A Framework for Human Motion Segmentation Based on Multiple Information of Motion Data

Xiaofei Zan[1], Weibin Liu[1,*] and Weiwei Xing[2]

[1]Institute of Information Science, Beijing Jiaotong University

Beijing, China

[e-mail: wbliu@bjtu.edu.cn]

[2]School of Software Engineering, Beijing Jiaotong University

Beijing, China

[*]Corresponding author: Weibin Liu

Received January 25, 2019; revised March 12, 2019; accepted April 7, 2019; published September 30, 2019

Abstract

With the development of films, games and animation industry, analysis and reuse of human motion capture data become more and more important. Human motion segmentation, which divides a long motion sequence into different types of fragments, is a key part of mocap-based techniques. However, most of the segmentation methods only take into account low-level physical information (motion characteristics) or high-level data information (statistical characteristics) of motion data. They cannot use the data information fully. In this paper, we propose an unsupervised framework using both low-level physical information and high-level data information of human motion data to solve the human segmentation problem. First, we introduce the algorithm of CFSFDP and optimize it to carry out initial segmentation and obtain a good result quickly. Second, we use the ACA method to perform optimized segmentation for improving the result of segmentation. The experiments demonstrate that our framework has an excellent performance.

Keywords: Motion capture, Human motion, Motion segmentation, Density peak clustering, Time-Series clustering

http://doi.org/10.3837/tiis.2019.09.017 **ISSN : 1976-7277**

1. Introduction

Human motion capture technology is an effective way to get motion data. It records the human body's motion trajectory in three-dimensional space through sensors and transforms it into abstract motion data. This technology can be used for motion simulation or driving virtual human. Also, it is a crucial technology of movies, animation, computer games and other fields. Because the motion capture devices are costly and the operation of capture is quite cumbersome, we need to split long sequences into short sequences. This processing will make motion data storage, extraction, modification and synthesis easier. As a basis for analysis and reuse of human motion capture data, motion segmentation divides long sequences with different types of motion to get segments with independent semantics (as shown in **Fig. 1**). In recent years, how to segment motion capture data including various sports behaviors efficiently and accurately has become a hot topic in the field of computer animation.

Traditional methods only use low-level physical information (motion characteristics) or high-level data information (statistical characteristics) of motion data for motion segmentation, and they cannot utilize information of motion data fully. For using multiple information of motion data to get better human motion segmentation results, this paper proposes a framework combining the optimization algorithm of clustering by fast search and find of density peaks (CFSFDP [1]) and aligned cluster analysis (ACA [2]).

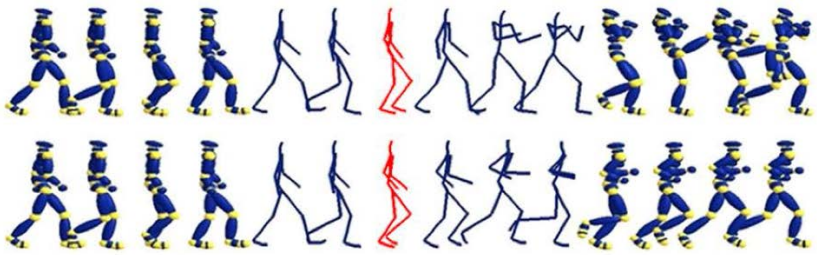

Fig. 1. Dividing long sequences into segments with independent semantics.

The main contributions of this paper could be summarized as follows.

1. We propose a framework using multiple information of motion data which combine low-level physical information and high-level data information for finishing human motion segmentation.
2. We build a combined feature of center distance and velocity to represent human motion.
3. We introduce the algorithm of CFSFDP and optimize it to solve the human motion segmentation problem.

The rest of paper is organized as follows.

Section 2 reviews previous research works on human motion segmentation. Section 3

overviews the framework of this paper. Section 4 introduces data extraction and timing reducing. Section 5 presents how to measure the distance between two motion frames. Section 6 describes how to perform initial segmentation in detail. And then section 7 details how to perform optimized segmentation of the initial segmentation result. Section 8 introduces something about experiments. Section 9 concludes our paper and draws future research directions.

2. Related Work

Segmentation of human motion data is closely related to data mining, machine learning, timing clustering and behavioral recognition. Many different motion segmentation methods can segment long sequences into different short segments. These methods are mainly divided into two types.

A. Methods based on motion characteristics

Many methods [3,4,5] first extract some physical features of motion data, such as velocity, acceleration, and joint angles. After that, they extract a segmentation point of the motion sequence according to the distribution changes of these features such as local extremum. These methods are simple and easy to implement. However, only using the low-level physical information makes the applicability of these methods poor and segmentation results lacking semantic meaning.

B. Methods based on statistical characteristics

These methods include machine learning methods and non-machine learning methods.

Machine learning methods: Barbic et al. [6] use Gaussian mixture model (GMM) solving the motion segmentation problem. The GMM-based method assumes that different motion segments come from different Gaussian distributions. In this way, the problem of motion segmentation translates into finding the optimal Gaussian distribution for the motion sequences under the condition of timing maintaining. This method is very straightforward and efficient, but it only suits for the segmentation of short motion sequences. Based on their previous work ACA [2], Zhou et al. [7] proposed a method of hierarchical aligned cluster analysis (HACA) for human motion segmentation. This method uses a non-supervised hierarchical bottom-up framework whose segmentation result is excellent.

Non-machine learning methods: Gong et al. [8] propose a kernelized temporal cut (KTC) method. This method searches for the frame with the smallest value of objective function through a one-dimensional linear search in a preset window to obtain a segmentation point. After that, they employ this segmentation point as a starting point to continue searching process. Devanne et al. [9,10] consider the motion sequence as a curve and use the geodesic distance to measure the similarity between two curves. Afterwards, they make use of a standard deviation to construct a standard deviation curve in a sliding window. The segmentation point could be got by finding the local minimum value of the curve. Krüger et

al. [11] use a feature bundling technique to finish preprocessing first, then exploit the self-similar structure of motion sequences to cluster human motion data. This method requires no interaction for the segmentation. Yu et al. [12] deal with the human motion segmentation problem based on graph partition method. They construct an undirected weighted graph according to motion data and apply the t-nearest neighbors and Nystrom method for clustering motion data to finish motion segmentation.

Methods described above can achieve good segmentation results under certain conditions. However, they only take advantage of low-level physical information, or high-level data information of motion data, but the data information has not been fully utilized. Based on the previous works, this paper presents a framework which is suitable for human motion segmentation. This framework makes use of both low-level physical information and high-level data information of motion data to acquire better segmentation results.

3. Framework Overview

ACA has a good performance in human motion segmentation. However, it requires to input the number of motion categories manually. Furthermore, because the last problem of this method to be solved is a non-convex optimization problem, the quality of the inputting data needs to be very good. Otherwise, it is easy to have a sharp increase in computation and the result may fall into the local optima. These affect the result significantly.

The algorithm of CFSFDP we introduced and optimized could get the number of motion categories automatically and generate a better clustering result efficiently.

Based on the advantages of the optimization algorithm of CFSFDP and disadvantages of ACA, this paper combines these two methods masterly and proposes a framework to deal with the human motion segmentation problem. **Fig. 2** illustrates the overview of this framework.

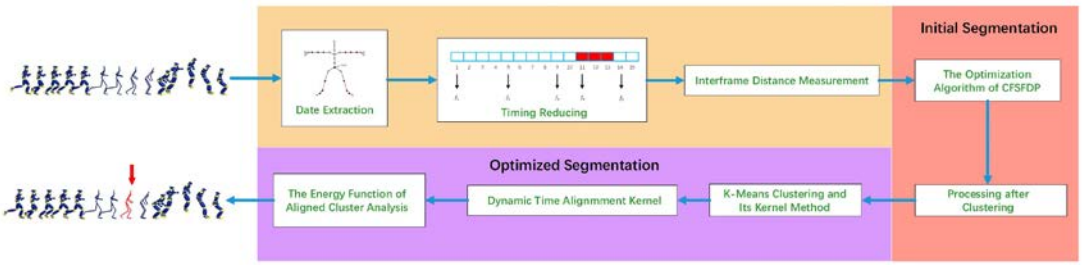

Fig. 2. The overview of this framework.

The main content of this framework has two parts: initial segmentation and optimized segmentation.

The first part is initial segmentation which only uses the motion characteristics of motion data for segmentation. After the initial segmentation is completed, the result of initial segmentation is used as the input of the optimized segmentation which basically uses statistical characteristics of the motion data for segmentation. By using these two parts, we utilize low-level physical information and high-level data information of motion data separately. The detail of the framework is shown as the following steps.

a) Data extraction. Combining the composition of the motion capture data with the characteristics of human motion, we take rotation data of key joints and convert them into the form of quaternion.

b) Timing reducing. We remove the less representative data and keep the primary information for the later processing.

c) Interframe distance measurement. We construct a combined feature of center distance and velocity to represent the motion feature and use it to calculate the distance between two motion frames.

Initial segmentation:

d) The optimization algorithm of CFSFDP. At first, we design the optimization algorithm of CFSFDP to select cluster centers, and then allocate the remaining non-center points to each cluster.

e) Processing after clustering. After the clustering is finished, we combine consecutive frames with the same class label to get the recognition matrix G according to the motion type corresponding to each motion segment and E saving the start and end frames of each motion segment.

Optimized segmentation:

f) K-means clustering and its kernel method. A nonlinear mapping is introduced to extend k-means clustering algorithm to a kernel space for forming the kernel k-means clustering algorithm.

g) Dynamic time alignment kernel. We bring dynamic time warping (DTW) method and embed it in a kernel function to measure the distance between time-series of different lengths.

h) The energy function of ACA. The kernel k-means clustering algorithm and dynamic time alignment kernel (DTAK) are combined to establish an energy function for the motion segmentation problem. After solving the problem of the minimum energy function, we are able to get the final segmentation result.

4. Data extraction and timing reducing

Human motion capture data is described by the motion data of each frame, and the motion data of each frame is reflected by the change of each joint finally. In Carnegie Mellon University Graphics Lab Motion Capture Database [13], the human skeleton has 30 joints

except the ROOT node. However, many bones, such as fingers, toes and head, are not much correlation with the change of motion. To reduce the amount of data that needs to be processed and speed up the calculation, we choose to discard the joints which are not essential and only consider the J joints that have a more significant influence on the motion. In this method, we set J as 12. More details are shown in **Fig. 3**.

The motion capture data is described by the rotation angle of each joint, that is, the Euler angle. To avoid the Gimbal Lock, massive data storage, and other issues, we convert Euler angles to quaternions.

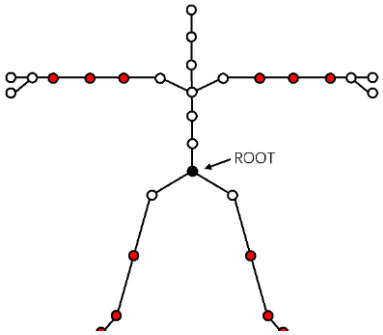

Fig. 3. The human skeleton. The red points indicate key joints.

Motion sequence usually has a long length. This has challenges to the current segmentation algorithm. We remove the less representative data and keeping the data of main information for data processing. Due to human motion is locally linear [14], we could assume that the human body moves smoothly. There we can reduce the number of frames without losing information [15].

5. Interframe distance measurement

To segment the motion sequence, we ought to cluster each frame of the original motion sequence. We consider each frame as a high-dimensional data point. Subsequently, we need to calculate the distance between every two high-dimensional data points, that is, the distance between two motion frames.

The measurement of the distance between two frames is actually the difference in the features of two motion frames. To express the feature of each frame more accurately, we define a combined feature of center distance and velocity.

The first group of features is the center distance feature [16].

As shown in **Table 1**, the distances from the ROOT node to the 12 key joints of upper and lower limbs are respectively extracted. Given frame i, the eigenvector d_i of center distance is represented as $d_i = \left(d_i^1, d_i^2, \ldots, d_i^{12}\right)$.

The second group of features is the velocity.

We define a pose $x = (x_{root}, r_1, r_2, \dots, r_{12})$ of each frame consisting of the root position vector x_{root} and the rotation quaternions $r_1, r_2, \dots, r_{12}$ of 12 key joints. The velocity $v = (v_{root}, q_1, q_2, \dots, q_{12})$ is composed of the root displacement vector v_{root} and the displacement quaternions $q_1, q_2, \dots, q_{12}$ of 12 key joints. After that, we could calculate these variables through finite difference. Given two poses x and x', we are capable to calculate the finite difference according to the following equation.

$$v = x' \ominus x = (x'_{root} - x_{root}, r'_1 r_1^{-1}, r'_2 r_2^{-1}, \dots, r'_{12} r_{12}^{-1}) \quad (1)$$

Afterwards, we are able to get the combined feature of each frame by synthesizing d and v. Finally, we subtract the combined feature of two frames and the result is the interframe distance we need.

Table 1. Calculation of the center distance feature

Upper Limbs	Lower Limbs
d_i^1: ROOT→lhumerus	d_i^7: ROOT→lfemur
d_i^2: ROOT→lradius	d_i^8: ROOT→ltibia
d_i^3: ROOT→lwrist	d_i^9: ROOT→lfoot
d_i^4: ROOT→rhumerus	d_i^{10}: ROOT→rfemur
d_i^5: ROOT→rradius	d_i^{11}: ROOT→rtibia
d_i^6: ROOT→rwrist	d_i^{12}: ROOT→rfoot

6. Initial Segmentation Based on the Optimization Algorithm of CFSFDP

This part constructs an optimization algorithm of CFSFDP to process human motion segmentation problem. Using the optimization algorithm of CFSFDP can make the segmentation unsupervised. Also, in this way it can provide a good start for the later optimized segmentation and improve the overall performance of the framework.

When we employ this clustering method to segment human motion data, we mainly take advantage of the motion characteristics of data. The motion characteristics here contain information of the human bone structure and the speed of motion. Therefore, this part is utilizing the low-level physical information of the motion data to complete segmentation.

6.1 The optimization algorithm of CFSFDP

We introduce and optimize the algorithm of CFSFDP to process human motion segmentation problem. Here, we optimize the definition of local density and construct the cluster center selection algorithm to determine the number of clusters automatically.

6.1.1 The key parts of the algorithm

The key parts of the algorithm are drawing the decision graph and selecting cluster centers. These two parts and corresponding optimization work are shown below.

6.1.1.1 Drawing the decision graph

Using the k-nearest neighbor information of sample i to define the local density ρ_i can better reflect the actual distribution information of data samples [17].

Definition 1 The local density. We define the local density ρ_i of sample i as Eq. (2), where $KNN(i)$ is a set of k-nearest neighbor samples of sample i and d_{ij} is the distance between sample i and j.

$$\rho_i = \sum_{j \in KNN(i)} \exp(-d_{ij}) \tag{2}$$

According to Eq. (2), we know the local density of a sample is only related to its k-nearest neighbor samples, which can better reflect the local information of a sample point.

Definition 2 The local distance. We define the local distance δ_i of sample i as follows: If $\rho_i < \rho_j$:

$$\delta_i = min_j d_{ij} \tag{3}$$

If ρ_i is maximum local density:

$$\delta_i = max_j d_{ij} \tag{4}$$

So, we are able to calculate the local density ρ_i and the local distance δ_i of sample i ($i = 1, \ldots, N$) according to Eqs. (2), (3) and (4), respectively. After normalizing δ_i and ρ_i, we could construct a decision graph of δ_i with respect to ρ_i (as shown in **Fig. 4**). According to the decision graph, we could select the cluster centers.

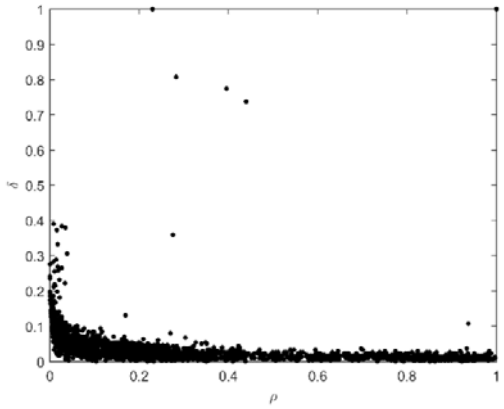

Fig. 4. Decision graph.

6.1.1.2 Selecting cluster centers

To achieve cluster centers, we introduce the concept of cluster center weight according to the decision graph.

Definition 3 Cluster center weight. We define the cluster center weight w_i of sample i as Eq. (5).

$$w_i = \delta_i \cdot \rho_i \tag{5}$$

To find the set of points with the biggest degree of deviation, that is, the points at the top right of decision graph, we rank the cluster center weight in a descending order and obtain the first m points. From **Fig. 5** we are able to get that the cluster center weight generally shows a decline trend, but the degree of decline is changeable. There is an important point in the graph, at which point the overall declining trend slows down.

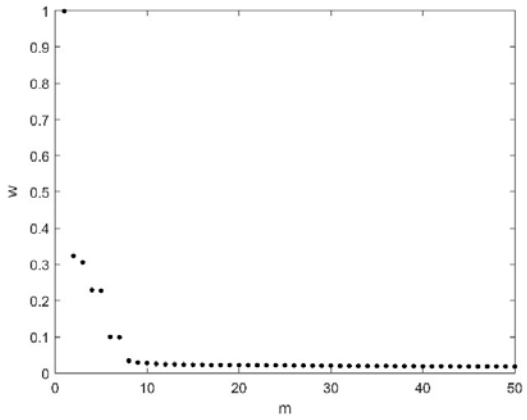

Fig. 5. Rank the cluster center weight in a descending order.

We use a slope of the line between two points representing the declining trend of the cluster center weight, that is k_i^m.

$$k_i^m = \frac{w_{i+m} - w_i}{m} \tag{6}$$

Where k_i^m represents the average rate of change of the cluster center weight in the interval $[i, i+m]$. It describes the overall trend of w in this interval.

Definition 4 The inflection point. We define the inflection point x as Eq. (7).

$$x = arg\ (max\left(\frac{k_i^1}{k_1^{i-1}}\right)) \tag{7}$$

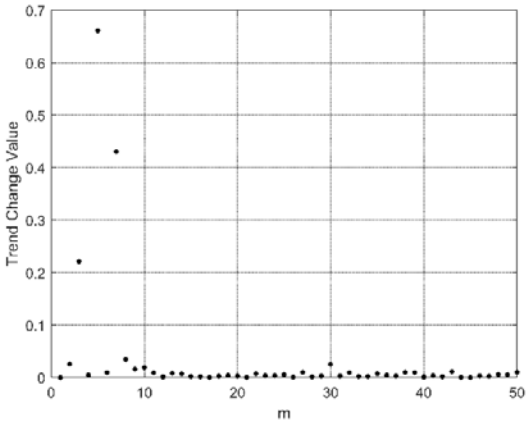

Fig. 6. Inflection point decision graph.

From Eq. (6), we can see that k_1^{i-1} is the slope from the first point to the i-th point and represents the average rate of change of the point set $\{1, 2, \ldots, i\}$. k_i^1 is the slope of the i-th point to the $i+1$-th point. Therefore, the inflection point can be expressed as the critical point. At this point the overall trend of deviation changes the most rapidly. **Fig. 6** shows the trend of deviation. In this figure, the maximum point is the inflection point. The points in front of the inflection point are cluster center points.

Algorithm 1 shows specific steps of the cluster center selection algorithm.

Algorithm 1: Cluster center selection

Input:

The local density ρ and the local distance δ

Output:

Cluster center points CI

1. Find the cluster center weight w for each point.
2. Sort the cluster center weight in a descending order.
3. Calculate k_i^1、 k_1^{i-1}, and find the maximum value of $\frac{k_i^1}{k_1^{i-1}}$ to determine the inflection point $i = x$.
4. Set points $\{1,2, \ldots, x\}$ in front of the inflection point as cluster center points.

6.1.2 The flow of the algorithm

Algorithm 2 describes steps of the optimization algorithm of CFSFDP.

Algorithm 2: The optimization algorithm of CFSFDP

Input:

Data set $Data$, the number of neighbors K, the belonging cluster $C_i = -1$ of each sample i (means that all the samples are not assigned).

Output:

Clustering results C

1. Calculate the distance between samples and ρ, δ for each sample.
2. Get the cluster center set CI according to the cluster center selection algorithm.
3. Perform sample allocation strategy to allocate points except cluster centers.
4. Get clustering results C.

Finally, the clustering is completed and we get the cluster information to which each sample belongs.

6.2 Initial segmentation processing

After the data extraction, timing reducing and calculating the distance between every two motion frames, we consider each frame as a data point in high-dimensional space and motion segments of different motion types correspond to different clusters. Next, we apply the optimization algorithm of CFSFDP for clustering human motion data. After clustering the motion data of n frames into k clusters, we combine consecutive frames with the same class label to form motion segments of different lengths. Finally, we can get the result of initial segmentation that G encoding the motion type each motion segment corresponding to and E saving the start and end frames of each motion segment.

7. Optimized Segmentation Based on Aligned Cluster Analysis

ACA [2] has a good performance regarding accuracy, but it has some shortcomings. Subtly, the initial segmentation method can make up for the lack of ACA, and ACA is able to optimize the results of the initial segmentation. Thus, we take the results of the initial segmentation as input and perform optimized segmentation based on the ACA method.

ACA is “aligned cluster analysis”. The word “aligned” means this method brings in the concept of dynamic time alignment kernel (DTAK) for measuring the distance between two sequence fragments of different lengths. The word “cluster” represents it introduces the kernel k-means clustering algorithm for sequence fragments clustering. The idea of ACA is to combine the kernel k-means clustering algorithm with the dynamic time alignment kernel to cluster the sequence segments with variable length instead of the motion frames. After that, we get an energy function and transform the segmentation problem into an energy minimization problem. Solving this problem using coordinate descent strategy and dynamic programming algorithm, the segmented sequence fragments are obtained finally.

When we apply ACA to carry out optimized segmentation, we mainly take advantage of the statistical characteristics of data. The statistical characteristics mainly refer to the characteristics of the distribution of data in statistics. And it has nothing to do with what the data represents. Therefore, this part is making use of the high-level data information of the motion data to finish segmentation.

7.1 K-means clustering and its kernel method

K-means clustering is based on the sum of squared error criterion. For the segmentation of human motion capture data, clustering is about dividing motion sequences consisting of n frames into k uncorrelated clusters. We are capable to introduce the k-means clustering algorithm to cluster motion data. Supposing the sample set matrix $X \in R^{d\times n}$, $x_i \in R^d$ is the i-th data point, the energy function is as follows [18,19]:

$$J_{km}(Z,G) = \sum_{c=1}^{k}\sum_{i=1}^{n} g_{ci}\|x_i - z_c\|^2 = \|X - ZG\|_F^2 \tag{8}$$

where Z is the set of all cluster centers, $z_c \in R^d$ represents the center of class c, $G \in \{0,1\}^{k\times n}$ represents a recognition matrix. If the sample point x_i belongs to c, then $g_{ci} = 1$, otherwise $g_{ci} = 0$.

K-means clustering uses the mean of result as a representative data of a class. This determines that only when the natural distribution of the class is spherical or nearly spherical, we could achieve a better result. In order to overcome this limitation, the kernel k-means is introduced. The kernel is used to map the data into a high-dimensional space [20]. The new data is described as $\phi(x_i)$. By calculating the Gaussian kernel function, the similarity between two frames x_i and x_j is that:

$$k_{ij} = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{9}$$

The energy function of kernel k-means is:

$$J_{kkm}(G) = \sum_{c=1}^{k}\sum_{i=1}^{n} g_{ci} \underbrace{\|\phi(x_i) - z_c\|^2}_{dist_\phi^2(x_i,z_c)} = \|\phi(X) - ZG\|_F^2 \tag{10}$$

$$dist_\phi^2(x_i, z_c) = k_{ii} - \frac{2}{n_c}\sum_{j=1}^{n} g_{cj}k_{ij} + \frac{1}{{n_c}^2}\sum_{j_1,j_2=1}^{n} g_{cj_1}g_{cj_2}k_{j_1j_2}, n_c = \sum_{j=1}^{n} g_{cj} \tag{11}$$

where x_i is the i-th sample, z_c is the center of class c, $dist_\phi^2(x_i, z_c)$ represents the squared distance of them, n_c is the total number of samples in the class c.

7.2 Dynamic time alignment kernel

Dynamic time warping (DTW) is the most commonly used time-series alignment method, but it does not satisfy the triangle inequality. Based on the DTW, Shimodaira et al. [21] proposed a dynamic time alignment kernel (DTAK) method.

To better understand the DTAK and apply it to the segmentation of the motion data, we give two time-series $X = [x_1, x_2, \dots, x_{n_x}] \epsilon R^{d\times n_x}$ and $Y = [y_1, y_2, \dots, y_{n_y}] \epsilon R^{d\times n_y}$. After that, the DTAK value between two-series can be defined as:

$$\tau(X, Y) = \frac{u_{n_x n_y}}{n_x + n_y} \tag{12}$$

where

$$u_{ij} = max\begin{cases} u_{i-1,j} + k_{ij} \\ u_{i-1,j-1} + 2k_{ij} \ (u_{11} = 2k_{11}) \\ u_{i,j-1} + k_{ij} \end{cases} \tag{13}$$

u_{ij} represents the elements in the cumulative similarity matrix U, $k_{ij} = \phi(x_i)^T\phi(y_i)$ represents the frame kernel, and here we use Gaussian kernel. The following is an example of building the cumulative similarity matrix U for two short sequences and calculating the dynamic time alignment kernel.

We suppose two time-series are $X_{1\times 6} = [1,3,2,4,3,3]$ and $Y_{1\times 7} = [1,2,4,1,4,3,3]$. According to the Gaussian kernel function given above, the similarity matrix $K \in R^{6\times 7}$ of these two time-series can be calculated. For simplicity, we set σ to infinity towards zero. Because the two-time series with different lengths of time and need to be aligned, we calculate the cumulative similarity matrix U of the two-series by Eq. (13). Finally, the lower right corner of U is normalized according to Eq. (12). In this way, we get the value of DTAK of the two time-series: $\tau(X,Y) = \frac{11}{13} = 0.85$.

7.3 The energy function of ACA

Given a motion sequence $X = [x_1, x_2, \dots, x_n] \in R^{d\times n}$, ACA decomposes X into m independent fragments and each fragment corresponding to one of k classes. Let $S_i = [x_{e_i}, \dots, x_{e_{i+1}-1}] = x_{[e_i,e_{i+1})} \in R^{d\times n_i}$ denotes the i-th segment. We use E saving the start and end frames of each motion segment. Assuming n_{max} represents the maximum length of segment, it determines the interval size of the decomposition. Here, we give a recognition matrix $G \in \{0,1\}^{k\times m}$. If segment S_i belongs to class c, then $g_{ci} = 1$, otherwise, $g_{ci} = 0$.

We combine the kernel k-means with DTAK to cluster the sequence segments [22]. The aligned cluster analysis method extends the kernel k-means method by minimizing Eq. (14).

$$J_{aca}(G,E) = \sum_{c=1}^{k}\sum_{i=1}^{m} g_{ci} \underbrace{\left\|\psi\left(X_{[e_i,e_{i+1})}\right) - \psi(z_c)\right\|^2}_{dist_{\psi}^2(S_i,z_c)} \tag{14}$$

$S_i = X_{[e_i,e_{i+1})}$ represents a segment, $dist_{\psi}^2(S_i,z_c)$ is the square of the distance between the i-th segment in the feature space and the center of class c [23]. And it is defined by the nonlinear mapping $\psi(\cdot)$.With reference to the calculation method of the distance between any sample in kernel space and the mean of a class, we can get the following equation:

$$dist_{\psi}^2(S_i,z_c) = \tau_{ii} - \frac{2}{m_c}\sum_{j=1}^{m} g_{cj}\tau_{ij} + \frac{1}{m_c^2}\sum_{j_1,j_2=1}^{m} g_{cj_1}g_{cj_2}\tau_{j_1j_2} \tag{15}$$

where $m_c = g_{c1} + g_{c2} + \cdots + g_{cm}$ is the number of fragments belonging to class c. The dynamic kernel function τ is defined by the following equation.

$$\tau_{ij} = \psi(S_i)^T\psi\left(S_j\right) \tag{16}$$

7.4 Optimized segmentation processing

After the initial segmentation is complete, the optimized segmentation is performed. We construct an energy function according to Eq. (14), and then transform the motion segmentation problem into an integer programming problem of two variables (G and E). But optimizing G and E is a NP hard problem. We use a coordinate descent strategy to alternately compute G and E. The specific approach is to use the Winner-take-all strategy [24] to calculate G and use dynamic programming algorithm to calculate E. At each

iteration we should solve the following sub-problem:

$$G, E = arg\,min_{G,E} J_{aca}(G, E) = arg\,min_{G,E} \sum_{c=1}^{k} \sum_{i=1}^{m} g_{ci} dist_{\psi}^{2}(S_i, \dot{z}_c) \quad (17)$$

where $\dot{z}_c$ is the mean of the cluster calculated from the initial segmentation result $(\dot{G}, \dot{E})$. Repeat calculation steps until convergence, and we get the final optimized segmentation result.

8. Experiments

We conducted some experiments to evaluate the proposed framework for segmenting human motion sequences.

8.1 Experiment design

The experiment data we used is Subject 86 dataset [13] (Carnegie Mellon University) and HDM05 dataset [25] (University of Bonn). The Subject 86 dataset is used by most segmentation methods. It contains 14 motion sequences, each of them ranges from 4579 to 10617 frames. These data include walking, running, jumping, boxing, kicking and other common sports. The HDM05 dataset contains 12 motion sequences consisting of common sports. We first use Subject 86 dataset to evaluate the segmentation performance in multiple ways, and then use HDM05 dataset to test the generalization capabilities of our framework.

In this experiment, the number of key joints $J = 12$, the number of neighbors $K = 5$, and the length limit $n_{max} = 60$.

8.2 Experiment results and analyses

At first, we perform segmentation experiments on Subject 86 dataset and we also do contrast experiments using TMM [11], Nystrom [12], HACA [7], ACA [2] and GMM [6] method. As shown in **Fig. 7**, we give the results of manual segmentation and these six methods to segment the 14 motion sequences separately.

The first line in each small figure of **Fig. 7** is ground truth, which is the result of segmentation by human watching the motion sequence frame by frame. Therefore, the ground truth is the result of manual segmentation. Due to the continuity of human motion, for the ground truth, we consider that the frames in the transition period of the motion type are segmentation frames. The black line of ground truth in the figure indicate the range where the segmentation frames are located. For the non-manual methods, the segmentation frame is a single frame that separates two different types of motion. Thus, the segmentation frame of non-manual method is considered to be correct as long as it falls within the corresponding range of the ground truth.

The last few lines in small figure are segmentation results of our framework, TMM, Nystrom, HACA, ACA and GMM, respectively. For each motion sequence, different colors represent different motions, and the same colors represent the same motions.

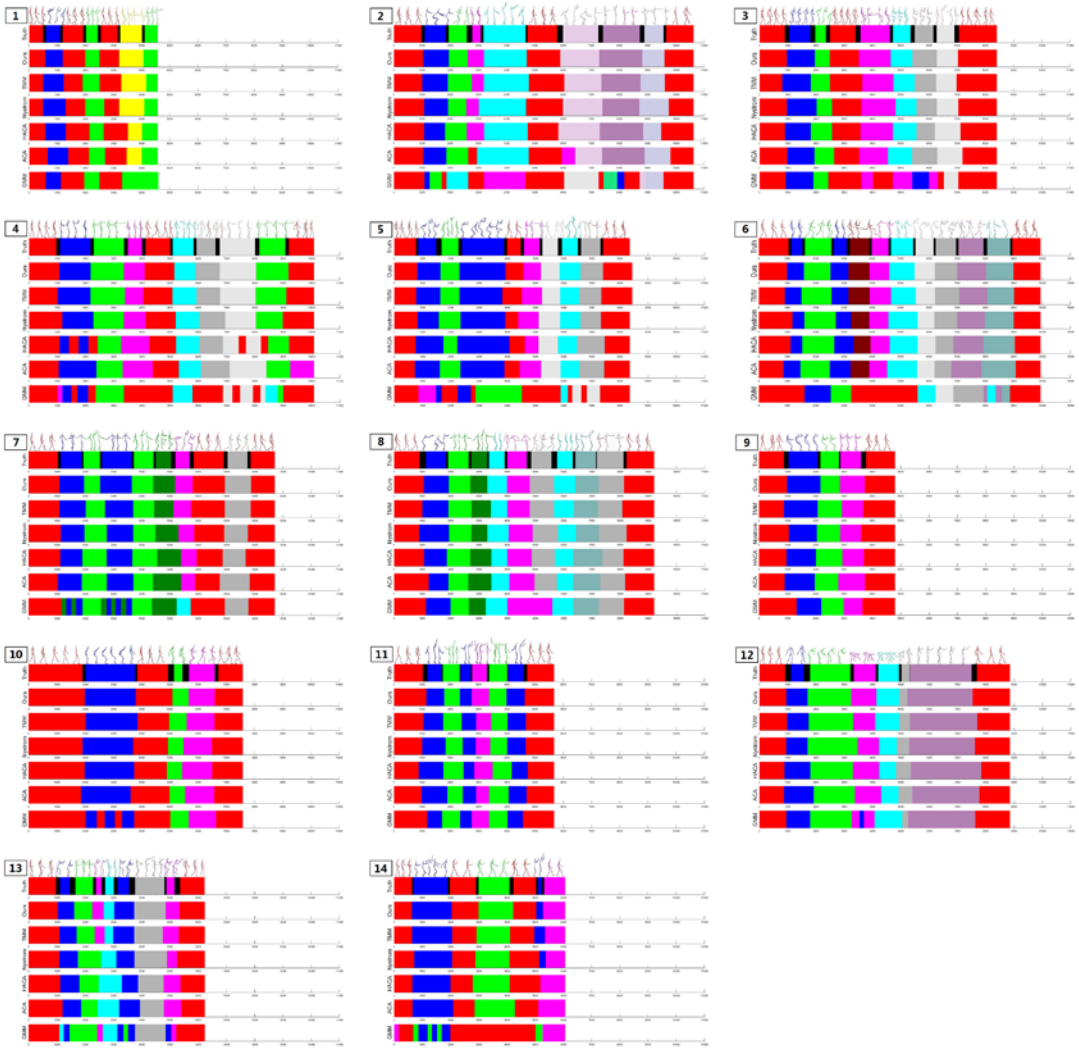

Fig. 7. The results of segmentation.

In order to evaluate the accuracy of segmentation results, we need to compare segmentation results with ground truth. We use the following equation to calculate the confusion matrix of segmentation results (G_{alg}, H_{alg}) and the ground truth (G_{tru}, H_{tru}):

$$C = G_{alg} H_{alg} H_{tru}^{T} G_{tru}^{T} \in R^{k \times k} \tag{18}$$

Among them, G is the recognition matrix, which represents the correspondence between each independent segment and the cluster, the indication matrix H denotes the correspondence between each sample and segment.

Subsequently, we use the Hungarian algorithm to calculate the accuracy.

$$acc = max_{P} \frac{tr(CP)}{tr(C1_{k \times k})} \tag{19}$$

Where $P \in \{0,1\}^{k\times k}$ is a permutation matrix.

We use accuracy to evaluate the segmentation results of our framework and five methods for processing 14 motion sequences. **Fig. 8** indicates the average accuracy of each method for per sequence. From **Fig. 8**, we are able to know that our framework has the best performance in these methods with respect to average accuracy.

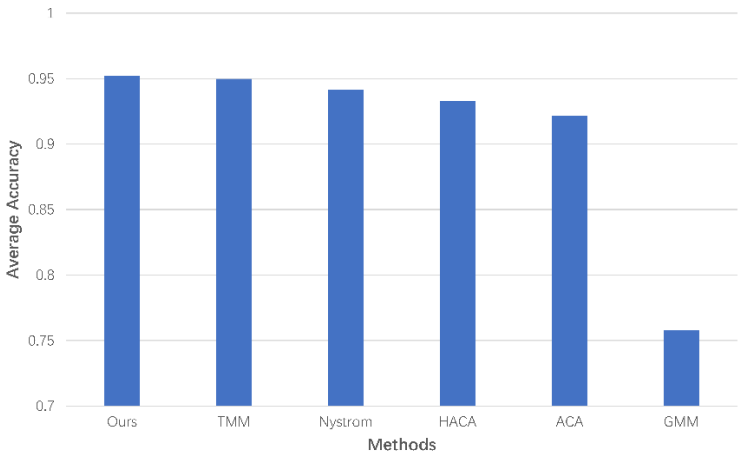

Fig. 8. Average accuracy of each method for per sequence.

Also, we employ the standard deviation to measure the stability of these six segmentation methods according to their accuracy performance on 14 motion sequences. **Fig. 9** illustrate the standard deviation of the six methods. We can easily get that the standard deviation of our framework is the smallest and our method is the best in terms of stability.

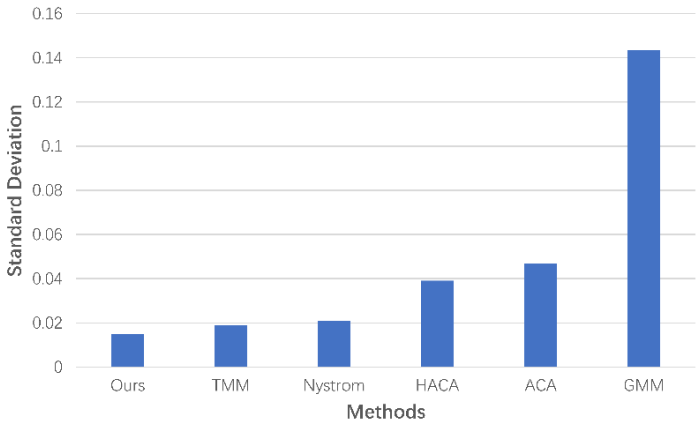

Fig. 9. Standard deviation.

In addition to accuracy, we could evaluate experimental results based on the segmentation frames reported. The evaluation metric we used is the precision/recall framework.

$$precision = \frac{ReportedCorFrames}{ReportedFrames} \times 100\% \tag{20}$$

$$recall = \frac{ReportedCorFrames}{CorFrames} \times 100\% \tag{21}$$

Where ReportedCorFrames indicates the correct segmentation frames reported, ReportedFrames represents is the total number of segmentation frames reported and CorFrames represents the total correct segmentation sections processing by human. Precision is defined as the percentage of the number of correct segmentation frames reported in all reported segmentation frames. Recall is defined as the percentage of the number of correct segmentation frames reported in all correct frames of manual segmentation. The closer the values of precision and recall are to one, the better the result of the segmentation. **Fig. 10** gives scores of precision and recall for these six methods. We can get that our framework is better than the other five methods respecting precision/recall performance according to this figure.

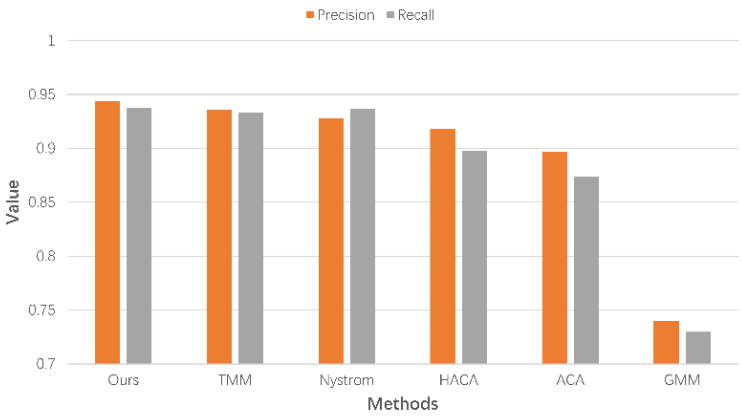

Fig. 10. Precision and recall.

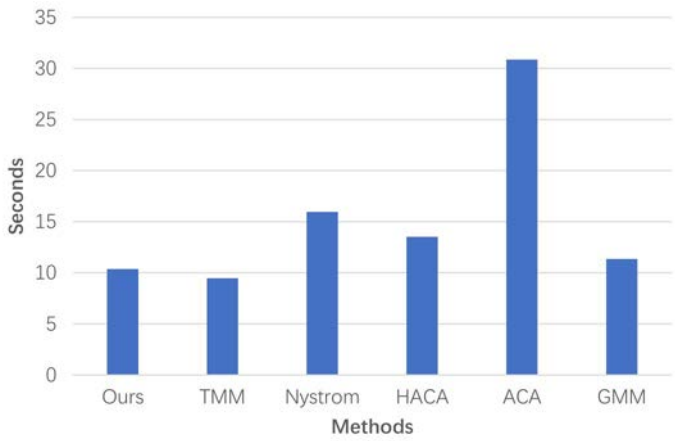

Fig. 11. Average time cost by each method for per sequence

We recorded the time spent by these six methods to complete each segmentation task. **Fig. 11** indicate the average time cost by each method for per sequence. From **Fig. 11** we could get that our framework is just a little slower than TMM, but faster than the other four methods. Therefore, our framework has a good performance in terms of speed.

Because of the optimization algorithm of CFSFDP, our framework only needs to specify the number of neighbors K. For all sequences to be segmented, it could calculate the number of motion categories automatically and complete the segmentation work without artificial specifying.

According to experimental experience, we can set the K as 5 or 6 or 7 generally. From **Fig. 12** we can see that the framework is insensitive to the value of K when the motion sequence of 3,5,8,9 and 12 are processed and the value of K is 5 to 7. The theoretical explanation is that K has no effect on the selection of the cluster center and the K only affect the allocation of the extreme isolated points. However, there are very few extreme isolated points in actual situations especially in human motion segmentation.

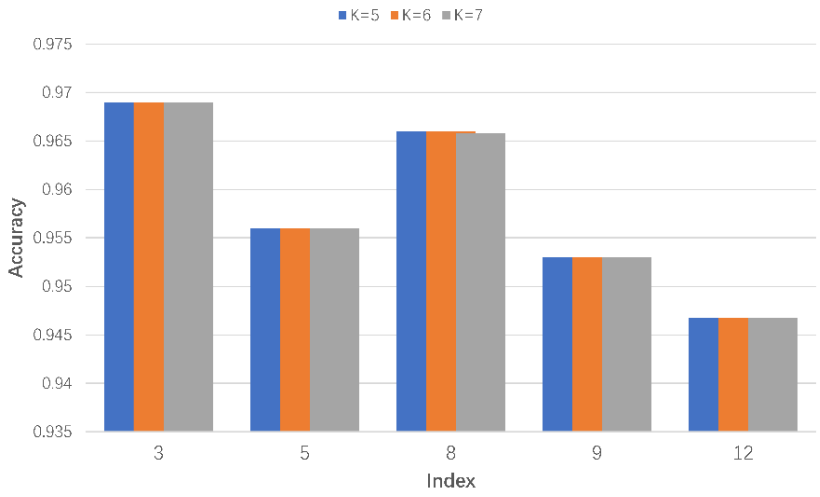

Fig. 12. Accuracy of the framework when K is 5, 6, and 7.

Table 2. Segmentation performance on Subject 86 dataset and HDM05 dataset.

	Ours			**TMM**			**Nystrom**		
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.
Subject 86	0.952	0.943	0.938	0.949	0.936	0.933	0.941	0.928	0.936
HDM05	0.881	0.885	0.929	0.870	0.868	0.904	0.814	0.807	0.839

To test the generalization capabilities of our framework, we also performed experiments on HDM05 dataset. **Table 2** shows the segmentation performance of our framework, TMM and Nystrom on Subject 86 dataset and HDM05 dataset. As can be seen from **Table 2**, our framework also has a good segmentation performance on HDM05 dataset.

9. Conclusion

This paper proposes an unsupervised framework using low-level physical information and high-level data information of human motion data to finish human motion segmentation. First, we introduce the algorithm of CFSFDP and optimize it to carry out initial segmentation and obtain a good result quickly. Second, we use the ACA method to perform optimized segmentation for improving the result of segmentation. The ingenious combination of these two parts allows us to get excellent segmentation results.

Our framework uses both motion characteristics and statistical characteristics of data and helps us make use of the data information fully to accomplish human motion segmentation. And we construct a combined feature of center distance and velocity. This combined feature could truly represent the characteristics of motion data and make a great contribution to improving the performance of the segmentation algorithm. What's more, we introduce the algorithm of CFSFDP and optimize it to solve the human motion segmentation problem. It is able to determine the number of motion categories and generate a good clustering result as the input of the optimized segmentation. Moreover, we skillfully combining the optimization algorithm of CFSFDP and ACA which let the former make up for the latter's shortcomings and make the performance of the latter better. Through experiments we get that our framework has a nice performance in terms of accuracy and speed.

Although we try to make the framework as simple as possible and use fewer memory resources, there is still much work to do in this regard. In future work, we should simplify the framework to make it take up fewer memory resources and have a lower time complexity. At the same time, we also need to minimize the impact of parameter setting on the result and make the framework more intelligent and universal.

Acknowledgments

This research is partially supported by National Natural Science Foundation of China (No.61876018, No. 61370127, No.61473031, No.61472030), Program for New Century Excellent Talents in University (NCET-13-0659).

References

[1] A. Rodriguez, A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1491-1496, 2014. Article (CrossRef Link)

[2] F. Zhou, F. De la Torre, J. K. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," in *Proc. of IEEE Conference on Automatic Face and Gestures Recognition*, pp. 1-7, 2008. Article (CrossRef Link)

[3] M. Müller, T. Röder, M. Clausen, "Efficient content-based retrieval of motion capture data," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 677-685, 2005. Article (CrossRef Link)

[4] T. Kwon, S. Y. Shin, "Motion modeling for on-line locomotion synthesis," in *Proc. of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 29-38, 2005. Article (CrossRef Link)

[5] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Trans. Image Process*, vol. 24, no. 11, pp. 3939-3949, 2015. Article (CrossRef Link)

[6] J. Barbič, A. Safonova, J. Pan, C. Faloutsos, J.K. Hodgins, N.S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proc. of Graphics Interface 2004*, pp. 185-194, 2004. Article (CrossRef Link)

[7] F. Zhou, F. De la Torre, J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 35, no. 3, pp.582-596, 2013. Article (CrossRef Link)

[8] D. Gong, G. Medioni, X. Zhao, "Structured time series analysis for human action segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell*. vol. 36, no. 7, pp. 1414-1427, 2014. Article (CrossRef Link)

[9] M. Devanne, H. Wannous, P. Pala, S. Berretti, M. Daoudi, A. Del Bimbo, "Combined shape analysis of human poses and motion units for action segmentation and recognition," in *Proc. of IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recog*, pp. 1-6, 2015. Article (CrossRef Link)

[10] M. Devanne, S. Berretti, P. Pala, H. Wannous, M. Daoudi, A. D. Bimbo, "Motion segment decomposition of rgb-d sequences for human behavior understanding," *Pattern Recognit*, vol. 61, pp. 222-233, 2017. Article (CrossRef Link)

[11] B. Krüger, A. Vögele, T. Willig, "Efficient unsupervised temporal segmentation of motion data," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 797-812, 2017. Article (CrossRef Link)

[12] X. Yu, W. Liu, W. Xing, "Behavioral segmentation for human motion capture data based on graph cut method," *Journal of Visual Languages & Computing*, vol. 43, pp. 50-59, 2017. Article (CrossRef Link)

[13] CMU Graphics Lab Motion Capture Database. Article (CrossRef Link)

[14] K. Forbes, E. Fiume, "An efficient search algorithm for motion data using weighted PCA," in *Proc. of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 67-76, 2005. Article (CrossRef Link)

[15] R. Lan, H. Sun, "Automated human motion segmentation via motion regularities," *Vis. Comput*, vol. 31, no. 1, pp. 35-53, 2015. Article (CrossRef Link)

[16] S. Peng, X. Liu, "Double-feature combination based approach to motion capture data behavior segmentation," *Computer Science*, vol. 40, no. 8, pp. 303-308, 2013. Article (CrossRef Link)

[17] J. Xie, H. Gao, W. Xie, "K-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset," *Scientia Sinica (Informationis)*, vol. 46, pp. 258-280, 2016.

[18] F. De la Torre, T. Kanade, "Discriminative cluster analysis," in *Proc. of the 23rd international conference on Machine learning*, pp. 241-248, 2006. Article (CrossRef Link)

[19] H. Zha, X. He, C. Ding, M. Gu, "Spectral relaxation for k-means clustering," in *Proc. of Advances in Neural Information Processing Systems 14 (NIPS'01)*, pp. 1057-1064, 2001. Article (CrossRef Link)

[20] I. S. Dhillon, Y. Guan, B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proc. of the 10th ACM SIGKDD International Conference*, pp. 551-556, 2004. Article (CrossRef Link)
[21] H. Shimodaira, K. Noma, M. Nakai, S. Sagayama, "Dynamic time-alignment kernel in support vector machine," in *Proc. of Advances in NIPS14*, pp. 921-928, 2001. Article (CrossRef Link)
[22] F. Zhou, F. De la Torre, J. F. Cohn, "Unsupervised discovery of facial events," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574-2581, 2010. Article (CrossRef Link)
[23] F. De la Torre, "A least-squares framework for component analysis," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, no. 6, pp. 1041-1055, 2012. Article (CrossRef Link)
[24] S. Roweis, Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Computation*, vol. 11, no. 2, pp. 305-345, 1999. Article (CrossRef Link)
[25] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, A. Weber, "Documentation mocap database hdm05," *Tech. Rep. CG-2007-2, Universität Bonn*, 2007. Article (CrossRef Link)

Xiaofei Zan received his B.S. degree in Computer Science and Technology from Tianjin University of Science and Technology. He is pursuing his M.S. degree in Signal and Information Processing at Beijing Jiaotong University, China. His research interest is computer animation which includes human motion segmentation, human motion synthesis and character animation.

Weibin Liu received the Ph.D. degree in Signal and Information Processing from Institute of Information Science at Beijing Jiaotong University, China, in 2001. During 2001-2005, he was a researcher in Information Technology Division at Fujitsu Research and Development Center Co., LTD. Since 2005, he has been with the Institute of Information Science at Beijing Jiaotong University, where currently he is a professor in Digital Media Research Group. He was also a visiting researcher in Center for Human Modeling and Simulation at University of Pennsylvania, PA, USA during 2009-2010. His research interests include computer vision, computer graphics, image processing, virtual human and virtual environment, and pattern recognition. He is a member of the IEEE, ACM, IEICE and CCF.

Weiwei Xing received her B.S. degree in Computer Science and Technology and Ph.D. degree in Signal and Information Processing from Beijing Jiaotong University, in 2001 and 2006 respectively. During 2011-2012, she was a visiting scholar at University of Pennsylvania. Currently, she is a professor at School of Software Engineering, Beijing Jiaotong University. Her research interests mainly include intelligent information processin