Efficient Unsupervised Behavioral Segmentation of Human Motion Capture Data

Xiaomin Yu[1], Weibin Liu*[1] and Weiwei Xing[2]

[1]Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[2]School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China
E-mail:wbliu@bjtu.edu.cn

Abstract *-With the development of human motion capture, realistic human motion capture data has been widely implemented to many fields. However, segmenting motion capture data sequences manually into distinct behavior is time-consuming and laborious. In this paper, we introduce an efficient unsupervised method based on graph partition for automatically segmenting motion capture data. For the N-Frame motion capture data sequence, we construct an undirected, weighted graph G=G(V,E), where the node set V represent frames of motion sequence and the weight of the edge set E describes similarity between frames. In this way, behavioral segmentation problem on motion capture data may be transformed into graph cut problem. However, the traditional graph cut problem is NP hard. By analyzing the relationship between graph cut and spectral clustering, we apply spectral clustering to the NP hard problem of graph cut. In this paper, two methods of spectral clustering, t-nearest neighbors and the Nystrom method, are employed to cluster motion capture data for getting behavioral segmentation. In addition, we define an energy function to refine the results of behavioral segmentation. Extensive experiments are conducted on the dataset of multi-behavior motion capture data from CMU database. The experimental results prove that our novel method is robust and effective.*

Keywords: *motion capture data, behavioral segmentation, graph cut, spectral clustering*

I. INTRODUCTION

Human motion capture, once associated with producing special effects for film and television production, entertainment games and virtual reality, is common today in diverse applications ranging from health care to consumer electronics. The data of human motion capture can be extensively applied in many fields, such as producing realistic animation movies [1][2],physical rehablitation [3][4]and ergonomic analysis[5]. However, the cost of capturing motion data is very high. It's necessary to better reuse motion capture data. Motion capture data sequence is usually comprised of multiple types of behaviors. The present work suggests that automatic segmentation of human motion capture data into distinct behavior based on statistical properties of the motion can be an efficient and quite robust alternative to hand segmentation. This paper focuses on efficient and robust technique which can be able to automatically segment motion capture data sequence into distinct behaviors, as depicted in Figure 1. The main contributions of this paper can be concluded as follows:

(1) Constructing N-Frame motion capture data sequence into an undirected, weighted graph G=G(V,E), which is based on the similarity of data frames. According to this way, we transform motion capture data segmentation problem into the problem of graph cut.

(2) We employ an algorithm, which can automatically extract the number of behavior and cluster centers, to the motion capture data. It's based on an unsupervised cluster method. We consider the cluster numbers as the behavior numbers for motion capture data sequence.

(3) Analyzing in detail for graph cut method and spectral clustering, we transform this NP hard problem into eigenvalues and eigenvectors in spectral space. Two methods of spectral clustering, t-nearest neighbors and the Nystrom method,are employed to cluster motion capture data for getting behavioral segmentation.

(4) For the clustering fragments, we define energy function and use dynamic programming to refine the results of behavioral segmentation for motion capture data.

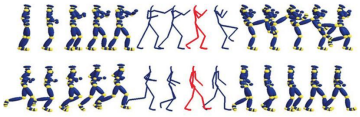


Figure 1: Behavior segmentation. Segmenting motion capture data into distinct behavior.

*Corresponding author: Weibin Liu, wbliu@bjtu.edu.cn

DOI reference number: 10.18293/DMS2016-016

The remainder of this paper is organized as follows. Firstly, we introduce works of predecessors for motion segmentation in Section II. In Section III, how to calculate similarity of motion capture data frames is adopted. The details for identifing behavior's clusters centers for original motion capture data sequences are presented in Section IV. In Section V, we analyze in detail for the relationship between graph cut and spectral clustering. Then, transforming this NP hard problem of graph cut into spectral clustering problem. In this way, realizing behavioral segmentation for motion capture data sequence. Details of energy function for refining the results of behavioral segmentation is represented in Section VI. In section VII, we introduce the experiments, which are conducted on the dataset of multi-behavior motion capture data from CMU database. Finally, we provide the discussion and conclusion for this behavior segmentation method in Section VIII.

II. RELATED WORK

Temporal segmentation is related to numbers of different fields such as data mining [6][7], behavior recognition [8] and so on. Researchers have proposed several techniques to segment motion capture data into distinct behaviors. Maybe they focus on different perspectives and concerns, their purpose is to establish efficient and robust segmentation methods for motioncapture data. The methods can be generally categorized into several types: classifier, clustering, machine learning and dimension reduction and so on.

Supervised learning methods generally formulize segmentation motion capture data as a classification problem, where classifiers are trained from a carefully selected training set. In other words, Its mean that they are usually applying example segments or pre-computed templates and matching them to test sequences. Muller et al.[9]constructed motion templates to behaviors segmentation. Lv et al. [10]defined Hidden Markov Models (HMM) to realize human Motion capture data segmentation. Support Vector Machine(SVM) which is based on an annotation training database to segment motion capture data was constructed by Arikan et al. [11]. However, these methods relied on the training set and they would fail to pick up the segments whose corresponding behaviors were not contained in the training set.

It's natural for researchers to use another technique to overcome this limitation, which can be named as unsupervised learning methods. In these methods, motion capture data segmentation is located by clustering motion frames. zhou et al.[12]used aligned cluster analysis (ACA) to temporally cluster poses into motion primitives which were then assigned to different behavior classes. ACA extends standard kernel k-means clustering: the cluster means include a number of features and a dynamic time warping (DTW) kernel is used to achieve temporal invariance. However, ACA method needs users determine the cluster number with respect to temporal constraint. In order to overcome this limitation, zhou et al. [13] derived an unsupervised hierarchical bottom-up framework, which is called hierarchical aligned cluster analysis (HACA) to realize segmentation. HACA provided a crude method to find a low-dimensional embedding for the time series. HACA is efficiently optimized with a coordinate descent strategy and dynamic programming.

The researchers tried to solve the dilemma of non-learning way to behavior segmentation. Balazia et.al [14] introduced an unsupervised key-pose detection algorithm for segmentation of motion capture data and this proposed algorithm partition motions at the level of gestures. Barbic et al. [15] chose segments using an indication of intrinsic dimensionality from Principal Component Analysis (PCA). Its based on the observation that simple motions exhibit lower dimensionality than more complex motions. As an extension of the traditional PCA, Barbic et al. defined a proper probability model for PCA which is named probabilistic PCA (PPCA). We can easily know that the directions outside the subspace were discarded, whereas they were modeled with noise in PPCA.

III. SIMILARITY MEASURE OF MOTION CAPTURE DATA

A. *Distance of Frames*

This paper employs the human skeleton model which has 31 joints, as illustrated in Figure 2. For every frame, it has 62-dimensions, which includes root position vector, root orientation vector and other joints' direction vector. The ith frame's pose consists of all joints rotation angle in the ith frame expect the root position vector and the root orientation vector which including 6-dimensional. Each pose $p_i = \{a_{i,1}, a_{i,2}, a_{i,3} \dots a_{i,56}\}$ is represented as a point in 56-dimensional,which $a_{i,j}$ is one of an Euler angle. The velocity v_i of the ith frame is computed by the Euclidean distance between p_i and p_{i+1}.

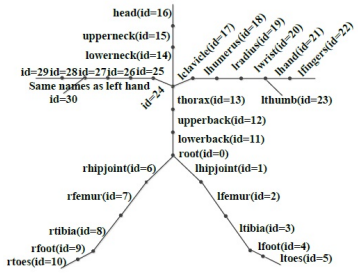


Figure 2: Human Skeleton Model

$$v_i = \begin{cases} \sqrt{(a_{i+1,1} - a_{i,1})^2 + \ldots + (a_{i+1,56} - a_{i,56})^2} & i \neq n \\ v_{i-1} & i = n \end{cases} \quad (1)$$

Calculating the distance by:

$$d_{i,j} = \alpha d(p_i, p_j) + \beta d(v_i, v_j) \quad (2)$$

Where $d(v_i, v_j)$ is the difference of velocity between the ith frame and the jth frame and $d(p_i, p_j)$ is the weighted difference of joint orientations. The weights of α and β are set to 0.5. The term of $d(p_i, p_j)$ is given by:

$$d(p_i, p_j) = ||p_{i,0} - p_{j,0}||^2 + \sum_{k=1}^{m} w_k ||log(q_{j,k}^{-1} q_{i,k})||^2 \quad (3)$$

In (3), $p_{i,0}, p_{j,0} \in R^3$ are the global translational positions of the figure at frame i and j, respectively; m is the number of joints; and $q_{i,k}, q_{j,k} \in S^4$ are the orientations of joint k and frames i and j, respectively. The log-norm term represents the geodesic norm in quaternion space. w_k describes the weight of k-th joint. Lee et al [16] set the weights manually. While the weights of unimportant joints are set to zero, the weights of important joints, such as shoulders, elbows, hips, knees, hips and so on, are set to one. For different joint, it's well known to occupy different role. In this way, there will be a larger error. Wang et al [17] compute a set of optimal weights for the cost function using a constrained least-squares technique, where the weights are calculated in two ways: through a cross-validation study and a medium-scale user study. In order to reduce the error, we use the optimal weights proposed by Wang and Bodenheimer, which are shown in table 1.The weights of remaining joints are set to zero. According to this, we can get a distance matrix D_{n*n} , where n is the length of the original motion capture data sequence. Apparent that $d_{ij} = d_{ji} (i \neq j)$ and $d_{ij} = 0 (i = j)$.

B. *Frame Kernel Matrices*

In general, there are several methods which are used to indicate the similarity of frames for motion capture data. There are several different formulas to define it. In order to better construct the similarity matrix, we optimized the novel representation method of frames similarity. Calculating the similarity of frames by:

$$w_{ij} = e^{(-\frac{dist_{ij}^2}{2\sigma_i \sigma_j})} \quad (4)$$

It's time-consuming to get an optimum parameter σ_i for this function by multiple experiments. In order to determine the value of parameter σ_i, a parameter construction method of neighborhood adaptive scale [18] is introduced:

Table 1: Joints with non-zero weights.

Joints	Weight
Right and Left Hip	1.0000
Right and Left Knee	0.0901
Right and Left Shoulder	0.7884
Right and Left Elbow	0.0247

$$\sigma_i = \frac{1}{k} \sum_{m=1}^{K} D_{ii_m} = \frac{1}{k} \sum_{m=1}^{K} ||i - i_m|| \quad (5)$$

Setting the k that the average number of neighbors is around 1% of the total frame number for every motion capture data sequences. The distance matrix between frames is computed by (2).

IV. IDENTIFY CENTERS OF BEHAVIORAL CLUSTERS

In order to automatically extract the number of motion categories for motion capture data sequence, we use this algorithm, which has good effect for any data points shape. As always, this algorithm has its basis only in the distance between data points. It's able to detect non-spherical clusters and automatically find the correct number of clusters.

This algorithm has its basis in the assumption that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any point with a higher local density. It's clear that the same motion behavior with sufficient similarity. We consider every motion capture data frame as a high-dimensional point. For each high-dimensional point, we should compute two quantities: its local density ρ_i and its distance δ_i from points of higher density.

Rodriguez et al. [19] introduce two methods for computing local density:

$$\rho_i = \sum_{j \in I_s \setminus \{i\}} X(d_{ij} - d_c) \quad (6)$$

$$X(x) = \begin{cases} 1 & x < 0 \\ 0 & x \geq 0 \end{cases} \quad (7)$$

d_c is a cut-off distance. ρ_i describes the number of points, which the distance with i is smaller than d_c . Obviously, ρ_i is discrete.

We can see that formula ρ_i has a significant limitation of different motion capture data points with the same local density. In order to avoid this limitation, we calculate ρ_i by:

$$\rho_i = \sum_{j \in I_s \setminus \{i\}} e^{-(\frac{d_{ij}}{d_c})^2} \quad (8)$$

Which is satisfied with more data points of the distance with i smaller than d_c and the bigger ρ_i . δ_i is measured by computing the minimum distance between point i and any other points with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i}(d_{ij}) \quad (9)$$

The motion capture data frames which are satisfied the conditions of cluster centers should have the larger ρ and δ. Computing $\alpha = \rho * \delta$ and choosing the larger α , which are more likely to become cluster centers. Because of the magnitude of ρ and δ is uniform, it needs to be normalized.

In order to extract cluster centers, we observe the discrete derivative $\Delta_i = |\alpha_i - \alpha_{i-\theta}|$, where θ must be enough to avoid noise in the data. For$\{\alpha_i\}_{i=1}^{N}$, we calculate the average $\overline{\Delta}_i$ and standard derivation ε_i of all data Δ_i . So we extract the point as cluster center when $\Delta_i > 3\varepsilon_i$. we automatically extract cluster numbers k and cluster centers $\{c_1, c_2, c_3, \ldots, c_k\}$ for motion capture data sequences. In this way, we extract k as the number of behavior for motion capture data sequence.

V. BEHAVIORAL SEGMENTATION BASED ON SPECTRAL CLUSTERING

Yuan et al.[20] proposed a graph partition model with temporal constraints to perform temporal segmentation problem. As we all know, the problem of motion capture data segmentation is typical temporal data segmentation. So we base on graph partition theory to segment motion capture data. Giving a N-Frame motion capture data sequence $\{f_1, f_2, f_3, \ldots, f_N\}$, we construct an undirected, weighted graph G=G(V,E). Each node in the set V describes a frame and frames is connected by edges. The similarity between frames represents the weight of each edge. This problem for Segmenting motion capture data can be restated as follows: we want to find a partition of the graph such that the edges between different sub-graphs have a very low weight (which means that points in different clusters are dissimilar from each other) and the edges within the same group have high weight (which means that points within the same cluster are similar to each other).

A. *Building Graph for Motion Capture Data*

For the N-frame motion capture data sequence, we construct G(V,E) as an undirected graph with vertex set $V = \{f_1, f_2, f_3, \ldots, f_N\}$. In the following we assume that the graph G is weighted. Each edge between two vertices f_i and f_j carries a non-negative weight $w_{ij} \geq 0$. The matrix $W = (w_{ij})_{i,j=1,2,3,\ldots,N}$ represents the weighted adjacency matrix of the graph. When $w_{ij} = 0$ means that the vertices f_i and f_j are not connected by an edge. Because of G is an undirected graph, we define $w_{ij} = w_{ji}$. The degree of a vertex $f_i \in V$ is defined as:

$$d_i = \sum_{j=1}^{N} w_{ij} \tag{10}$$

We construct the degree matrix D, which is defined as the diagonal matrix with the degrees $d_1, d_2, d_3, \ldots, d_N$ on the diagonal. Defining an indicator vector $I = (I_1, I_2, I_3, \ldots, I_N)^{'} \in R^N$ as the vector with entries $I_i = 1$ if $I_i \in F$ (F means subset of vertices $F \subset V$) and $I_i = 0$ otherwise.

For two disjoint sets $F_1, F_2 \subset V$, we define:

$$W(F_1, F_2) = \sum_{i \in F_1, j \in F_2} w_{ij} \tag{11}$$

Considering two different ways of measuring the size of the subset $F_c \subset V$.

$< 1 > |F_c|$ = the number of vertices in F_c;

$< 2 > vol(F_c) = \sum_{i \in F_c} d_i$.

Intuitively, $|F_c|$ measures the size of F_c by its number of vertices, while $vol(F_c)$ measures the size of F_c by summing over the weights of all edges attached to vertices in F_c. The nonempty sets $F_1, F_2, F_3, \ldots, F_k$ form a partition of the graph if $F_{ci} \cap F_{cj} = \emptyset$ and $F_1 \cup \ldots \cup F_k = V$.

B. *Construt Graph Laplacians*

For the undirected graph G(V,E), we define its Laplacian matrix:

$$L = D - W \tag{12}$$

The Laplacian matrix satisfies the following properties:

$< 1 >$ L has m non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_m$;

$< 2 >$ For every vector $I \in R^N$, we have $I^{'} L I = \frac{1}{2} \sum_{i,j=1}^{N} w_{ij}(I_i - I_j)^2$.

The proof of it as follows:

$$\begin{aligned} I^{'} L I &= I^{'}(D - W)I = \sum_{i=1}^{N} d_i I_i^2 - \sum_{i,j=1}^{N} I_i I_j w_{ij} \\ &= \frac{1}{2}(\sum_{i=1}^{N} d_i I_i^2 - 2 \sum_{i,j=1}^{N} I_i I_j w_{ij} + \sum_{j=1}^{N} d_j I_j^2) \\ &= \frac{1}{2} \sum_{i,j=1}^{N} w_{ij}(I_i - I_j)^2 \end{aligned} \tag{13}$$

For N-Frame motion capture data sequence $\{f_1, f_2, \ldots, f_N\}$, we consider every frame as data point for the undirected graph. It is generally known that several popular constructions to transform data points with pairwise similarities w_{ij} or pairwise distances d_{ij} into a graph. When constructing similarity graphs the goal is to model the local neighborhood relationships between data points.

Considering two different ways of constructing similarity graph:

$< 1 >$ t-nearest neighbor graphs: The goal is to connect vertex F_i with vertex F_j if F_j is among the t-nearest neighbors of F_j . We set the t that the average number of neighbors is around 2% of the total number for every motion capture data sequences.

$< 2 >$ The fully connected graph: we use the Gaussian similarity function, which is defined in (4). The parameter σ controls the width of the neighborhoods.

C. *Apply Spectral Clustering to Graph Cut*

a. *Relationship Between Graph Cut and Spectral Clustering*

For this undirected and weighted graph, the mincut approach simply consists of choosing a partition $F_1, F_2, F_3, \ldots, F_k$ which minimizes:

$$cut(F_1, F_2, F_3, \ldots, F_k) = \frac{1}{2} \sum_{i=1}^{k} W(F_i, \bar{F}_i) \tag{14}$$

Where $\bar{F}$ for the complement of F. Introducing the factor 1/2 for notational consistency, otherwise we would count each edge twice in the cut. In order to make the weight of cuts edges, which are minimum, it's to make the above objective function minimum.

There are two most common objective functions to encode this are RatioCut, which is defined by Hagen et al.[21], and the nor-

malized cut (Ncut), which is proposed by Shi et al[22].

$$\begin{aligned} RatioCut(F_1,\ldots,F_k) &= \frac{1}{2}\sum_{i=1}^{k}\frac{W(F_i,\bar{F}_i)}{|F_i|} \\ &= \sum_{i=1}^{k}\frac{cut(F_i,\bar{F}_i)}{|F_i|} \end{aligned} \tag{15}$$

$$\begin{aligned} Ncut(F_1,\ldots,F_k) &= \frac{1}{2}\sum_{i=1}^{k}\frac{W(F_i,\bar{F}_i)}{vol(F_i)} \\ &= \sum_{i=1}^{k}\frac{cut(F_i,\bar{F}_i)}{vol(F_i)} \end{aligned} \tag{16}$$

Choosing a motion capture data sequence,which has two behavior, and we analyze the relationship between graph cut and spectral clustering.

The goal of us is to solve the optimization problem:

$$\min_{F\subset V} RatioCut(F,\bar{F}) \tag{17}$$

Defining the vector $I=(I_1,I_2,I_3,\ldots,I_N)^{'}\in R^N$ and getting

$$I_i=\begin{cases}\sqrt{|\bar{F}|/|F|} & if \quad f_i\in F\\ -\sqrt{|F|/|\bar{F}|} & if \quad f_i\in\bar{F}\end{cases} \tag{18}$$

According to the property of Laplacian matrix, which is $I^{'}LI=\frac{1}{2}\sum_{i,j=1}^{N}w_{ij}(I_i-I_j)^2$, we can get:

$$\begin{aligned} I^{'}LI &= \frac{1}{2}\sum_{i,j=1}^{N}w_{ij}(I_i-I_j)^2 \\ &= \frac{1}{2}\Big(\sum_{i\in F,j\in\bar{F}}w_{ij}\Big(\sqrt{\frac{|\bar{F}|}{|F|}}+\sqrt{\frac{|F|}{|\bar{F}|}}\Big)^2 \\ &+ \sum_{i\in\bar{F},j\in F}w_{ij}\Big(-\sqrt{\frac{|\bar{F}|}{|F|}}-\sqrt{\frac{|F|}{|\bar{F}|}}\Big)^2\Big) \\ &= cut(F,\bar{F})\big(\frac{|F|+|\bar{F}|}{|F|}+\frac{|F|+|\bar{F}|}{|\bar{F}|}\big) \\ &= |V|*RatioCut(F,\bar{F}) \end{aligned} \tag{19}$$

In other words, we can get that the optimization problem and the Laplacian matrix has a big relationship. It's easy to extend to k subgraphs.

Because of these constraint conditions:

$$\begin{aligned} \sum_{i=1}^{N}I_i &= \sum_{i\in F}\sqrt{\frac{|\bar{F}|}{|F|}}-\sum_{i\in\bar{F}}\sqrt{\frac{|F|}{|\bar{F}|}} \\ &= |F|*\sqrt{\frac{|\bar{F}|}{|F|}}-|\bar{F}|*\sqrt{\frac{|F|}{|\bar{F}|}}=0 \end{aligned} \tag{20}$$

$$I^{'}*\mathbf{1}=\sum_{i=1}^{N}I_i=0 \tag{21}$$

$$||I||^2=\sum_{i=1}^{N}I_i^2=|F|*\sqrt{\frac{|\bar{F}|}{|F|}}+|\bar{F}|*\sqrt{\frac{|F|}{|\bar{F}|}}=N \tag{22}$$

We can get the new optimization problem:

$$\begin{aligned} &\min_{I\in R^N} I^{'}LI \\ &subject \quad to \quad I^{'}*\mathbf{1}=0, \quad ||I||=\sqrt{N} \end{aligned} \tag{23}$$

Assuming that $LI=\lambda I$, at the moment, λ is eigenvalues and I is $L^{'}s$ eigenvectors. Multiplying from the left and the right by $I^{'}$ for $LI=\lambda I$, we can get $I^{'}LI=\lambda I^{'}I$ (where $I^{'}I=N$). Because of N is constant, minimizing the formula $I^{'}LI$ can be replaced by minimizing λ. For the Normalized cut, we define the cluster indicator vector $I=(I_1,I_2,I_3,\ldots,I_N)^{'}\in R^N$ by:

$$I_i=\begin{cases}\sqrt{vol(\bar{F})/vol(F)} & if \quad f_i\in F\\ -\sqrt{vol(F)/vol(\bar{F})} & if \quad f_i\in\bar{F}\end{cases} \tag{24}$$

Likely the RatioCut, we can get:

$$\begin{aligned} I^{'}LI &= \frac{1}{2}\sum_{i,j=1}^{N}w_{ij}(I_i-I_j)^2 \\ &= \frac{1}{2}\Big(\sum_{i\in F,j\in\bar{F}}w_{ij}\Big(\sqrt{\frac{vol(\bar{F})}{vol(F)}}+\sqrt{\frac{vol(F)}{vol(\bar{F})}}\Big)^2 \\ &+ \sum_{i\in\bar{F},j\in F}w_{ij}\Big(-\sqrt{\frac{vol(\bar{F})}{vol(F)}}-\sqrt{\frac{vol(F)}{vol(\bar{F})}}\Big)^2\Big) \\ &= cut(F,\bar{F})\big(\frac{vol(F)+vol(\bar{F})}{vol(F)}+\frac{vol(F)+vol(\bar{F})}{vol(\bar{F})}\big) \\ &= vol(V)*NCut(F,\bar{F}) \end{aligned} \tag{25}$$

$vol(V)$ is a constant. The goal of us is to solve the optimization problem:

$$\min_{F\subset V} NCut(F,\bar{F}) \tag{26}$$

It's equal to $\min_{I\in R^N} I^{'}LI$. In order to deal with this optimization problem, we calculate the Laplacian matrixs minimum eigenvalue. However, the smallest eigenvalue of L is 0 with eigenvector 1. The Rayleigh-Ritz theorem can be used to solve this problem. The solution of this problem is given by the vector I which is the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix L . So we can approximate a minimizer of RatioCut or Ncut by the second eigenvector of L . Extending to k clusters, we compute the eigenvalues of the Laplacian matrix and sort them according to order from small to big. Eigenvectors corresponding to eigenvalues are also sort in increasing. Extracting k eigenvectors as what we want. In this way, we succeeded in converting the graph cut, which is a NP problem, into the Laplacian matrix eigenvalues (eigenvectors) problem.

b. Apply Nystrom Method to Spectral Clustering

The $Nyström$ method[23], which uses a sub-matrix of the dense similarity matrix: This method is a technique for finding an

approximate eigendecomposition. Here, we denote by W , which is a N*N similarity matrix. Assume that we randomly select sample $l << N$ points from the data. The matrix A represent the $l \times l$ matrix of similarities between the same points, B be the $l \times (n-l)$ matrix of affinities between the l sample points and the $(n-l)$ remaining points, C contains the similarities between all $(n-l)$ remaining points and O be the $(n \times l)$ matrix consisting of A and B^T. We can get rearrange the similarity matrix W such that:

$$W = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \quad and \quad O = \begin{bmatrix} A \\ B^T \end{bmatrix} \tag{27}$$

For the $Nystr\ddot{o}m$ method, it use matrix A and B to approximate similarity matrix W.

$$\begin{aligned} W &\approx \tilde{W} = OA^{-1}O^T \\ &= \begin{bmatrix} A & B \\ B^T & B^TA^{-1}B \end{bmatrix} = \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1} \begin{bmatrix} A & B \end{bmatrix} \end{aligned} \tag{28}$$

Where the matrix C is now replaced by $B^TA^{-1}B$. The matrix A makes eigendecomposition and we can get $A = U_A\Sigma_AU_A^T$, where Σ_A contains the eigenvalues of A and U_A are the corresponding eigenvectors. According to the $Nystr\ddot{o}m$ method, we can get :

$$\tilde{\Sigma} = (\frac{N}{l})\Sigma_A, \tilde{U} = \sqrt{\frac{l}{N}}OU_A\Sigma_A^{-1} \tag{29}$$

Moreover, the similarity matrix W has the eigendecomposition:

$$\tilde{W} = \tilde{U}\tilde{\Sigma}\tilde{U}^T \tag{30}$$

For the Laplacian matrix L , we normalize it by:

$$\begin{aligned} L &= D - W \\ &= D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \end{aligned} \tag{31}$$

Where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{N} W_{ij}$.

Computing the rows sums of $\tilde{W}$:

$$\begin{aligned} \tilde{W} * 1 &= \begin{bmatrix} A & B \\ B^T & B^TA^{-1}B \end{bmatrix} * \begin{bmatrix} 1_l \\ 1_{N-l} \end{bmatrix} \\ &= \begin{bmatrix} A * 1_l + B * 1_{N-l} \\ B^T * 1_l + B^TA^{-1}B * 1_{N-l} \end{bmatrix} \\ &= \begin{bmatrix} a_l + b_l \\ b_{N-l} + B^TA^{-1}b_l \end{bmatrix} \end{aligned} \tag{32}$$

Where a_l, b_l represents the row sums of A and B, b_{N-l} denotes the column sum of B and $\vec{1}$ means a column vector of ones. According to this, we can get:

$$\begin{aligned} D^{-\frac{1}{2}}\tilde{W}D^{-\frac{1}{2}} &= D^{-\frac{1}{2}} \begin{bmatrix} A & B \\ B^T & B^TA^{-1}B \end{bmatrix} D^{-\frac{1}{2}} \\ &= \begin{bmatrix} D_1^{-\frac{1}{2}}AD_1^{-\frac{1}{2}} & D_1^{-\frac{1}{2}}BD_2^{-\frac{1}{2}} \\ D_2^{-\frac{1}{2}}B^TD_1^{-\frac{1}{2}} & D_2^{-\frac{1}{2}}B^TA^{-1}BD_2^{-\frac{1}{2}} \end{bmatrix} \end{aligned} \tag{33}$$

where D_1 is $l * l$ and D_2 is $(N-l) * (N-l)$. For the similarity matrix W, we want to show it can be diagonalized. Supposing that $U = \begin{bmatrix} A & B^T \end{bmatrix}^T A^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}}$, we can get:

$$\begin{aligned} W &= \begin{bmatrix} A & B^T \end{bmatrix}^T A^{-1} \begin{bmatrix} A & B \end{bmatrix} \\ &= \{\begin{bmatrix} A & B^T \end{bmatrix}^T A^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}}\}\Sigma\{\Sigma^{-\frac{1}{2}}v^TA^{-\frac{1}{2}}[A \quad B]\} \\ &= U\Sigma U^T \end{aligned} \tag{34}$$

Multiplying from the left by $V\Sigma^{\frac{1}{2}}$ and from the right by $\Sigma^{\frac{1}{2}}V^T$ for the unitary matrix, we can get:

$$\begin{aligned} & V\Sigma V^T \\ &= V\Sigma^{\frac{1}{2}}(U^TU)\Sigma^{\frac{1}{2}}V^T \\ &= V\Sigma^{\frac{1}{2}}(\begin{bmatrix} A \\ B^T \end{bmatrix} A^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}})^T * \\ &(\begin{bmatrix} A \\ B^T \end{bmatrix} A^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}})\Sigma^{\frac{1}{2}}V^T \\ &= V\Sigma^{\frac{1}{2}}(\Sigma^{-\frac{1}{2}}V^TA^{-\frac{1}{2}}[A \quad B])* \\ &(\begin{bmatrix} A \\ B^T \end{bmatrix} A^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}})\Sigma^{\frac{1}{2}}V^T \\ &= A^{-\frac{1}{2}}[A \quad B] \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-\frac{1}{2}} \\ &= A + A^{-\frac{1}{2}}BB^TA^{-\frac{1}{2}} \end{aligned} \tag{35}$$

Because of this property, we reduce the computational complexity and we require only the first k eigenvectors of the Laplacian matrix. Calculating the first k columns of U via

$$U = \begin{bmatrix} \tilde{A} \\ \tilde{B}^T \end{bmatrix} \tilde{A}^{-\frac{1}{2}}(V_N)_{:,1:k}(\Sigma^{-\frac{1}{2}})_{1:k,1:k} \tag{36}$$

Then we normalize U along its rows to get $\tilde{U}$.

$$\tilde{U}_{ij} = \frac{U_{ij}}{\sqrt{\sum_{r=1}^{k} U_{ir}^2}}, i = 1, 2, \ldots, N, j = 1, 2, \ldots, k \tag{37}$$

c. *K-means Step for Normalized Matrix* $\tilde{U}$

Defining $\{u\}_{j=1}^N$ is the vectors corresponding to $\tilde{U}'s$ rows. we have extracted cluster numbers k and cluster centers $\{c_1, c_2, c_3, \ldots, c_k\}$ by the method, as introduced in the Section 4. Through the corresponding relationship between normalized matrix $\tilde{U}$ and the clusters centers, we update the cluster centers, when they correspond to same frames. K-means[24] algorithm aims at minimizing the objective function know as squared error function given by:

$$J(\tilde{C}) = \sum_{i=1}^{k} \sum_{\tilde{u}_j \in C_i} ||u_j - \tilde{c}_i||^2 \tag{38}$$

Where $'C_i'$ is the number of data points in i^{th} cluster. According to assign the points to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
Algorithmic steps for k-means: Let $\tilde{U} = \{u_1, u_2, \ldots, u_N\}$ be the elements to be clustered and $\tilde{C} = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_k\}$ be the sets of corresponding centers:

1) Getting k cluster centers $\tilde{C} = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_k\}$;

2) Calculate the distance between each element to be clustered and cluster centers;

3) Assign each element to be clustered to the cluster center whose distance from the cluster center is minimum of all the cluster centers;

4) Recalculate the new cluster center using: $\tilde{c}_i = \frac{1}{C_i} \sum_{j=1}^{C_i} u_i$, Where $'C_i'$ is the number of data points in ith cluster.

5) Recalculate the distance between each element to be clustered and new obtained cluster centers;

6) Stop the iteration until no elements was reassigned, otherwise repeat to step 3).

According to this step and analysis, we can get the N-Frame motion capture data sequences categories information.

D. Behavioral Segmentation Based On Spectral Clustering

We have introduced the method of t-nearest neighbor graphs in Section 5.B. According to this, it can be easily to find that this method is useful to reduce the store of dense similarity matrix. In this way, the dense similarity matrix is represented by the sparse similarity matrix. The algorithm of cluster, which uses the sparse similarity matrix, as follows:

Algorithm 1: clustering uses the sparse similarity matrix.

Input: N-Frame motion capture data $\{f_1, f_2, f_3, \ldots, f_N\}$, every frame is regarded as high-dimensional data point and k: number of desired clusters.

1) Construct similarity matrix $W \in R^{N \times N}$;

2) Modify W to be the sparse matrix S;

3) Compute the Laplcian matrix by $L = I - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$;

4) Compute the K eigenvectors of L and construct $U \in R^{N \times k}$;

5) Normalize U along its rows to get $\tilde{U}$. $\tilde{U}_{ij} = \frac{U_{ij}}{\sqrt{\sum_{r=1}^{k} U_{ir}^2}}, i = 1, 2, \ldots, N, j = 1, 2, \ldots, k$

6) Use k-means algorithm to cluster N rows of $\tilde{U}$ into k groups, which mean action segments for motion capture data sequences.

In subsection 5.C-b , introducing the Nystrom method, which can be used to cluster for motion capture data sequence.

Algorithm 2: Clustering uses the Nystrom method.

Input: N-Frame motion capture data $\{f_1, f_2, f_3, \ldots, f_N\}$, every frame is regarded as high-dimensional data point; number of samples l, which is determined by 2% the number of motion capture data, k: number of desired clusters and $\{c_1, c_2, c_3, \ldots, c_k\}$: cluster centers, which are determined by the algorithm.

1) Construct $A \in R^{l \times l}$ and $B \in R^{l \times (N-l)}$. The matrix A represent the matrix of similarities between the same points, B be the matrix of affinities between the sample point.

2) Calculate $D = diag(\begin{bmatrix} a_l + b_l \\ b_{(N-l)} + B^T A^{-1} b_l \end{bmatrix})$.

3) Calculate $\tilde{A} = D_{l \times l}^{-\frac{1}{2}} A D_{l \times l}^{-\frac{1}{2}}, \tilde{B} = D_{l \times l}^{-\frac{1}{2}} B D_{(N-l) \times (N-l)}^{-\frac{1}{2}}$.

4) Construct $A + A^{-\frac{1}{2}} B B^T A^{-\frac{1}{2}}$. calculate eigendecomposition $V_N \Sigma_N V_N^T$ for it and ensure the eigenvalues are in decreasing order.

5) Calculate $U = \begin{bmatrix} \tilde{A} \\ \tilde{B}^T \end{bmatrix} \tilde{A}^{-\frac{1}{2}} (V_N)_{:,1:k} (\Sigma^{-\frac{1}{2}})_{1:k,1:k}$ as the first k eigenvector of the Laplacian matrix.

6) Normalize U along its rows to get $\tilde{U}$.$\tilde{U}_{ij} = \frac{U_{ij}}{\sqrt{\sum_{r=1}^{k} U_{ir}^2}}, i = 1, 2, \ldots, N, j = 1, 2, \ldots, k$

7) Use k-means to cluster N rows of $\tilde{U}$ into k classes.

Two methods of these can realize clustering motion capture data sequence into motion fragments.

VI. REFINE BEHAVIORIAL SEGMENTATION RESULTS

According to temporal reverting, we can get $S = \{s_1, s_2, s_3, \ldots, s_{k'}\}$ for N-Frame motion capture data sequence, where s_i represents the subsequence after clustering and temporal reverting. Setting the single representation $G = \{g_{c_i}\}_{c \in \{c_1, c_2, \ldots, c_k\}, i=1,2,\ldots,N}$. g_{c_i}=1 if f_i belongs to class c, otherwise g_{c_i}=0.

According to extensive experiments, we can find that clustering and temporal reverting for motion capture data can lead some error. We define energy function to reduce noise and realize behavior segmentation. Because of the temporal property, behavior segmentation points locate in adjacent motion subsequences. For each motion subsequence, we only calculate it for belonging to the former one or the last one. The energy function is defined by:

$$F(S, G) = \sum_{i,j \in 1,2,\ldots,k'} \sum_{c_i=1}^{k} g_{c_i} dist(s_i, s_j) \tag{39}$$

Where s_i represents motion subsequences for motion capture data after clustering and temporal reverting. Pavel Senin [25] provided the good resolution for calculating the distance for temporal sequences, which is named dynamic time warp (DTW).

$$\begin{aligned} DTW(s_i, s_j) &= dist(s_i, s_j) \\ &= \arg\min\{ \sum_{f_{i'} \in s_i, f_{j'} \in s_j} ||f_{i'} - f_{j'}|| \} \end{aligned} \tag{40}$$

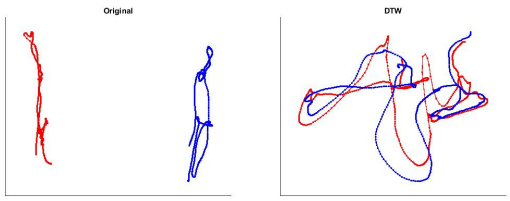
Dynamic programming is applied to deal with behavior segmentation. The formula is defined as follows:

$$J(S^{'}) = \min\{J(S^{'} - 1) + min\{F(S^{'} - 1, G)\}\} \tag{41}$$

In this way, we can realize behavior segmentation. After clustering, long action subsequences can be considered as independent behavior and we use the energy function to deal with error subsequences. Where s_i represents action subsequences for motion capture data after clustering and temporal reverting. The Figure 3 shows an example of two motion capture data sequences aligned by DTW.

VII. EXPERIMENTS

In order to prove the feasibility and effectiveness of this method, we use the Carnegie Mellon University motion capture database [CMU][26], whose data were captured with a Vicon optical motion capture system of 12 MX-40 cameras at 120HZ. Extensive experiments are conducted on the dataset of multi-behavior



(a) (b)

Figure 3: The result of two motion capture data sequences aligned by DTW, which is demonstrated in 2-dimensional space. (a) represents original sequences and (b) shows aligned result.

motion capture data from CMU database. each motion capture data sequence is the combination of roughly several natural actions (walking, running, punching, jumping and so on.).

The greatest advantage of our method for behavior segmentation is automatic and without human intervention. For arbitrary motion capture data, which contains several behaviors, we can realize behavior segmentation. Figure 4 shows the segmentation results obtained through PPCA, GMM, our methods (t-nearest and Nystrom method) and manual segmentation. In every chart, x-axis represents the motion frames and y-axis defines methods abbreviation. Each chart contains five small bars. From the top downwards, each bar represents the results of behavior segmentation by PPCA, GMM, t-nearest neighbors to cluster, Nystrom method applied to cluster and manual segmentation. The first four methods use black vertical line to represent the results of segmentation.

For the manual segmentation, it use black strip to represent segmentation results. Since the continuity of behaviors and the restriction of human recognition, the human segmentation labels are not single frame and they are a short sequence of frames. Manual segmentation results are regarded as ground truth for segmentation of motion capture data. Due to the continuity of motion, we think that segmentation results located in the vicinity of ground truth are segmentation successful.

According to experiments, we can find that our method not only can segment motion capture data, but also can mark the same behavior segments. We use same color to mark the same behavior segments. The experiments results are close to ground truth. That as long as the segmentation results falling in this sequence are the right points of segmentation. We compare these behavior segmentation algorithms in the standard precision/recall framework.

$$\begin{cases} precision = \frac{\#reportedcorcuts}{\#reportedcuts} \times 100\% \\ recall = \frac{\#reportedcorcuts}{\#corcuts} \times 100\% \end{cases} \quad (42)$$

where #reportedcorcuts indicates the reported correct cuts, #reportedcuts is the total number of reported cuts and #corcuts represents the total correct cuts. The closer precision and recall are to one, the more effective the algorithm is. Table 2 gives scores of precision and recall for PPCA, GMM, t-nearest and Nystrom method.

Table 2: Precision and recall scores for the PPCA,GMM,t-nearest and Nystrom method

Behavior Methods	Precision	Recall
PPCA	80.83%	87.38%
GMM	54.00%	72.97%
t-nearest	90.09%	90.91%
Nystrom	92.79%	93.63%

VIII. DISCUSSION AND CONCLUSION

In this paper, we introduce a novel automatically method to segment motion capture data. Firstly, we use a novel method to extract cluster numbers k, which can be regarded as the number of behavior contained, and cluster centers. This cluster algorithm is based on local density for input data. Then, we construct undirected weighted graph, which uses motion capture data. In order to deal with this NP hard problem, we analyze relationship between graph cut and spectral clustering. According to this, the problem of behavior segmentation for arbitrary motion capture data sequence is converted to spectral clustering problem. According to the priori knowledge of behavior numbers and cluster centers, the application of t-nearest neighbors and Nystrom method used respectively to cluster for motion capture data sequence. Defining energy function to refine segmentation for motion capture data. These method can reduce computational complexity, save restore space and improve calculation speed. Energy function, which is defined by us, can reduce the error of segmentation results.

Although we have reduce the restore space for this method, it takes up remarkable restore space. In future work, we should further improve the calculate speed and reduce restore space. Meanwhile, to further study the effect of single behavior fragments on the motion analysis and motion synthesis.

ACKNOWLEDGMENTS

This research is partially supported by National Natural Science Foundation of China (No. 61370127, No.61473031, No.61472030), Program for New Century Excellent Talents in University (NCET-13-0659), Fundamental Research Funds for the Central Universities (2014JBZ004), Beijing Higher Education Young Elite Teacher Project (YETP0583). The opinions expressed are solely those of the authors and not the sponsors.

References

[1] Paulo Sousa, João L Oliveira, Luis Paulo Reis, and Fabien Gouyon. Humanized robot dancing: humanoid motion retargeting based in a metrical representation of human dance styles. In *Progress in Artificial Intelligence*, pages 392–406. Springer, 2011.

[2] Jianyuan Min and Jinxiang Chai. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics (TOG)*, 31(6):153, 2012.

[3] Adso Fern'ndez-Baena, Antonio Susin, and Xavier Lligadas. Biomechanical validation of upper-body and lower-body

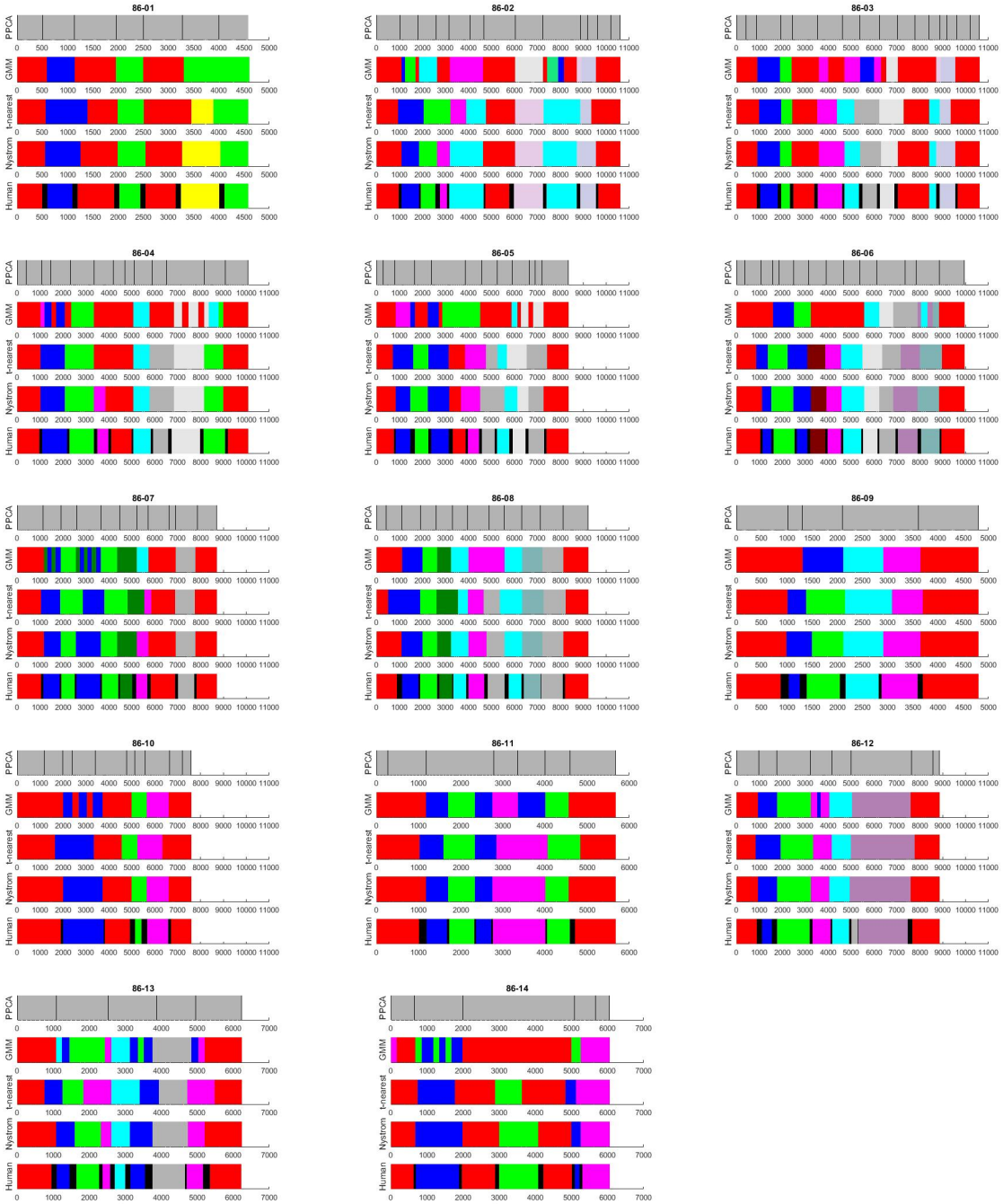


Figure 4: The results of segmentation. It shows the segmentation results obtained through PPCA, GMM, t-nearest neighbors to cluster, Nystrom method applied to cluster and manual segmentation. Different behaviors are indicated by different colors. The five colors stripes in the same chart represent the same motion capture data sequence.

joint movements of kinect motion capture data for rehabilitation treatments. In *Intelligent Networking and Collaborative Systems (INCoS), 2012 4th International Conference on*, pages 656–661. IEEE, 2012.

[4] Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6):2566–2570, 2011.

[5] Ilaria Pasciuto, Sergio Ausejo, Juan Tomás Celigüeta, Ángel Suescun, and Aitor Cazón. A hybrid dynamic motion prediction method for multibody digital human models based on a motion database and motion knowledge. *Multibody System Dynamics*, 32(1):27–53, 2014.

[6] Paul Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and computing*, 16(2):203–213, 2006.

[7] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296. IEEE, 2001.

[8] Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*, pages 1055–1062. ACM, 2007.

[9] Meinard Müller, Andreas Baak, and Hans-Peter Seidel. Efficient and robust annotation of motion capture data. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 17–26. ACM, 2009.

[10] Fengjun Lv and Ramakant Nevatia. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *Computer Vision–ECCV 2006*, pages 359–372. Springer, 2006.

[11] Okan Arikan, David A Forsyth, and James F O'Brien. Motion synthesis from annotations. *ACM Transactions on Graphics (TOG)*, 22(3):402–408, 2003.

[12] Feng Zhou, F Torre, and Jessica K Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–7. IEEE, 2008.

[13] Feng Zhou, F Torre, and Jessica K Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):582–596, 2013.

[14] Michal Balazia, Jan Sedmidubsky, and Pavel Zezula. Semantically consistent human motion segmentation. In *Database and Expert Systems Applications*, pages 423–437. Springer, 2014.

[15] J. Barbič, A. Safonova, JY. Pan, C. Faloutsos, JK. Hodgins, and NS. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface 2004*, pages 185–194. Canadian Human-Computer Communications Society, 2004.

[16] Jehee. Lee, Jinxiang. Chai, Paul S.A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 491–500. ACM, 2002.

[17] Jing Wang and Bobby Bodenheimer. An evaluation of a cost metric for selecting transitions between motion segments. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 232–238. Eurographics Association, 2003.

[18] J Macqueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1967*.

[19] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[20] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang. A formal study of shot boundary detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(2):168–186, 2007.

[21] Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-aided design of integrated circuits and systems, ieee transactions on*, 11(9):1074–1085, 1992.

[22] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[23] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y Chang. Parallel spectral clustering in distributed systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):568–586, 2011.

[24] Tapas Kanungo, David M Mount, Nathan S Netanyahu, and Angela Y Piatko. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[25] Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, pages 1–23, 2008.

[26] CMU. Carnegie mellon university graphics lab: Motion capture database. `http://motioncapture.cs.cmu.edu`.