 9] was put forward to extract the video intensive tracking trajectory and calculate the trajectory features, HOG/HOF features and MBH features. Then Fisher Vector to code the features was used, and the Support Vector Machine (SVM) based on the coded feature vectors to realize the recognition of human actions was trained. With the development of convolutional neural network technology, Ji et al. [10] came up with the convolutional neural network in 2D images as an extension and convolved multiple frames of local space–time by constructing 3-D convolution. In the aspect of video-based action recognition, methods based on Two-Stream [11], C3D [12], and combined attention mechanism of Long Short-Term Memory (LSTM) [13], have been successively raised.

However, on the one hand, the non-skeleton-based action recognition methods are extremely affected by the background, illumination, and other unrelated factors. In addition, the performance of human behaviour varies greatly from various perspectives, results in low recognition accuracy. On the other hand, image data contain a large amount of data, great challenges present in computational complexity and sequence feature construction.

Action recognition based on skeleton features

Vemulapalli et al. [14] proposed the human skeleton as a set of SE(3) points, where each point represented the relative geometric transformation between a pair of rigid bodies, and the skeleton sequence was expressed as a curve in the Lie group, $SE(3) \times SE(3) \times \cdots \times SE(3)$ was a curved manifold that maps the curve to Lie algebra. They utilized Dynamic Time Warping (DTW) to align each sequence with the reference curve, and then removed the high-frequency coefficients through the Fourier space–time pyramid, SVM was used for classification. Anirudh et al. [15] raised skeleton motion trajectory features on account of the position information of the human skeleton joint points at different times, and then structured the feature space on a Riemannian manifold. Finally, they measured the trajectory similarity in the manifold space. Ding et al. [16] came up with a Spatio-Temporal Feature Chain (STFC), which illustrated a spherical coordinate system to express the motion trajectory using the direction of motion and the curvature of the trajectory, and eliminated the periodic sequence of the constructed subgraph, so as to solve the influence of noise and periodic sequence. Saeed et al. [17] generated motion templates for joint trajectories, used DTW to warp the samples and classified the extracted wavelet features using Random Forest (RF). Liu et al. [18] built a spatio-temporal LSTM model, utilized the traversal method of tree structure to extract the link features of human skeleton, and then extended the traditional LSTM analysis in time domain to time and space domain. Alaa Barkoky et al. [19] proposed a Complex Network-based feature extraction from RGB-D information. To better leverage the data in the non-Euclidean space, some works processed data directly on the graph structure, and applied the graph convolutional network (GCN) to model the skeleton-based action recognition [20, 21]. In general, the dynamic 3-D skeleton node data can well represent the human action information. And in the process of modelling, the action recognition model based on 3-D skeleton construction has a certain robustness, is not affected by the texture, background, and other factors unrelated to behaviour, and it can achieve an excellent recognition effect.

Selection method for skeleton keyframes

Studies illustrated that all the frames of the sequence were not all about behaviour identify meaningful [22]. The extraction of some representative frames from the video frame sequence for calculation, namely keyframes, can not only reduce data redundancy and computational complexity, but also express behavioural features more accurately and significantly improve the accuracy of recognition. The effectiveness, the timing and the number of keyframes are the critical factors that affect the accuracy of action recognition. Effectiveness means that the selected frames can reflect the action sequences, which makes this action sequence differ from other varieties behaviours to greatest extent. Timing refers to the order in which the keyframes occur. Too much or too little number of keyframes is not conducive to the expression of behaviour. Too few keyframes cannot reflect the content of action sequence and too many keyframes will cause information redundancy, which will affect the accuracy of action recognition.

Currently, uniform sampling method, cluster-based selection method and optimization-based selection method are used for keyframes selection. The simplest method is uniform sampling, but for slow motion and strenuous motion, it is easy to produce oversampling and under-sampling. In the clustering method, Lillo et al. [4] divided the body into four spatial regions: right arm, right leg, left arm and left leg. Each body region was represented by 21-dimensional feature vectors, which were clustered by K-Means algorithm to get a dictionary of body posture. Miranda et al. [23] put forward the characteristics of direction change, which consisted of the spherical coordinates of the joints of the head, the elbows and the knees relative to the torso, and each pose was described in terms of the angle of rotation of the connected joints. Then, multi-class SVM to detect key poses was used, and utilized Random Forest (RF) to recognize human action from key pose sequences. Enea et al. [24] proposed standardized relative direction vectors as pose features, used K-Means algorithm to gather K poses, and obtained different recognition effects under different numbers of joint poses. The keyframes obtained by the clustering method have a strong ability to summarize the description of the movement, and the accuracy of action recognition has also been improved to a certain extent. However, in the process of clustering, the specific motion meaning of data is not taken into account. In the clustering space, frames separated by a certain distance may be clustered into the same category, which ignores the temporal sequence of motion and easily leads to distortion of action analysis.

The optimization-based keyframes selection method requires the definition of a suitable objective function for optimal search. Zhang et al. [25] proposed a keyframes extraction algorithm based on multiple population genetic algorithms. They defined a fitness function to give different weights to the reconstruction error and compression rate, and used interpolation to select the keyframes. Liu et al. [26] combined the simplex method and genetic algorithm to extract keyframes from the captured human motion data, and regarded the keyframes interpolation reconstruction error and compression rate as the optimization goals. Yang et al. [27] raised a keyframes extraction method on account of quantum particle swarm optimization algorithm to solve the fast search problem in the process of keyframes reconstruction. This type of method has remarkable results in reconstructing motion sequences, but the calculation of reconstruction errors is complicated, time-consuming and requires specifying the reconstruction error parameter threshold. The selected number of frames is more suitable for motion reconstruction in the field of animation whilst is not suitable for action recognition.

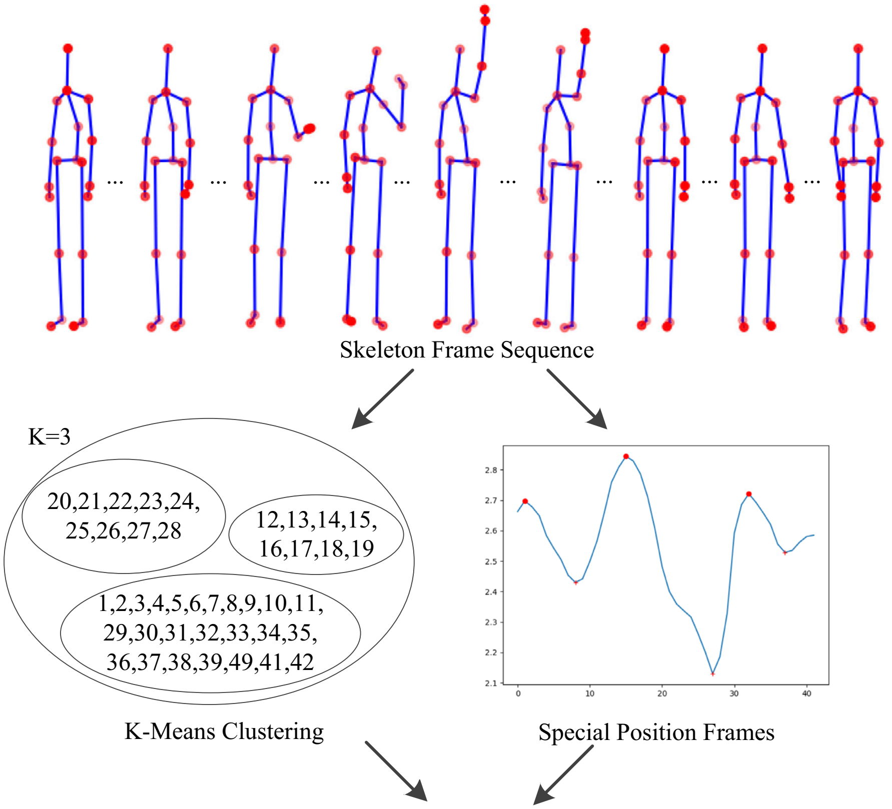
In this work, the inflection point frame is proposed to be the population initialization marker and redefine the population initialization rules, as well as it is used as the background knowledge for generating random vectors of variation to improve the global search capability of the algorithm. An evaluation model based on fused domain information and the number of key frames is proposed. In order to maintain the temporality of inter-frame information evaluation, the behaviour sequence is divided into multiple domains based on keyframes, and each keyframe is used to represent the frame sequence in the current domain, and the number of keyframes can be adjusted adaptively according to the compression rate whilst fully preserving the temporality of the motion.

Proposed algorithm

Motivation and method

Figure 1 is an example of keyframes selection process for a “high throw” action with 42 frames. A group of ID (from 1 to 42) is used to index them. Then, a K-Means method is used to group these 42 frames into K subsets. As Fig. 1 shown, the ID of 42 frames are divided into 3 subgroup when $K = 3$. After that, the frame, who is closest to the cluster centre in one subset, is selected as the keyframe. Then a sequence of keyframes can be indicated by an integer string, such as $\{1, 12, 20\}$, which will be used as the parameter to input a classifier to get the recognition result. To seek a valid value of K, the K is set to $\{3, 6, 9, 12\}$ respectively, and the obtained keyframes sequences are input into a unified SVM classifier

Fig. 1 An example of a “high throw” action



Method		Keyframe Sequence	Accuracy /%
Clustering Method	K=3	1, 12, 20	53.5
	K=6	1, 8, 11, 14, 20, 26	78.4
	K=9	1, 5, 8, 11, 14, 20, 22, 26, 29	80.3
	K=12	1, 5, 8, 11, 12, 14, 17, 19, 20, 22, 26, 29	77.2
Special Position Frames Method		2, 9, 16, 28, 33, 38	86.6

for action recognition, and the recognition accuracy rates are 53.5%, 78.4%, 80.3% and 77.2%. Obviously, amongst them, the optimal value of K is 9. In the process of extracting keyframes by K-Means, the valid K value needs to be specified in advance since different K will have a direct impact on the keyframes selection result. However, the determination of K value often requires many attempts, which is not easy to be handled automatically. In addition, the clustering method is easy to classify the frames separated by a period of time into one category. So, whether the obtained keyframe is the centroid or a frame in the clustering set, it is difficult to retain the time sequence relationship of the original frame sequence.

In above example, a new phenomenon is noticed by visualizing the trajectory of part joint points of the body, which is some extreme values of the trajectory of action amplitude will appear at part of points. The trajectories of some joint points can be mapped to a relationship between the joint momentum and the frame sequence to obtain the location of these local extreme points. As Fig. 1 shown, following above calculation procedure, the position of those inflection point of trajectories can be gotten at 6 frames, whose ID are {2, 9, 16, 28, 33, 38}. And, the accuracy of action recognition based on these 6 frames reaches 86.6%, which is significantly higher than the K-Means method. This example inspires us that these special frames, which are named as inflection point frames, have a certain discriminative value for the keyframes selection.

On the other hand, the essence of keyframes selection is an optimization problem. A binary string can be used to express a skeletal frame sequence, which means the keyframes selection problem can be transformed into a binary coded optimization problem to solve. From the above, we believe that the special inflection point frames in the skeleton frame sequence can be utilized as the identification of certain specific gene positions in the binary string, and it is helpful for us to design more effective encoding space initialization and search mechanism.

According to the above considerations, this work proposes to design a multi-population based on multi-objective Differential Evolution Algorithm (MMDE) to solve the task of keyframes optimization. As shown in Fig. 2, in this method, a binary string is used to encode the skeleton frame sequence. Meanwhile, in the population initialization stage, the inflection point frames are used as the identifications of frame segments. Then, the sequence is divided into multiple domains and the quality of keyframes is evaluated with the domain information. The quality of intra-domain and inter-domain keyframes is evaluated by weighting the characteristics of different locations with the domain information and the number of keyframes as the objective function. Combined with the compression rate of keyframes, different numbers of effective keyframes can be extracted adaptively for different types of action. Finally, the classification model is trained by the keyframes features to recognize the action.

Definitions

The skeleton model diagram used in this study is shown in Fig. 3, which includes 20 points in a skeleton frame. Each of the points contains the three coordinates in a 3-D space. Based on it, two key definitions are given:

Definition 1. Inflection point frames. In a consecutive sequence of skeleton-based frames, the frames have the extreme value of action amplitude of some joint points in the motion trajectory. Those frames are called the inflection point frame.

The motion trajectory of a certain joint behaviour can be represented by the position coordinate of the joint point sequence in the 3-D space, such as $S_3 = \{p_{1n}, p_{2n}, ..., p_{in}\}$, in which $p_{in} = (x_{in}, y_{in}, z_{in})$ represents the position coordinate of the n-th joint point in the i-th frame, $i \in \{1, 2, \ldots, t\}$. However, it is not easy to find the extreme points in a 3-D space. So S_3 has been mapped to a 2D space, which consists of a set of momentum change values of the joint points relative to their origin coordinates. It can be expressed as the following formula

$$S_3 \to S_2 = \{E_1, E_2, \ldots, E_i\}, \tag{1}$$

where the momentum change value $E_i = \sum_{m=0}^{m=n} \sqrt{(x_{im})^2 + (y_{im})^2 + (z_{im})^2}$, $i \in \{1, 2, \ldots, t\}$. Some local maximum or minimum can be found by traversing the sequence of momentum change values. Then, the frames with the same ID as these extreme points in the motion trajectory can be set as the inflection point frames. And, the set of inflection point frames can be store as $set = \{i | E_i is\, a\, local\, extremum\, point\}$.

Definition 2. Domain. The domain R with its keyframe is constituted by this keyframe and some adjacent frames on its left and right sides.

The role of keyframes' domain is to use a series of consecutive frames to evaluate the keyframe's value contained within it, which is beneficial to retain the timing relationship of the action and avoid the motion analysis distortion. In a sequence of skeleton frames $A = \{f_1, f_2, \ldots, f_t\}$, f_i represents the pose feature of i-th frame, $i \in \{1, 2, \ldots, t\}$. Then, a binary string C, whose length is also t, can be used to represent a specific sequence of frames. For example, $C =$ [1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0], in which "0" means the corresponding frame in skeleton frame sequence is non-keyframe. Otherwise, "1" means yes. In this example, 5 frames are chosen to be the keyframes, which are $\{f_1, f_5, f_8, f_{11}, f_{18}\}$. To facilitate the calculation, the middle position of two adjacent keyframes is used to divide their respective domain. Then the divided 5 domains of 5 keyframes in above example are $\{f_1^1, f_2^1, f_3^1\} \in R_1$, $\{f_4^2, f_5^2, f_6^2\} \in R_2$, $\{f_7^3, f_8^3, f_9^3\} \in R_3$, $\{f_{10}^4, f_{11}^4, f_{12}^4, f_{13}^4, f_{14}^4\} \in R_4$ and $\{f_{15}^5, f_{16}^5, f_{17}^5, f_{18}^5, f_{19}^5\} \in R_5$.

Algorithm

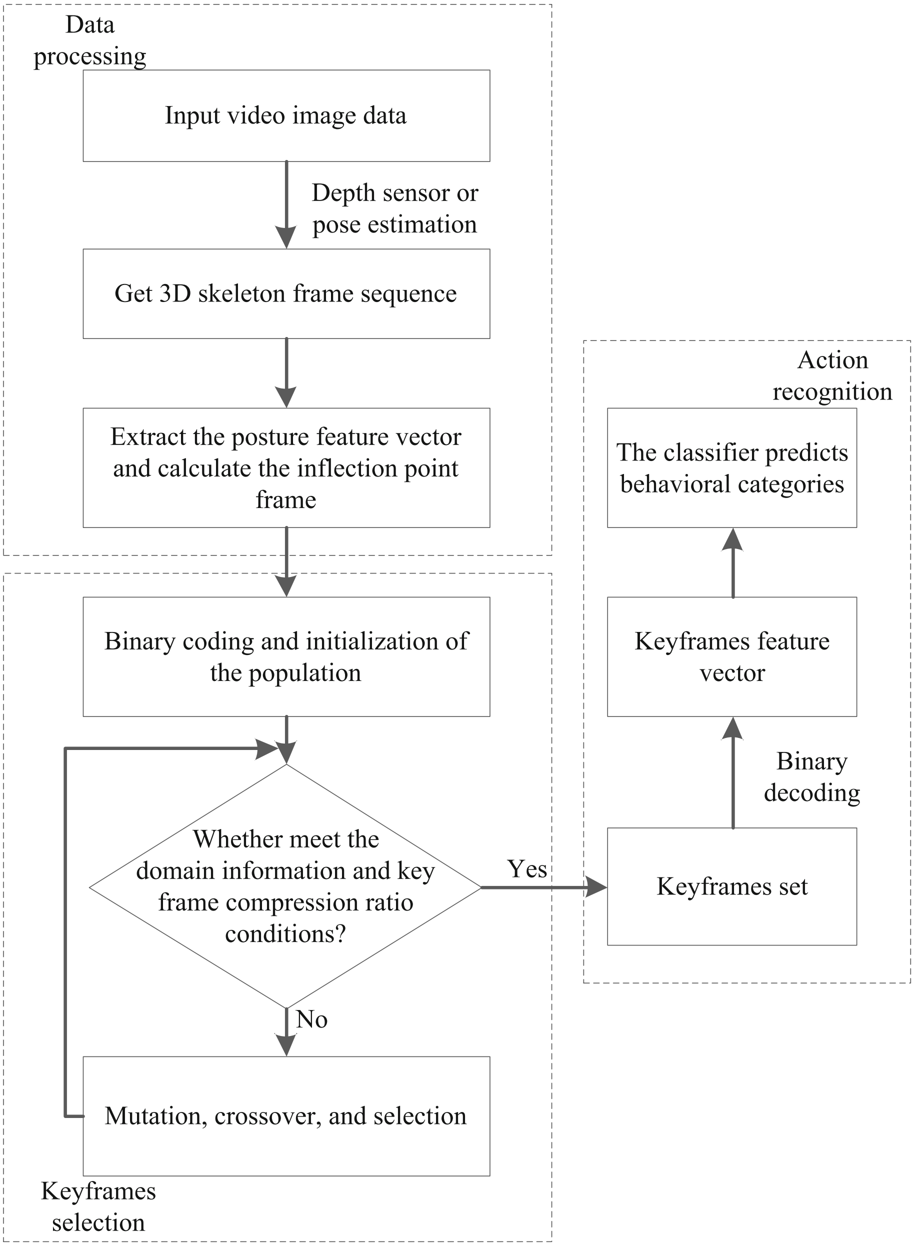
Population initialization based on inflection point frames

Evolutionary algorithm is a global optimization method based on population search. Its encoding mechanism often directly affects the design of the search mechanism [28–32]. In this paper, the 0–1 variable is used to represent the state of each frame in the sequence, and the chromosome is encoded in the form of binary string. In a behaviour sequence containing n frames, the k-th chromosome can be represented as $C_k = [a_{k1}, a_{k2}, a_{k3}, \ldots, a_{kn}]$, where a_{kn} is the state of the n-th frame. Based on the inflection point frame, 3 rules are designed to initialize the population, which are.

(1) the start frame and the end frame are set as the candidate keyframes;

(2) the inflection point frames are set as candidate keyframes;

Fig. 2 Keyframes selection based on inflection point frames


Data processing
Input video image data
Depth sensor or pose estimation
Get 3D skeleton frame sequence
Extract the posture feature vector and calculate the inflection point frame
Binary coding and initialization of the population
Whether meet the domain information and key frame compression ratio conditions?
Yes
No
Mutation, crossover, and selection
Keyframes selection
Action recognition
The classifier predicts behavioral categories
Keyframes feature vector
Binary decoding
Keyframes set

(3) to make the 3-D skeleton sequence more smooth, the adjacent two frames of a candidate keyframes on its left and right sides are set to non-keyframe. That means the corresponding initial values of the two frames in *C* are 0. And, the initial values of rest genes in *C* are generated randomly.

The population initialization on account of the inflection point frames can avoid the blindness of the initial search of the algorithm, whilst ensuring the diversity of the population, improving the quality of the initial population and accelerating the convergence speed of the algorithm. This process can be expressed as

$$X_{i,j,0} = t(j) \odot rand(0,\ 1) \tag{2}$$

where $X_{i,j,0}$ represents the *j*-th gene in the *i*-th individual of the first generation. If the frame number *j* is the start frame, the end frame is included in the inflection point frames set, then $t(j)$ is 1, otherwise it is 0, and $rand(0,\ 1)$ represents non-continuous 0 or 1 random number, $\odot$ is the logical operator *OR*.

Evaluation function of keyframes

In order to make the selected keyframes have a strong distinguishing ability for the expression of action and the number of keyframes is as small as possible, the domain information (*DI*) and the frame compression (*FC*) rate are defined as the

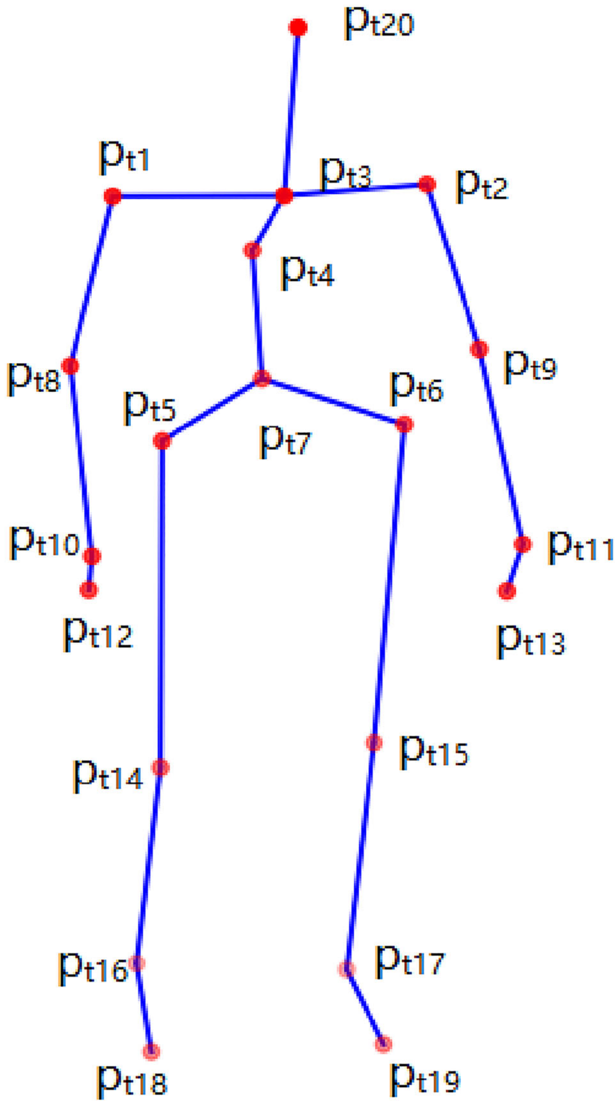


Fig. 3 Schematic diagram of skeleton model

target evaluation function, which is expressed as

$$fun = \min\{DI,\ FC\}, \tag{3}$$

Domain information (DI)

The posture features of the skeleton frame are evaluated through domain information. In the posture feature representation, many features that can represent the posture of the human body have been proposed in the existing literature. The normalized joint vector features are extracted in each frame, and the calculation formula is $d_{tn} = \frac{p_{tn}-p_{t7}}{\|p_{t3}-p_{t7}\|}$, where p_{tn} is the n-th joint point coordinates of the t-th frame, p_{t7} is the centre of the hip as the central reference point, p_{t3} is the neck joint point, and $\|\cdot\|$ represents the Euclidean distance. For each skeleton joint point, the vector d_{tn} relative to the centre reference point is calculated, and the feature vector is obtained by normalizing the distance from the neck to the centre of the abdomen.

Only the normalized joint vector feature has a weak expression of the relative direction between the joints, so the joint vector angle feature is extracted by

$$\theta_{tn} = \arccos\left(\frac{(p_{tn}-p_{t7})\cdot(p_{t3}-p_{t7})}{\|p_{tn}-p_{t7}\|\cdot\|p_{t3}-p_{t7}\|}\right). \tag{4}$$

The final pose feature of the t-th frame is composed of the normalized joint vector and the angle of the joint vector, which can be expressed as $f_t = [d_{t1}, d_{t2}, ..., d_{tn}, \theta_{t1}, \theta_{t2}, ..., \theta_{tn}]$. Compared with other feature extraction methods, the posture feature of this method is simple and intuitive to calculate, more interpretable. Besides, it can fully express the distance and direction relationship between the joint points. Furthermore, it has a strong ability to express human posture.

According to the definition of the domain, f_t is divided into different domains, and the domain information of the keyframes is obtained as

$$Intra = \sum_{r=1}^{m}\sum_{i=1}^{n} dis\left(f_i^r - f_0^r\right), \tag{5}$$

where f_i^r represents the posture feature vector of the i-th frame in the r-th domain, the keyframe vector in the r-th domain is represented as f_0^r, and $dis(\cdot)$ is used to represent the information between two frames. $dis\left(f_i^r - f_0^r\right) = w\cdot\left\|\left(f_i^r - f_0^r\right)\right\|_{2\times N}$ is the dot product of w and the Euclidean Distance of each feature between the two frames. w represents the weight of each joint feature and determined by the proportion of the movement of important joint points.

The inter-domain information of the keyframes is defined as

$$Inter = \sum_{r=1}^{m}\sum_{j=r+1}^{m} u_{rj}\left(dis\left(f_0^r - f_0^j\right)\right), \tag{6}$$

where $dis\left(f_0^r - f_0^j\right)$ represents the keyframes information in the r-th domain and the j-th domain, and u_{rj} is the weight coefficient between the two domains. The weight coefficient is related to the size of the domain interval, the keyframes information difference of adjacent domains is relatively small, and the information difference between keyframes with a large domain interval is relatively large.

The evaluation function of domain information is obtained by integrating intra-domain information and inter-domain information

$$DI = \frac{Intra}{1 + Inter} \tag{7}$$

Frame compression rate (FC)

From the perspective of keyframes extraction, considering the goal of minimizing domain information, the number of keyframes should be as small as possible to reduce data redundancy [33, 34]. Therefore, when selecting keyframes, both the domain information value and the number of extracted keyframes must be measured. Another objective function for measuring the number of keyframes is defined as frame compression ratio

$$FC = \frac{frames_{key}}{frames_{total}}, \tag{8}$$

where $frame_{key}$ is the number of selected keyframes, and $frame_{total}$ is the total number of frames included in the action sequence.

Binary-based search process

In this paper, the multi-population based on multi-objective differential evolution algorithm of keyframes selection (MMDE-KS) is proposed. To enhance the quality of solution, different population structure and search operation from the classic differential evolution algorithm are adopted. For the generated individuals, the fitness of them are calculated by formula (3). According to the dominance relations, the individuals will be sorted. Then the population will be divided into three sub-populations *Pop*1, *Pop*2 and *Pop*3 by cutting the ordered queue into three equal part. In the search process, the mutation operator is improved using logical operations instead of addition and subtraction operations. Meanwhile, the individuals involved in the operation will be selected from the three sub-populations respectively. The mutation operator is expressed as

$$V_{i,G} = X_{r1,G} \odot F \otimes \left(X_{r2,G} \oplus X_{r3,G}\right), \tag{9}$$

where $\oplus$ represents *XOR*, $\otimes$ represents *AND*, and $\odot$ represents *OR*. In the process of generating the random vector F, in order to fully retain the prior knowledge of the inflection point frames, the chromosome position of the inflection point frames are fixed at "1", and the other positions are determined by the randomly generated number and the generation probability.

An example of the mutation process is shown in Fig. 4. First, $X_{r2,G}$ and $X_{r3,G}$ are selected from the *P*op2 and *Pop*3 populations to perform the *XOR* operation. Then the *XOR* result is *AND*ed with the random vector F. As shown in the figure, the chromosome position of the inflection point frames of the random vector F are fixed to "1", which make the keyframes give priority to the inflection point frames in the search process and speeds up convergence. Other chromosome bits are generated by random numbers and generation probability to generate "0" or "1" to ensure the diversity of the population. Finally, the $X_{r1,G}$ and $F \otimes \left(X_{r2,G} \oplus X_{r3,G}\right)$ selected by *Pop*1 were calculated by *OR* to obtain $V_{i,G}$.

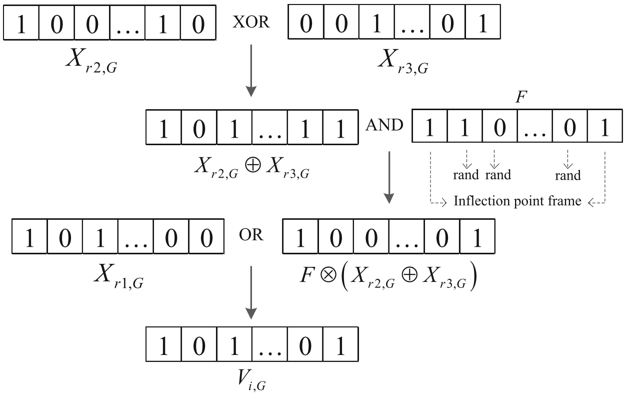


Fig. 4 Mutation process

The crossover operator can be expressed as follows:

$$U_{i,G} = \begin{cases} V_{i,G} \text{ if } rand_j \le CR \\ X_{r1,G} \text{ otherwise} \end{cases}, \tag{10}$$

where j is the chromosome bit index, $rand_j$ is a randomly generated number, $rand_j \in [0, 1]$,and crossover is performed with the probability *CR*.

When performing the selection operation, the evaluation indexes corresponding to $U_{i,G}$ and $X_{i,G}$ are calculated respectively. If $fun(U_{i,G}) \succ fun(X_{i,G})$, that is $X_{i,G}$ dominates $U_{i,G}$, then selected $X_{i,G}$ to join the candidate community. Otherwise, non-inferior solution individuals $X_{i,G}$ and $U_{i,G}$ were added into the candidate community together. Then, according to the individual fitness value, the candidate population is sorted and next population Np is generated. When the termination condition has been reached, one solution will be chosen from the Pareto optimal solutions set as the final outcome and converted to the required sequence of keyframes. Since the number of keyframes is a key factor to affect the time cost of recognition computation, the selected solution will be the one with smallest *FC*.

Experiments

Datasets and settings

The experimental environment used in this method is Intel Core i5-8400 CPU@2.80 GHz, 16 GB memory, Windows10 operating system, Python3.6. In order to verify the effectiveness of the proposed model, experiments are performed on the public datasets of MSR-Action3D, UTKinect-Action and

Florence3D-Action. The obtained keyframes binary string was decoded to obtain the keyframes sequence. In order to highlight the value and influence of the keyframes, only the basic classifier is used for classification, and combined with neural network-based classification models on large-scale NTU RGB + D datasets.

The MSR-Action3D dataset [35] is collected by Microsoft Kinect. 10 testers complete 20 actions, and each tester performs each action 2–3 times. It contains a total of 557 action sequences, and the data set provides RGB video, depth images and 3-D coordinate data marking the positions of 20 skeleton joint points. Since it contains multiple extremely similar action categories, this data set is somewhat challenging. The UTKinect-Action dataset [36] is obtained by Microsoft Kinect, which contains 10 types of actions, and 10 testers perform each action twice. Excluding the noisy sequences, there are a total of 199 effective action sequences, and each action sequence is represented by the 3-D position coordinates of 20 skeleton joint points. The Florence3D-Action dataset [37] is collected by Microsoft Kinect. It contains 10 testers performing 9 different actions, and each tester performs each action 2–3 times. It contains a total of 215 action sequences, and each action sequence is represented by the 3-D position coordinates of 15 skeleton joint points. The NTU RGB + D dataset [38] is captured by three Kinect cameras concurrently. It contains 60 action classes and 56,880 video samples. Each action sequence is represented by the 3-D position coordinates of 25 skeleton nodes.

In the experiment, according to the test standards of different individuals' cross-validation, we train half of the individual action sequence and test the other half. The parameters of the algorithm are set as follows: the population size Np is 60, the maximum number of iterations Gmax is 100, the generation probability of the vector F is 0.3, and the crossover probability CR is 0.7.

Results

To verify the effect of the keyframes extracted by the proposed method in this paper, XGBoost, RF, Softmax and Liner-SVM are used as actions classifiers to compare the original frames and the keyframes selected by the K-Means method. The results of action recognition on three datasets are shown in Table 1.

From Table 1, on the MSR-Action3D dataset, the number of clusters of K-Means is set to $K = \{3, 6, 9, 12\}$, and the recognition effect is best when $K = 9$. Set $K = \{3, 5, 7\}$ on the UTKinect-Action dataset, and the recognition effect is best when $K = 5$. Set $K = \{4, 6, 8, 10\}$ on the Florence3D-Action dataset, and the recognition effect is best when $K = 8$. It can be seen from the table, compared with the original frame, the method in this paper improves the accuracy of action recognition on different classifiers by about 10–15%. Compared with the keyframe selected by K-Means, it is increased by about 5–10%, and it performs best on the Liner-SVM classifier.

Table 2 shows the experimental statistical results of the accuracy of action recognition on the Liner-SVM classifier using the keyframes selection method extracted in this paper. The average accuracy rate on the MSR-Action3D dataset is 94.6%, the average accuracy rate on the UTKinect-Action dataset is 97.6%, and the average accuracy rate on the Florence3D-Action dataset is 94.2%. From the first quartile (25 Quartile) and the third quartile (75 Quartile) in Table 1, it can be seen that the accuracy values are mainly concentrated between 90 and 95%. At the same time, the standard deviation (Std Dev) is less than 5.1, which indicates that the keyframes selected by the method in this paper can better complete action recognition.

Comparisons

On the MSR-Action3D dataset, the effect of the proposed keyframes selection method in action recognition is compared with some typical methods proposed in recent years. The HO3DJ method used the histogram of the 3-D joint positions in the spherical coordinate system to represent the human pose characteristics, and the hidden Markov model was utilized to classify the pose vocabulary by clustering. The Eigenjoints [39] method proposed a combination of static posture, motion and offset features of the action information, and the naive Bayes nearest neighbour classifier was used for action classification. The Lie Group [14] method mapped the action curve to the Lie group curve, and utilized a combination of DTW, Fourier time pyramid and linear SVM to classify action. Actionlet [40] represented each action by learning a collection model of a subset of action joints, and took advantage of SVM to classify the actions. The ST-LSTM [18] method utilized tree structure traversal to obtain the joint point features of the human skeleton. Then the LSTM network would be constructed to establish a spatio-temporal classification model of action. The Transition Forests [41] method constructed an integrated decision tree, and extracted dynamic transition features from the nodes of the same layer to represent temporal changes. The Key Poses [42] method raised to use the centroid of the action feature cluster as the code of the action, and used the SVM classifier with radial basis as the kernel function performs action recognition.

The experimental result comparison is shown in Table 3. It can be seen from the table that the accuracy rate in the AS1 subset reaches 96.2%, and the recognition effect is better than other algorithms. The accuracy rate in the AS2 subset is 90.1%, which is slightly lower than the Transition Forest method. This method constructs an integrated

Table 1 Experimental results based on different classifiers for three datasets

Method	MSR-Action3D			UTKinect-Action			Florence3D-Action		
	Original frames/%	K-Means frames/%	Ours keyframes/%	Original frames/%	K-Means frames/%	Ours keyframes/%	Original frames/%	K-Means frames/%	Ours keyframes/%
XGBoost	69.4	80.9	86.2	78.1	89.5	93.7	72.3	82.1	88.2
RF	75.2	86.8	91.2	87.2	91.3	96.5	76.9	85.6	89.7
Softmax	74.4	85.4	88.1	85.3	91.7	95.0	78.4	84.9	90.5
Liner-SVM	76.9	87.7	**94.6**	84.8	93.4	**97.6**	83.1	88.3	**94.2**

Bold values indicate the best results of the comparison methods under the same metric

Table 2 Experimental statistical results

Dataset	Mean%	Std	25 Quartitle	75 Quartitle
MSR-Action3D	94.6	5.03	90.8	95.9
UTKinect-Action	97.6	3.09	94.0	99.0
Floence3D-Action	94.2	4.39	88.1	97.8

Table 3 Accuracy of 8 methods on the MSR-Action3D dataset

Method	AS1 accuracy/%	AS2 accuracy/%	AS3 accuracy/%	Average accuracy/%
HO3DJ	88.0	85.5	63.3	78.9
Eigenjoints	74.5	76.1	96.4	82.3
Lie Group	94.3	83.8	97.4	91.8
Actionlet	–	–	–	88.2
ST-LSTM	–	–	–	**94.8**
Transition Forests	96.1	**90.5**	97.1	94.6
Key Poses	79.5	71.9	92.3	81.2
MMDE-KS	**96.2**	90.1	**97.5**	94.6

Bold values indicate the best results of the comparison methods under the same metric

decision tree and extracts dynamic transition features from nodes in the same layer to represent temporal changes. The Transition Forest method classification model is more complicated and the decision tree usually has redundant nodes whilst our method pays more attention to the expression of keyframe pose features. In the AS3 subset, the recognition accuracy reaches 97.5%, and the recognition effect is better. The average accuracy rate reaches 94.6%, which is better than most machine learning algorithms and 2.8% higher than the Lie Group method. The Lie Group method ignores important information during uniform sampling. This is a 6.4% improvement over the Actionlet method. Actionlet method is suitable for the description of simple actions. When the actions are more complex, there are relatively many joints involved, which makes it difficult to express the actions. Although our method is slightly inferior to the ST-LSTM deep learning method, the ST-LSTM method cascades the skeleton joint point features and visual texture motion features (including HOG, HOF and convolution features). The model can still achieve a good recognition accuracy under the condition of low complexity and few features.

Table 4 Accuracy of 7 methods on the UTKinect-Action dataset

Method	Accuracy/%
HO3DJ	90.9
Template of Trajectory	96.8
Simplices	96.5
ST-LSTM	97.0
Lie Group	97.1
Key Poses	94.3
MMDE-KS	**97.6**

Bold value indicates the best results of the comparison methods under the same metric

Table 4 is the comparison result of this method and other algorithms on the UTKinect-Action dataset. The Template of Trajectory method generated motion templates from joint

Table 5 Accuracy of 6 algorithms on Florence3D-Action dataset

Method	Accuracy/%
Motion Trajectories	87.0
Lie Group	90.7
Riemannian Trajectories	89.6
Rolling Rotations	91.4
Key Poses	86.1
MMDE-KS	**94.2**

Bold value indicates the best results of the comparison methods under the same metric

trajectories. This method utilized DTW to distort the sample, and RF was utilized to classify the extracted wavelet features. In the Simplices method, the action segment was mapped to the flow space to classify the actions. It can be seen from Table 4 that our method has an accuracy rate of 97.6%, which is better than the six comparison algorithms. It is 0.6% higher than ST-LSTM, 0.5% higher than Lie Group, 0.8% higher than Template of trajectory-based method, and template-based method is not suitable for classification of complex actions. It is 3.3% higher than the Key Poses method. This method uses the centroid of the cluster as the code of the action, and the timing relationship between the action frames is ignored in the clustering process

Table 5 is the comparison result of this method and other algorithms on the Florence3D-Action dataset. The method in literature [43] structured the joint motion trajectory feature on the Riemannian flow pattern, and realized the action classification in the Riemannian space. The Rolling Rotations [44] method was an extension of the Lie Group method. It uses the relative 3-D rotation between various body parts to represent the skeleton data and maps it to the $SO_3 \times \ldots \times SO_3$ space. Compared with the Lie Group method, the effect is improved, but its model is too complicated

It can be seen from Table 5 that our method has an accuracy of 94.2% on the Florence3D-Action dataset, which is 3.5% higher than Lie Group. It is 4.6% higher than the Riemannian Trajectories method [15], and the action recognition method based on trajectory features is not conducive to the expression of complex actions.

Table 6 shows the results of combining keyframes selection with ST-GCN [45] for action recognition on the NTU RGB + D dataset. The standard test methods Cross-Subject (CS) and Cross-View (CV) are used. The original model with nine layers of spatio-temporal convolution is replaced by six layers, the first two layers are 64 channels, the middle two layers are 128 channels, the last two layers are 256 channels, the dropout parameter is 0.5, and the optimization The dropout parameter is 0.5, and the optimization method is SGD, with a learning rate of 0.01.

Table 6 Accuracy of 5 methods on NTU RGB + D dataset

Methods	CS Accuracy	CV Accuracy
ST-GCN	81.5	88.3
ST-LSTM	69.2	77.7
VA-LSTM	79.4	87.6
AS-GCN	86.8	94.2
MMDE-KS + ST-GCN	83.2	93.7

As can be seen from Table 6, the keyframes selected in this paper combined with the ST-GCN method achieved 83.2% and 93.7% recognition accuracy under the CS and CV standards of the NTU RGB + D dataset, respectively. This result outperforms the ST-GCN recognition accuracy based on the original literature, and improves by 1.7% and 5.3% under CS and CV standards. Compared with other deep learning methods, the recognition accuracy of MMDE-KS + ST-GCN is better than that of ST-LSTM and VA-LSTM [46], and slightly lower than that of AS-GCN [47] method. From the above experiments, it can be seen that MMDE-KS is able to select the keyframes that can effectively represent the behavioural motion, thus further improving the recognition accuracy of the model.

Discussion

Influence of inflection point frames identification on accuracy

In order to evaluate the effect of inflection point frames as a priori knowledge in the process of population initialization and mutation, we conduct experiments on Florence3D-Action data set in two cases of randomly initialized population and mutation vector and the initial population and mutation vector based on inflection point frames. The accuracy varies with the number of iterations, as shown in Fig. 5.

It can be seen from the figure that when the inflection point frame processing is added, the recognition accuracy rate stabilizes to 94.3% at the 80-th generation. In the case of frames without inflection points, the recognition accuracy converges slowly, and the optimal recognition effect is 91.1% around the 130-th generation. Therefore, the identification based on the inflection point frame has a better effect in the keyframes selection and the convergence speed is faster, which verifies the effectiveness of the inflection point frame.

Influence of inflection point frame identification on difference action categories

The Florence3D-Action dataset on the keyframes features selected by the inflection point frames identification is

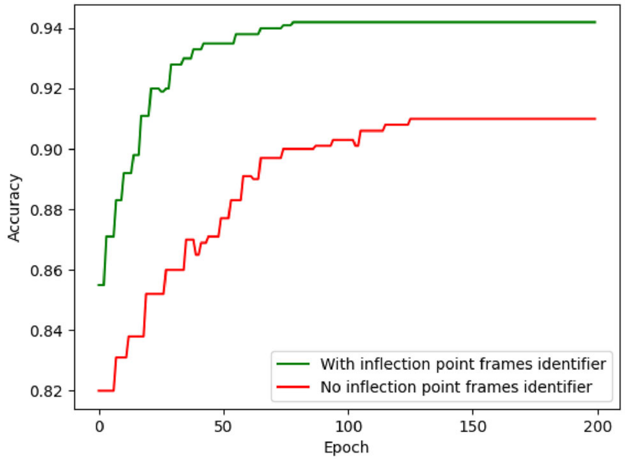


Fig. 5 Changes of accuracy during the iterations with or without inflection point frames

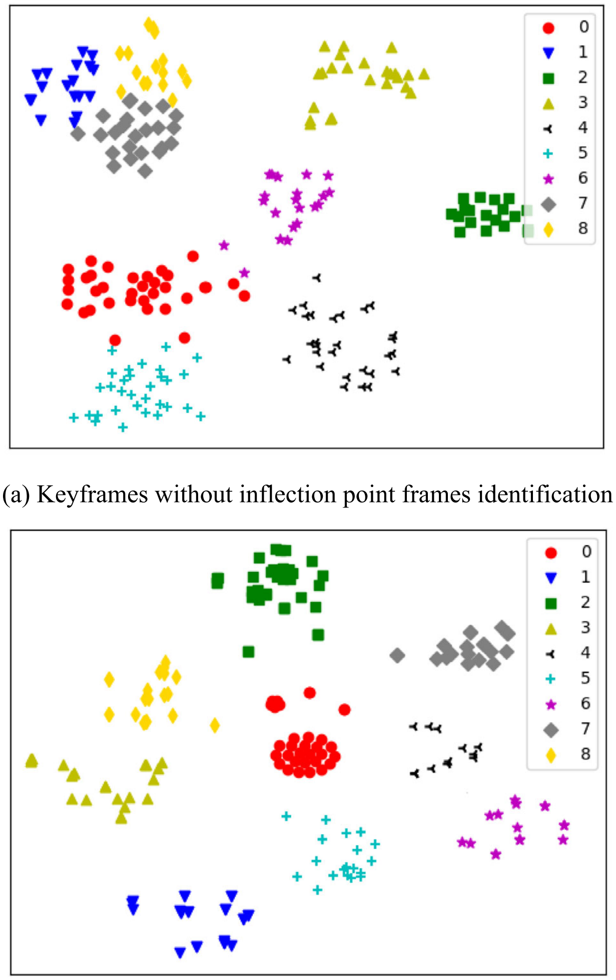
Table 7 Evaluation results of the clustering based on two methods

Method	AMI	Homogeneity	V-Measure
No inflection point frames identifier	0.654	0.678	0.681
With inflection point frames identifier	**0.696**	**0.722**	**0.727**

Bold values indicate the best results of the comparison methods under the same metric

clustered, and Adjusted Mutual Information (AMI), Homogeneity and V-Measure coefficients are used to measure the clustering category relationship with real action labels. The larger the value of the evaluation coefficient, the better the clustering effect. Table 7 shows the evaluation results of the two methods of clustering effect, and Fig. 6 shows the feature clustering results of the keyframe sequence selected with or without the inflection point frame identification.

Combined with the results in Table 7 and Fig. 6, it can be found that the clustering effect of keyframes features extracted with inflection point frames identification is more obvious. The reason is that when there is an inflection point frame mark, it is easy to pick the frame with the maximum motion range of the body, which is more expressive of the physical meaning of the movement, and the boundary between different classes is clear as well as the degree of cohesion between different classes is high. In the evaluation index, the values of AMI, Homogeneity function and V-measure are 0.6961, 0.7225 and 0.7278, which are higher than the clustering result of non-inflexion frame. This fully demonstrates and proves the function of inflection point frame identification in feature expression.



(a) Keyframes without inflection point frames identification

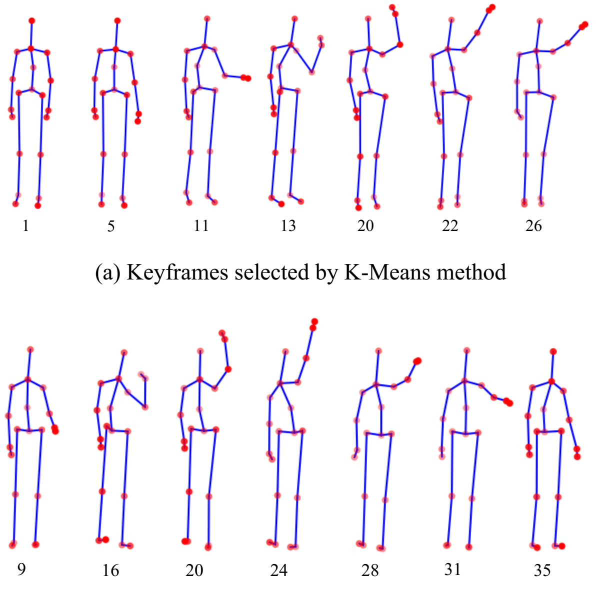
(b) Keyframes with inflection point frames identification

Fig. 6 Feature clustering results of two methods

Differences of the selection of keyframes in the division of domains

In order to evaluate the difference of the domain division on the selected keyframes, the K-Means method and the keyframes selected by our method are used to visualize the “high throw” action as shown in Fig. 7. The keyframes selected by K-Means are {1, 5, 11, 13, 20, 22, 26}, and the keyframes selected by our method are {9, 16, 20, 24, 28, 31, 35}. By comparing the keyframes selected by the two methods on the whole, the keyframes in the K-Means method are mainly concentrated in the first half cycle of the whole sequence (42 frames in total), which cannot fully express the whole action sequence.

In addition, as can be seen from the Fig. 8, amongst the keyframes selected by the K-Means method, the correlation of adjacent frames in features is much greater than our method, which results in data redundancy. The keyframes

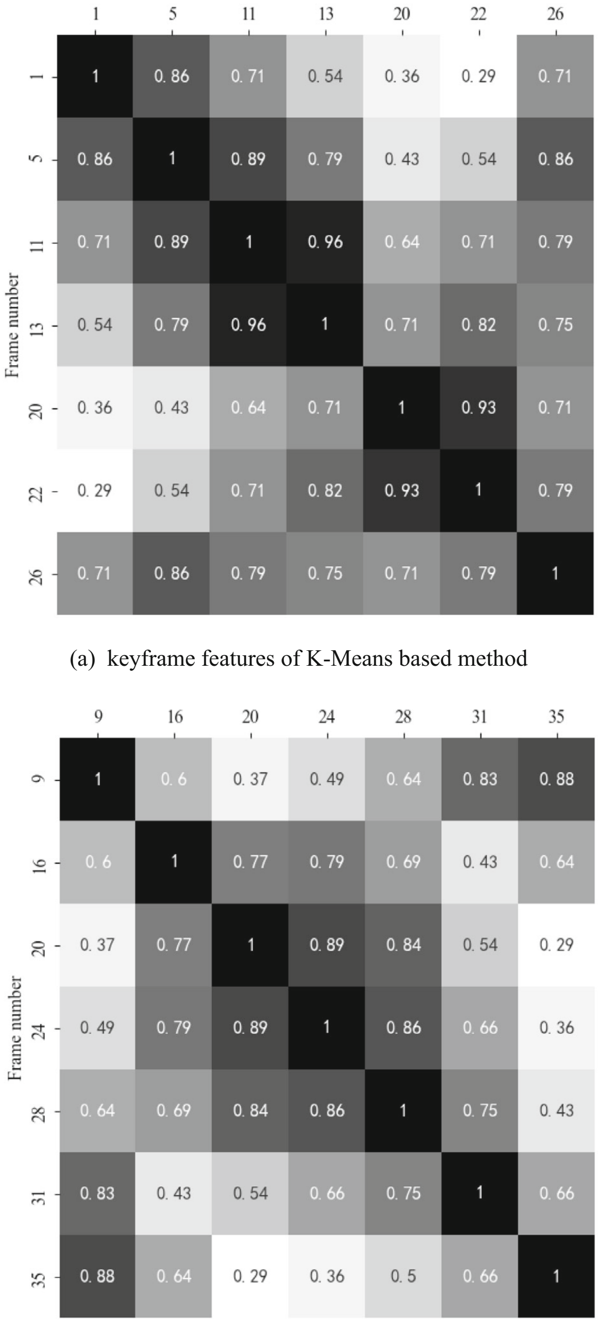


(a) Keyframes selected by K-Means method

(b) Keyframes selected by our method

Fig. 7 Visual comparison between two keyframes selection methods

selected by our method show strong generality and are more expressive of the physical meaning of action. The skeleton frame sequence is divided into multiple domains, and the keyframes in each domain are evaluated by domain information to fully preserve the timing characteristics of action. The domain information synthesizes the intra-domain information and inter-domain information to make the intra-domain information as small as possible. So, the compactness between the selected keyframes and the current domain is better. Meanwhile, the information between domains is as large as possible. That is the similarity of the selected keyframes between each domain is small, and the set of keyframes can describe the action sequence well.



	1	5	11	13	20	22	26
1	1	0. 86	0. 71	0. 54	0. 36	0. 29	0. 71
5	0. 86	1	0. 89	0. 79	0. 43	0. 54	0. 86
11	0. 71	0. 89	1	0. 96	0. 64	0. 71	0. 79
13	0. 54	0. 79	0. 96	1	0. 71	0. 82	0. 75
20	0. 36	0. 43	0. 64	0. 71	1	0. 93	0. 71
22	0. 29	0. 54	0. 71	0. 82	0. 93	1	0. 79
26	0. 71	0. 86	0. 79	0. 75	0. 71	0. 79	1

(a) keyframe features of K-Means based method

	9	16	20	24	28	31	35
9	1	0. 6	0. 37	0. 49	0. 64	0. 83	0. 88
16	0. 6	1	0. 77	0. 79	0. 69	0. 43	0. 64
20	0. 37	0. 77	1	0. 89	0. 84	0. 54	0. 29
24	0. 49	0. 79	0. 89	1	0. 86	0. 66	0. 36
28	0. 64	0. 69	0. 84	0. 86	1	0. 75	0. 43
31	0. 83	0. 43	0. 54	0. 66	0. 75	1	0. 66
35	0. 88	0. 64	0. 29	0. 36	0. 5	0. 66	1

(b) Keyframe features of our method

Fig. 8 Keyframe features matrixes

Conclusion

In this paper, the 3-D skeleton keyframes selection method based on inflection point frames identification and the binary code DE optimization mechanism for the human action recognition are proposed. First, the definition of the inflection point frames is proposed, which are calculated using the joint motion momentum of the human body. Second, the inflection point frames are utilized as a key mark to convert the keyframes selection problem into a multi-objective optimization problem under the binary code space to solve. In order to preserve the time sequence information of the action, the frame sequence is divided into multiple domains to evaluate, including the domain information (DI) and keyframes compression rate (FC). The proposed method can adaptively select keyframes according to the complexity of the action. Moreover, a multi-objective differential evolution algorithm based on multiple search groups is designed, which divides the population by sorting individuals. Based on it, the individuals involved in mutation operations are selected from different groups to improve the global search ability and convergence speed of the binary coded population. Finally, the posture features based on the selected keyframes decoding are input to the SVM classifier to obtain the action classification results. The experiments are conducted on 3 public datasets, MSR-Action3D, UTKinect-Action and Florence3D-Action. The results show that the accuracy rates on the three datasets reached 94.6%, 97.6% and 94.2%. Besides, combined keyframes and deep learning classifier on the NTU RGB + D dataset, the accuracy can reach 83.2% and 93.7%. By comparing with multiple state-of-the-art methods, the effectiveness of the proposed method is verified in this paper. In addition, the effect of inflection points frames and the significance of dividing domains in frame sequence are discussed. In the future works, we will try to improve the multi-objective differential evolution algorithm to deal with the high-dimensional optimization problem that needs to be solved when processing long videos.

Acknowledgements This work was supported by the National Nature Science Foundation of China (61876138 and 61203311), the Key Research and Development Program of Shaanxi Province of China (2020SF-375 and 2022GY-315) and Xi ’an Key Laboratory of Big Data and Intelligent Computing.

Data availability The data that support the findings of this study are openly available in 4 public datasets from references [35–38].

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Dang LM et al (2020) Sensor-based and vision-based human activity recognition: a comprehensive survey. Pattern Recogn 108:107561
2. Elias P, Sedmidubsky J, Zezula P (2019) Understanding the gap between 2D and 3D skeleton-based action recognition. In: 2019 IEEE International Symposium on Multimedia (ISM), pp 192–195
3. Xuan TN, Ngo TD, Le TH (2019) A Spatial-temporal 3D human pose reconstruction framework. J Inform Process Syst 15(2):399–409
4. Lillo I, Soto A, Niebles JC (2014) Discriminative hierarchical modeling of spatio-temporally composable human activities. In: Proceedings of the IEEE International Conference on computer vision, p 812–819
5. Bobick AF, Davis JW (2001) The recognition of human movement using temporal templates. IEEE Trans Pattern Anal Mach Intell 23(3):257–267
6. Laptev I (2005) On space-time interest points. Int J Comput Vis 64(2–3):107–123
7. Wang H, Klaser A, Schmid C, et al (2011). action recognition by dense trajectories. In: Proceedings of the IEEE International Conference on computer vision, pp 3169–3176
8. Wang H et al (2013) Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vision 103(1):60–79
9. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on computer vision, pp 3551–3558.
10. Ji S et al (2012) 3D convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
11. Feichtenhofer C, Pinz A, Zisserman A (2016). Convolutional two-stream network fusion for video action recognition.In: Proceedings of the IEEE International Conference on computer vision, pp 1933–1941
12. Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri (2015) In: Proceedings of the IEEE International Conference on Computer Vision, 4489–4497.
13. Sudhakaran S, Escalera S, Lanz O (2019), LSTA: Long short-term attention for egocentric action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 9954–9963.
14. Vemulapalli R, Arrate F, Chellappa R (2014) Human action recognition by representing 3Dskeletons as points in a lie group. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 588–595
15. Anirudh R et al (2016) Elastic functional coding of Riemannian trajectories. IEEE Trans Pattern Anal Mach Intell 39(5):922–936
16. Ding W et al (2015) STFC: Spatio-temporal feature chain for skeleton-based human action recognition. J Vis Commun Image Represent 26:329–337
17. Ghodsi S, Mohammadzade H, Korki E (2018) Simultaneous joint and object trajectory templates for human activity recognition from 3-D data. J Vis Commun Image Represent 55:729–741
18. Liu J et al (2017) Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. IEEE Trans Pattern Anal Mach Intell 40(12):3007–3021
19. Barkoky A, Charkari NM (2022) Complex Network-based features extraction in RGB-D human action recognition. J Vis Commun Image Represent 82:103371
20. Liu Y, Zhang H, Xu D et al (2022) Graph transformer network with temporal kernel attention for skeleton-based action recognition. Knowl-Based Syst 240:108146
21. Zhang J, Ye G, Tu Z et al (2022) A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition. CAAI Trans Intell Technol 7(1):46–55
22. Schindler K, Van Gool L (2008) Action snippets: how many frames does human action recognition require. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 1–8
23. Miranda L et al (2014) Online gesture recognition from pose kernel learning and decision forests. Pattern Recognit Lett 39:65–73
24. Enea C, Samuele G, Ennio G et al (2016) A human activity recognition system using skeleton data from RGBD sensors. Comput Intell Neurosci 2016. https://doi.org/10.1155/2016/4351435
25. Qiang Z, Zhang S, Zhou D (2014) Keyframe extraction from human motion capture data based on a multiple population genetic algorithm. Symmetry 6(4):926–937
26. Liu X-M, Hao A-M, Zhao D (2013) Optimization-based key frame extraction for motion capture animation. Vis Comput 29(1):85–95

27. Yang T, Sun HJ, Jun YE (2014) Extraction of keyframe from motion capture data based on quantum-behaved particle swarm optimization. Appl Res Comput 2(205):526–530
28. Kumar PS (2020) Algorithms for solving the optimization problems using fuzzy and intuitionistic fuzzy set. Int J Syst Assur Eng Manag 11:189–222. https://doi.org/10.1007/s13198-019-00941-3
29. Kumar PS (2023) The PSK method: a new and efficient approach to solving fuzzy transportation problems. In: Boukachour J, Benaini A (eds) Transport and logistics planning and optimization. IGI Global, pp 149–197. https://doi.org/10.4018/978-1-6684-8474-6.ch007
30. Kumar PS (2020) Developing a new approach to solve solid assignment problems under intuitionistic fuzzy environment. Int J Fuzzy Syst Appl (IJFSA) 9(1):1–34. https://doi.org/10.4018/IJFSA.2020010101
31. Kumar PS (2023) The theory and applications of the software-based PSK method for solving intuitionistic fuzzy solid transportation problems. In: Habib M (ed) Perspectives and considerations on the evolution of smart systems. IGI Global, pp 137–186. https://doi.org/10.4018/978-1-6684-7684-0.ch007
32. Kumar PS (2019) Intuitionistic fuzzy solid assignment problems: a software-based approach. Int J Syst Assur Eng Manag 10:661–675. https://doi.org/10.1007/s13198-019-00794-w
33. Aziz RM (2022) Application of nature inspired soft computing techniques for gene selection: a novel frame work for classification of cancer. Soft Comput 26:12179–12196
34. Aziz RM (2022) Nature-inspired metaheuristics model for gene selection and classification of biomedical microarray data. Med Biol Eng Compu 60(6):1627–1646
35. Li W, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3D points. In: IEEE International Workshop on CVPR for Human Communicative Behavior Analysis (in conjunction with CVPR2010), San Francisco, CA, June, 2010
36. Xia L, Chen CC, Aggarwal JK (2012) View invariant human action recognition using histograms of 3D joints. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pp 20–27
37. Seidenari L, Varano V, et al (2013), 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, pp 479–485
38. Shahroudy A, Liu J, Ng TT, et al (2016), NTU RGB+D: a large scale dataset for 3d human activity analysis. In: IEEE Computer Society, pp 1010–1019
39. Yang X, Tian YL (2012) Eigenjoints-based action recognition using Naive-Bayes-Nearest-Neighbor. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 14–19
40. Wang J, Liu Z, Wu Y, et al (2012) Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 1290–1297
41. Garcia-Hernando G, Kim TK (2017) Transition forests: Learning dis-criminative temporal transitions for action recognition and detection. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 432–440
42. Wang C, Flynn J, Wang Y, et al (2016) Recognizing actions in 3Dusing action-snippets and activated simplices. In: Proceedings of the 30th AAAI Conference on artificial intelligence, pp 3604–3610
43. Devanne M et al (2014) 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. IEEE Trans Cybern 45(7):1340–1352
44. Vemulapalli R, Chellappa R (2016) Rolling Rotations for Recognizing Human Actions from 3D Skeletal Data. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 4471–4479
45. Yan S, Xiong Y, Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI Conference on artificial intelligence, 32(1): 7444–7452
46. Zhang P, Lan C, Xing J, et al (2017) View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 2117–2126
47. Li M, Chen S, Chen X, et al (2019) Actional-structural graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 3595–3603