Multimed Tools Appl (2014) 72:951–966
DOI 10.1007/s11042-013-1416-x

Motion retrieval based on Switching Kalman Filters Model

Qinkun Xiao · Yichuang Luo · Haiyun Wang

Published online: 10 March 2013

Abstract A novel content-based motion descriptor is proposed. Firstly, the multi-view image information is captured to represent motion, and then the Switching Kalman Filters Model (S-KFM), which is a kind of the Dynamic Bayesian Network (DBN), is built based on the images fusion and the optical stream technology. Secondly, through the S-KFM inferring and sequence signal coding, a graph-based motion descriptor can be obtained. Lastly, motion matching results based on the graph model descriptor show our method is effective.

Keywords Motion retrieval · Multi-view · Graph model · S-KFM

1 Introduction

In recent years, computer animation has become very popular due to the increasing importance of it in many applications [20, 24, 25]. In computer animation, we are particularly interested in human motion. There are many methods developed to produce the human motion data. A well-known method is called the motion capture (MoCap). With the motion capture devices becoming more widely available, large motion databases start to appear [5–7, 15, 18]. However, as the number of motions grows, it becomes difficult to select an appropriate motion that satisfies certain requirements. Hence, motion retrieval has become one of the major research focuses in motion capture animation in recent years.

Motion retrieval research is still relatively new compared to retrieval research of other multimedia data. There are only a few motion retrieval methods in the literature. Many motion retrieval systems use the Dynamic Time Warping (DTW) as the similarity measure [9]. However, the DTW usually has low efficiency due to motion capture data consists of many

Q. Xiao (✉) · Y. Luo
Department of Electronics Information Engineering, Xi'an Technological University, Xi'an 710032, China
e-mail: xiaoqinkun10000@163.com

Y. Luo
e-mail: 623634511@qq.com

H. Wang
STMicroelectronics R&D of Asia-Pacific, Singapore 554574, Singapore
e-mail: haiyun_w@gmail.com

parameters and attributes. For increasing the DTW-based retrieval efficiency, dimension reduction methods are often employed [2]. In order for the DTW to support indexing and further increasing retrieval efficiency, [8] proposes an algorithm which is based on the Uniform Scaling to match the query. However, in order to handle the motions that contain both local and global differences, the computational cost of system is increased significantly when the DTW and the Uniform Scaling should be applied separately.

Besides the DTW-based methods, other works concern finding logically similar motions. For example, in [11], templates are created to describe motion, retrieval is based on the template matching. In [10], geometric features are used to build indexing tree automatically based on segmentation and clustering, and motion matching is based on peak points. In [14], a motion index tree is constructed based on a hierarchical motion description. The motion index tree serves as a classifier to determine the sub-library that contains the promising similar motions to the query sample. The Nearest Neighbor rule-based dynamic clustering algorithm is adopted to partition the library and construct the motion index tree. The similarity between the sample and the motion in the sub-library is calculated through elastic match.

Some works related to motion sequence analysis and estimation are also done in recently years [1, 4, 19, 22, 23], which are basis for finding more effective motion retrieval approaches. For example, in [4], a motion-compensated deinterlacing scheme based on hierarchical motion analysis is presented. For motion estimation, a Gaussian noise model for choosing the best motion vector for each block is introduced. In [23], a general framework to unsupervisedly discover video shot categories is studied. A new feature is proposed to capture local information in videos. In [1], a motion trajectory-based compact indexing and efficient retrieval mechanism for video sequences is proposed. This approach solves the problem of trajectory representation when only partial trajectory information is available due to occlusion. It is achieved by a hypothesis testing-based method applied to curvature data computed from trajectories. In [19], a robust logical relevance metric based on the relative distances among the joints is discussed. The [22] studies an adaptive tracking algorithm by learning hybrid object templates online in video. The templates consist of multiple types of features, each of which describes one specific appearance structure, such as flatness, texture or edge/corner.

In this paper, we are interested in finding motions that are entirely similar to a given query. Based on multi-view information and image fusion technology, we convert motion matching into a transportation problem to handle rotating, local scaling or global scaling. Based on graph model inference and sequence information coding, we can compute distance between two motions. We compare mainly with the DTW and the Uniform Scaling method. Though we have not implemented any indexing scheme, extending our method to support indexing can be easily achieved because our distance function is a metric. Our experimental results show that the proposed method is promising.

The rest of this paper is organized as follows. Overview of our method is presented in Section 2. Section 3 describes our method in detail. Section 4 evaluates the performance of our method through experiments. Section 5 briefly concludes this paper.

2 Overview of our method

Our method can be briefly described in Fig. 1. Motion retrieval frame can be separated two parts entirely: motion descriptor building and motion retrieval.

In stage of motion descriptor building, firstly, motion database would be constructed, in this paper, the CMU motion database [5] is used and some character motions, which can be discriminated each other, are selected out to building the motion database. Secondly, each

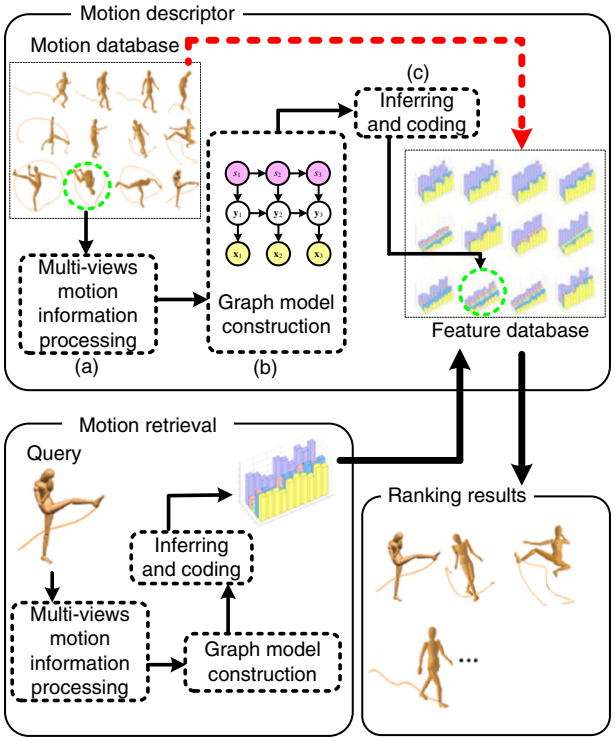


Fig. 1 Method overview

motion feature is extracted, the steps including (as shown in Fig. 1): (1) *Processing of multi-view motion information*. Each animation is put into the lightfield [21] to get multi-view images, and the PCA-based image fusion algorithm is used to get the fusion image sequence. (2) *Building motion graph model*. For each frame of motion, the multi-view information can be fused into fusion-images, optical stream signals [12] is computed based on difference between the adjacent fusion-images, and graph model is constructed based on the optical stream signals to represent the motion. (3) *Inference and coding*. Based on obtained graph model, the DBN inference algorithm [13, 16] is used to get hidden variable sequence information, and all variables in graph model can be coded [3] as the motion descriptor.

In stage of retrieval, the query motion is extracted feature according to above, and then χ^2-test is used to compute distances between query and motions in database, the retrieval results are sorted and outputted.

3 Graph-based motion descriptor

3.1 Motion descriptor building

(a) *Processing of multi-view motion information*. In order to decrease affection of rotating and scaling and represent motion fully, the multi-view images are utilized to describe animation [21]. Firstly, the animation is put into lightfield, as shown in Fig. 2, the *m* cameras are set on vertexes of polyhedron around the object. In *i*-th frame of the

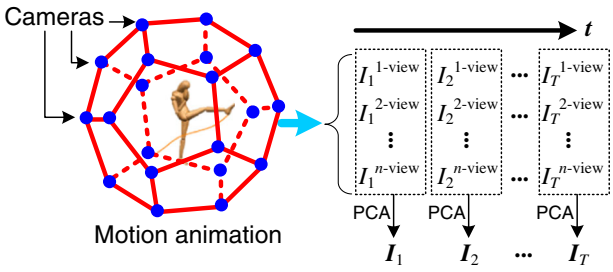


Fig. 2 Acquirement of the fusion-image sequence

motion, the images of different viewpoints are acquired around the object (denoted as: $I_i^{1\text{-view}}$, $I_i^{2\text{-view}}$,…, $I_i^{m\text{-view}}$). In order to decrease size of motion descriptor and retain useful information, image fusion algorithm based on the Principle Component Analysis (PCA) [17] is used to transform the m images into a fusion image $\boldsymbol{I}_i$. The multi-view information is got from 1st frame to T-th frame, and fusion-image sequence is obtained: $\boldsymbol{I}_1$, $\boldsymbol{I}_2$,…, $\boldsymbol{I}_T$.

The multi-view images can be fused through the pixel-based and the PCA-based method. We can further discuss the fusion method as follow.

(1) Let image $I^{i\text{-view}}$ be an $N \times N$ matrix: $I^{i\text{-view}} = [f_{ij}]_{N\times N}$, where the f_{ij} is the gray value of each pixel. The matrix $I^{i\text{-view}}$ $(1 \le i \le n)$ can also be denoted as a vector: $I^{i\text{-view}} = [f_{11}, f_{12}, \ldots, f_{NN}]^T = [f_1, f_1, \ldots, f_Q]^T$ $(Q = N^2)$, then the means and var of $I^{i\text{-view}}$ are: $\mu_f = \boldsymbol{E}[f]$, $K_f = \boldsymbol{E}[(f - \mu_f)(f - \mu_f)^T]$.

(2) A matrix $\boldsymbol{X}$ can also be constructed based on the multi-view images (denoted as $I^{1\text{-view}}$,…, $I^{m\text{-view}}$), suppose there are m images and size of each image is $n = N \times N$, we have:

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & \cdots & x_{mj} & \cdots & x_{mn} \end{pmatrix} \tag{1}$$

Where x_{ij} is gray value in j-th pixel of i-th image ($I^{i\text{-view}}$). The var matrix of $\boldsymbol{X}$ can be calculated as:

$$C = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1j} & \cdots & \sigma_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{i1} & \cdots & \sigma_{ij} & \cdots & \sigma_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{m1} & \cdots & \sigma_{mj} & \cdots & \sigma_{mn} \end{pmatrix} \tag{2}$$

Where $\sigma_{i,j}^2 = \frac{1}{n}\sum_{i=0}^{n-1}\left(x_{i,l} - \overline{x}_i\right)\left(x_{j,l} - \overline{x}_i\right)$, $\overline{x}_i$ is the average gray value of i-th image.

(3) Let $|\lambda\boldsymbol{\Psi}\text{-}\boldsymbol{C}| = 0$ ($\boldsymbol{\Psi}$ is unit matrix), and feature vector $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_m$ are obtained. We can get fusion coefficients: $\omega_i = \lambda_i / \sum_{i=1}^{m} \lambda_i$. Next, feature matrix $\boldsymbol{A}$ is calculated:

$$\left.\begin{array}{l} \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{pmatrix} \\ K_f \Lambda = A\Lambda \end{array}\right\} \Rightarrow A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \tag{3}$$

(4) Lately, the fusion image is got: $I = \sum_{i=1}^{m} \omega_i a_i$

We can give an example to further explain the fusion processing. As shown in Fig. 3. Firstly, before fusion computation, all the multi-views images should be normalized. The steps can be described as: based on center of motion object, we can find a rectangle ($a \times b$) to just enclose the motion object, and the enclosed pixels can be scale to 100×100 image, as shown in Fig. 3. Secondly, based prior knowledge, we know that images with different viewpoints have different efficiency for discriminating motions, then for different viewpoint images, we set different weights during the fusion processing. For example, as shown in Fig. 3, first according to experience, viewpoint images are selected and sorted according to discrimination efficiency, and the weight of first image is set to 1.0, the weight of secondly image can be set to 0.8, and so on. Lastly, according to the pixel-based image fusion and the PCA theory, all the multi-view images are composed together by Eqs. (1)–(3), as shown in Fig. 3.

(b) *Building motion graph model*. As we known, sequence data can be expressed by the Dynamic Bayesian Network (DBN) [16]. In this paper, we select the Switching Kalman Filters Model (S-KFM) [16], which is a kind of the DBN, to represent motion. As shown in Fig. 4, we use the DBN to build motion description, firstly, let optical

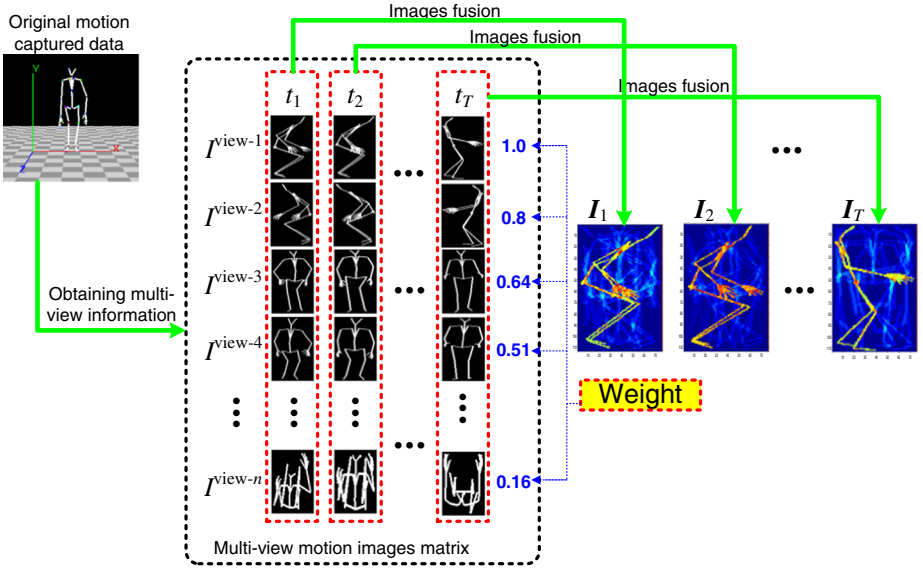


Fig. 3 An example of the multi-view images fusion

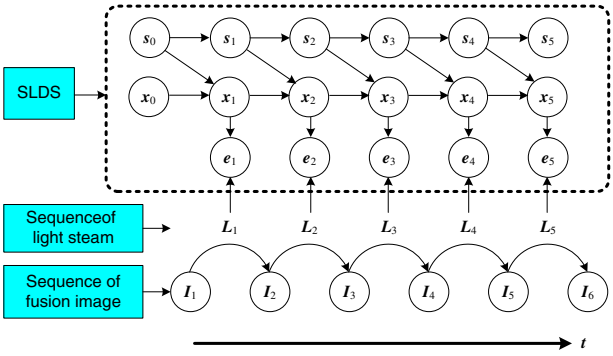


Fig. 4 The S-KFM graph model construction

stream signals be the observation signals of the DBN, the reason can be written as: in video tracking, the optical stream signals is often be used to detect moving object, the moving trend can be also described by difference between adjacent images. Based on the idea, we take the optical stream sequence of fusion-images ($\boldsymbol{I}_1$, $\boldsymbol{I}_2$,…, $\boldsymbol{I}_T$.) as observation values of the S-KFM. The optical stream sequence $\boldsymbol{L}_1$,…,$\boldsymbol{L}_5$ can be computed according to [12], we can describe in detail as follow.

Firstly, based on [12], the motion trend can be detected based on the optimal shifting vectors of corresponding points between the adjacent images. In the *i*-th frame image, let shifting value in point (x, y) be:

$$e_{x,y} = \sum_{x',y' \in W} \left(I(t+1)_{x'+\delta x, y'+\delta y} - I(t)_{x',y'} \right)^2 \tag{4}$$

where the (x',y') is any pixel point in given window W, and $\boldsymbol{I}(t+1)_{x'+\delta x,\, y'+\delta y}$ is the pixel of ($x'+\delta x$, $y'+\delta y$) at $t+1$ in fusion-image, the (δx, δy) is the shifting vector of point (x, y), the $\boldsymbol{I}(t)_{x',y'}$ is the pixel of (x', y') at t in fusion-image, a optimal solution $(\delta x,\delta y)^*$ can be found to make the $e_{x,y}$ minimize:

$$(\delta x, \delta y)^* = \arg\min_{(\delta x, \delta y)} \sum_{x',y' \in W} \left(I(t+1)_{x'+\delta x, y'+\delta y} - I(t)_{x',y'} \right)^2 \tag{5}$$

All optimal shifting vectors between adjacent fusion-images ($\boldsymbol{I}_i$ and $\boldsymbol{I}_{i+1}$) are combined together to express motion trend, that is denoted as optical stream $\boldsymbol{L}_i$. We can let the $\boldsymbol{L}_i$ be input signal of the S-KFM, which is also observation signal $\boldsymbol{e}_i$ in the DBN, then observation sequence is written as: $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_T\}$.

Secondly, given input signal $\boldsymbol{E}$, similar as noised signal processing, we can use the S-KFM to estimate hidden sequence and state switching signals (denoted as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$, $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_T\}$). Based on above analysis, the graph model of motion is constructed in Fig. 4.

(c) *Inference and coding*. The DBN Inference is to estimate the posterior probability of hidden states in system. Given observation sequence $\boldsymbol{E}$ (or called evidence sequence), hidden sequence $\boldsymbol{X}$ and switching sequence $\boldsymbol{S}$ can be obtained by using inference algorithm. The inference can be described as follow.

We suppose that all continuous variables or conditional probability density functions in the DBN are Gaussian distribution, let:

$$P(x_0) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-\mu_0)^2}{\sigma_0^2}} = N\left(\mu_0, \sigma_0^2\right) \tag{6}$$

and let $P(\boldsymbol{s}_0) = \boldsymbol{N}(\mu_{\boldsymbol{s}0}, \sigma^2_{\boldsymbol{s}0})$, let state transition probability $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \boldsymbol{N}(\mu_{\boldsymbol{x}1}, \sigma^2_{\boldsymbol{x}1})$ and $P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t) = \boldsymbol{N}(\mu_{\boldsymbol{s}1}, \sigma^2_{\boldsymbol{s}1})$, condition transition probability $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{s}_t) = \boldsymbol{N}(\mu_{\boldsymbol{xs}}, \sigma^2_{\boldsymbol{xs}})$, let observation probability $P(\boldsymbol{e}_t \mid \boldsymbol{x}_t) = \boldsymbol{N}(\mu_{\boldsymbol{e}1}, \sigma^2_{\boldsymbol{e}1})$.

In general, the DBN learning can be formulated as the ML learning problem. In this paper, initial system parameters are set according to prior knowledge, and the parameters can also be adjusted according to user's satisfaction for motion retrieval results. When retrieval system runs for a period of time, the large number of retrieved data can be obtained, the EM algorithm [16] is used to find optimal values of the DBN parameters $\{\mu_{\boldsymbol{x}0}, \sigma^2_{\boldsymbol{x}0}, \mu_{\boldsymbol{s}0}, \sigma^2_{\boldsymbol{s}0}, \mu_{\boldsymbol{x}1}, \sigma^2_{\boldsymbol{x}1}, \mu_{\boldsymbol{s}1}, \sigma^2_{\boldsymbol{s}1}, \mu_{\boldsymbol{xs}}, \sigma^2_{\boldsymbol{xs}}, \mu_{\boldsymbol{e}1}, \sigma^2_{\boldsymbol{e}1}\}$.

As shown in Fig. 4, we can calculate $P(\boldsymbol{x}_1)$ based on the $\boldsymbol{x}_0$ and the $\boldsymbol{s}_0$ according to Bayesian rule:

$$\begin{aligned} P(x_1) &= \iint_{x_0,s_0} P(x_1|x_0,s_0)P(x_0,s_0)dx_0ds_0 \\ &= \sum_{s_0=1}^{k} P(s_0)\int_{x_0} P(x_0)P(x_1|x_0,s_0)dx_0 \end{aligned} \tag{7}$$

Based on Eq. (7), because the $P(\boldsymbol{x}_0,\boldsymbol{s}_0) = P(\boldsymbol{x}_0)P(\boldsymbol{s}_0)$ (variable conditional independence) and the $\boldsymbol{s}$ is the discrete signal. The next, predicted data can be updated by computing the $P(\boldsymbol{x}_1, \boldsymbol{s}_1|\boldsymbol{e}_1)$. According to the Bayesian rule and the DBN filtering equation [16], the predicted data can be updated:

$$\begin{aligned} P(X_{1+t}|e_{1:1+t}) &= \alpha P(e_{1+t}|X_{1+t})P(X_{1+t}|e_{1:t}) \\ &= \alpha P(e_{1+t}|X_{1+t})\textstyle\sum_{X_t} P(X_{1+t}|X_t)P(X_t|e_{1:t}) \end{aligned} \tag{8}$$

Where α is a parameter, which ensures computed results to be normalized [16]. In above equation, if to replace the $\boldsymbol{X}_t$ with $(\boldsymbol{x}_t, \boldsymbol{s}_t)$, then we have:

$$\begin{aligned} P(x_{t+1},s_{t+1}|e_{1:t+1}) &= \alpha P(e_{t+1}|x_{t+1},s_{t+1})\sum_{s_t=1}^{k}\int_{x_t} P(x_t,s_t|e_{1:t})P(x_{1+t},s_{1+t}|x_t,s_t) \\ &= \alpha P(e_{t+1}|x_{t+1})\sum_{s_t=1}^{k} P(x_t|e_{1:t})P(s_t|e_{1:t})\int_{x_t} P(x_{1+t}|x_t,s_t)P(s_{1+t}|x_t,s_t) \\ &= \alpha P(e_{t+1}|x_{t+1})\sum_{s_t=1}^{k} P(s_t|e_{1:t})P(s_{1+t}|s_t)\int_{x_t} P(x_t|e_{1:t})P(x_{1+t}|x_t,s_t) \end{aligned} \tag{9}$$

According to the above formula, the $\boldsymbol{x}_t$ and $\boldsymbol{s}_t$ can be updated based on observation data $\boldsymbol{e}_t$. The next, due to:

$$P(x_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|x_{t+1})\int_{x_t} P(x_{t+1}|x_t)P(x_t|e_{1:t}) \tag{10}$$

And because: $P(x_{t+1},s_{t+1}|e_{1:t+1}) = P(x_{t+1}|e_{1:t+1})P(s_{t+1}|e_{1:t+1})$, so we have:

$$\begin{aligned} P(s_{t+1}|e_{1:t+1}) &= \frac{P(x_{t+1},s_{t+1}|e_{1:t+1})}{P(x_{t+1}|e_{1:t+1})} \\ &= \frac{P(x_{t+1},s_{t+1}|e_{1:t+1})}{\alpha P(e_{t+1}|x_{t+1})\int_{x_t} P(x_{t+1}|x_t)P(x_t|e_{1:t})} \\ &= \frac{P(e_{t+1}|x_{t+1})\sum_{s_t=1}^{k} P(s_t|e_{1:t})P(s_{1+t}|s_t)\int_{x_t} P(x_t|e_{1:t})P(x_{1+t}|x_t,s_t)}{P(e_{t+1}|x_{t+1})\int_{x_t} P(x_{t+1}|x_t)P(x_t|e_{1:t})} \end{aligned} \tag{11}$$

Now, based on observation $\boldsymbol{E} = \{\boldsymbol{e}_1,\boldsymbol{e}_2,...,\boldsymbol{e}_T\}$, we can estimate the hidden sequence $\boldsymbol{X} = \{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_T\}$ and the state switching signals $\boldsymbol{S} = \{\boldsymbol{s}_1,\boldsymbol{s}_2,..., \boldsymbol{s}_T\}$ according to recurrence formula: from Eqs. (7) to (11). Lastly, all sequence signals would be transformed into the quantized and normalized signals (denoted as $\boldsymbol{E}_{\text{norm}}$, $\boldsymbol{X}_{\text{norm}}$, $\boldsymbol{S}_{\text{norm}}$), as shown in Fig. 5. We can use the matrix G to describe the motion:

$$G = [g(i,j)]_{3\times T} = [E_{\text{morm}}; X_{\text{norm}}; S_{\text{norm}}] = \begin{bmatrix} \overline{e}_1, & \overline{e}_2 & \cdots, & \overline{e}_T \\ \overline{x}_1, & \overline{x}_2 & \cdots, & \overline{x}_T \\ \overline{s}_1, & \overline{s}_2 & \cdots, & \overline{s}_T \end{bmatrix} \tag{12}$$

and $\overline{e}_i = e_i / \sum_{i=1}^{T} e_i; \overline{x}_i = x_i / \sum_{i=1}^{T} x_i; \overline{s}_i = s_i / \sum_{i=1}^{T} s_i$

Where $\overline{e}_i, \overline{x}_i, \overline{s}_i$ are the quantized and normalized values of $\boldsymbol{e}_i$, $\boldsymbol{x}_i$ and $\boldsymbol{s}_i$, respectively, and the T in Eq. (12) denotes length of motion.

We can further explain the DBN inference through a toy. For easy calculation and expression, we use discrete data instead of continuous data. As shown in Fig. 6, the DBN parameters are: $P(\boldsymbol{s}_0) = P(\boldsymbol{x}_0) = [0.5\ 0.5]$, and $P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t) = [0.3\ 0.7; 0.7\ 0.3]$, condition transition probability $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{s}_t) = [0.1\ 0.2\ 0.3\ 0.4; 0.9\ 0.8\ 0.7\ 0.6]$, $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = [0.2\ 0.8; 0.8\ 0.2]$, let observation probability $P(\boldsymbol{e}_t \mid \boldsymbol{x}_t) = [0.4\ 0.6; 0.6\ 0.4]$.

Now, if suppose all nodes have 2 states (denoted as *Ture* = 1, *False* = 2), let inputted signals $\boldsymbol{e}_1=1$, we can calculate $P(\boldsymbol{s}_1=1|\boldsymbol{e}_1=1)$ and $P(\boldsymbol{x}_1=1|\boldsymbol{e}_1=1)$.

Firstly, based on Eq. (9), we have:

$$\begin{aligned} P(x_{t+1},s_{t+1}|e_{1:t+1}) &= \alpha P(e_{t+1}|x_{t+1})\sum_{s_t=1}^{k} P(s_t|e_{1:t})P(s_{1+t}|s_t)\int_{x_t} P(x_t|e_{1:t})P(x_{1+t}|x_t,s_t) \\ &\approx \alpha P(e_{t+1}|x_{t+1})\sum_{s_t=1}^{k} P(s_t|e_{1:t})P(s_{1+t}|s_t)\sum_{x_t=1}^{k} P(x_t|e_{1:t})P(x_{1+t}|x_t,s_t) \end{aligned} \tag{13}$$

When t=0, we have:

$$\begin{aligned} &P(x_1=1,s_1=1|e_1=1) \\ &= P(e_1=1|x_1=1)(P(s_0=1)\times P(s_1=1|s_0=1)+P(s_0=2)\times P(s_1=1|s_0=2)) \\ &\times P(x_1=1|e_1=1)\begin{pmatrix}[P(x_1=1|x_0=1,s_0=1)+P(x_1=1|x_0=1,s_0=2)] \\ +[P(x_1=1|x_0=2,s_0=1)+P(x_1=1|x_0=2,s_0=2)]\end{pmatrix} \\ &= 0.4\times(0.5\times 0.3+0.5\times 0.7)\times P(x_1=1|e_1=1)([0.1+0.2]+[0.3+0.4]) \\ &= 0.4\times 0.5\times P(x_1=1|e_1=1) \end{aligned}$$

Similar, we have $P(x_1=1,s_1=2|e_1=1) = 0.2\times P(x_1=1|e_1=1)$. (14)

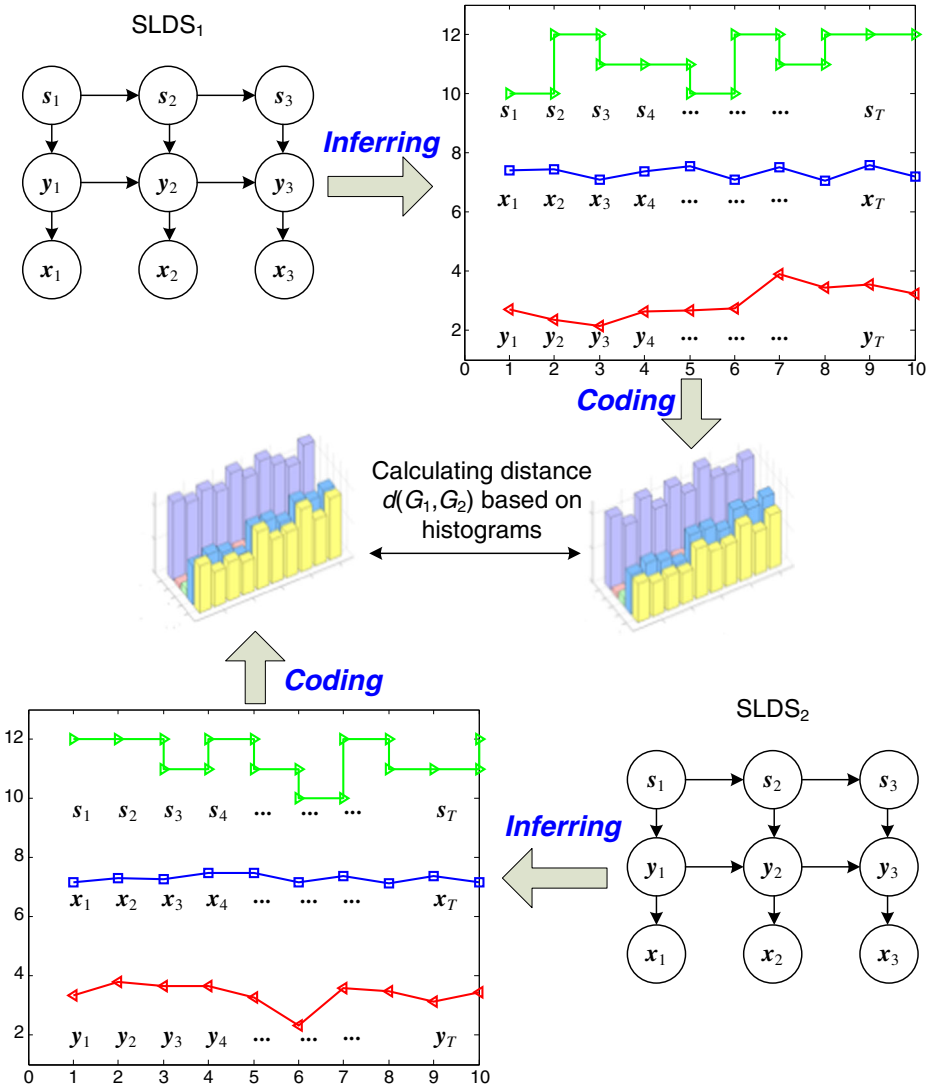


Fig. 5 Inference and coding based on the S-KFM

Then we can get $P(\boldsymbol{x}_1=1|\boldsymbol{e}_1=1)$ according to Eq. (10):

$$\begin{aligned} &P(x_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|x_{t+1}) \int_{x_t} P(x_{t+1}|x_t) P(x_t|e_{1:t}) \\ &\Rightarrow P(x_1 = 1|e_1 = 1) \approx P(e_1 = 1|x_1 = 1) \sum_{x_0} P(x_1|x_0) P(x_0) \\ &= P(e_1 = 1|x_1 = 1)(P(x_1 = 1|x_0 = 1)P(x_0 = 1) + P(x_1 = 1|x_0 = 2)P(x_0 = 2)) \\ &= 0.4 \times (0.2 \times 0.5 + 0.8 \times 0.5) = 0.2 \end{aligned} \tag{15}$$

Similar, we have:

$$\begin{aligned} &P(x_1 = 2|e_1 = 1) \\ &\approx P(e_1 = 1|x_1 = 2)(P(x_1 = 2|x_0 = 1)P(x_0 = 1) + P(x_1 = 2|x_0 = 2)P(x_0 = 2)) \\ &= 0.6 \times (0.8 \times 0.5 + 0.2 \times 0.5) = 0.3 \end{aligned} \tag{16}$$

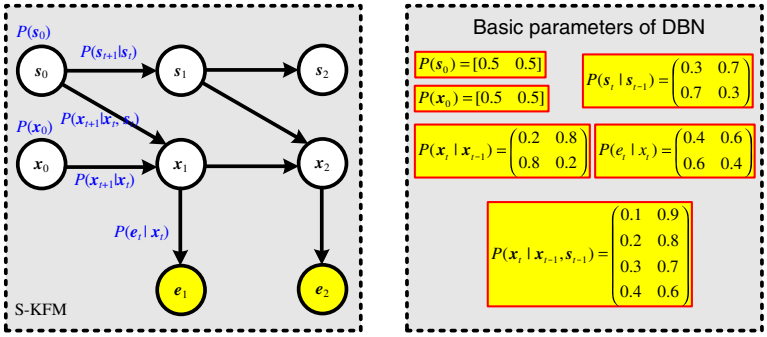


Fig. 6 An example of the DBN inference

Then, we get: $P(x_1|e_1 = 1) = \alpha[0.2 \quad 0.3] = [0.4 \quad 0.6]$, we put Eq. (16) into Eq. (14), have:

$$P(x_1 = 1, s_1 = 1|e_1 = 1) = 0.4 \times 0.5 \times 0.4 = 0.08 \tag{17}$$

Then we get:

$$P(s_1 = 1|e_1 = 1) = \frac{P(x_1 = 1, s_1 = 1|e_1 = 1)}{P(x_1 = 1|e_1 = 1)} = \frac{0.08}{0.4} = 0.2 \tag{18}$$

$$P(s_1 = 2|e_1 = 1) = \frac{P(x_1 = 1, s_1 = 2|e_1 = 1)}{P(x_1 = 1|e_1 = 1)} = \frac{0.08}{0.4} = 0.2 \tag{19}$$

In the end, we get calculation results: $P(s_1|e_1 = 1) = \alpha[0.2 \quad 0.2] = [0.5 \quad 0.5]$.

3.2 Motion retrieval

In stage of matching or retrieval, the query motion is extracted feature according to above, and then χ^2-test is used to compute distances between query and motions, as shown in Fig. 5, the retrieval results can be sorted and outputted. Assume there are two motions (denoted as G_1 and G_2), we have:

$$\begin{aligned} & d(\text{motion}_1, \text{motion}_2) \\ & = d(G_1, G_2) = \sum_{i=1}^{3}\sum_{j=1}^{T} \frac{|g_1(i,j) - g_2(i,j)|^2}{g_1(i,j) + g_2(i,j)} \\ & = \sum_{j=1}^{T} \frac{|g_1(1,j) - g_2(1,j)|^2}{g_1(1,j) + g_2(1,j)} + \sum_{j=1}^{T} \frac{|g_1(2,j) - g_2(2,j)|^2}{g_1(2,j) + g_2(2,j)} + \sum_{j=1}^{T} \frac{|g_1(3,j) - g_2(3,j)|^2}{g_1(3,j) + g_2(3,j)} \\ & = \sum_{j=1}^{T} \frac{\left|\bar{e}_j^{(1)} - \bar{e}_j^{(2)}\right|^2}{\bar{e}_j^{(1)} + \bar{e}_j^{(2)}} + \sum_{j=1}^{T} \frac{\left|\bar{x}_j^{(1)} - \bar{x}_j^{(2)}\right|^2}{\bar{x}_j^{(1)} + \bar{x}_j^{(2)}} + \sum_{j=1}^{T} \frac{\left|\bar{s}_j^{(1)} - \bar{s}_j^{(2)}\right|^2}{\bar{s}_j^{(1)} + \bar{s}_j^{(2)}} \end{aligned} \tag{20}$$

Where $g_1(i,j)$ and $g_2(i,j)$ are elements of G_1 and G_2, respectively. The $\bar{e}_j^{(1)}, \bar{s}_j^{(1)}, \bar{x}_j^{(1)}$ denote the j-th column elements in G_1, the $\bar{e}_j^{(2)}, \bar{s}_j^{(2)}, \bar{x}_j^{(2)}$ denote the j-th column elements in G_2. The parameters T in Eq. (20) denotes length of motion.

At last, the matching results are sorted according to distances between the query motion and motions in database, the top-p (p is feedback number of retrieval, the p is usually set to 20) motions can be feedback to user.

4 Experiments

To evaluate performance of the proposed motion descriptor on different motion clips, we discuss some of the experiments that we have conducted. We have constructed a motion database from 1000 different motion clips. For easy to test effective of motion matching and retrieval, we cut all motions into uniform length. We categorize the 1000 motions into 20 motion groups, normal speed walking, fast walking, slow walking, leg-wild walking, jumping, and so on. All the experiments presented here are performed on a PC with a Pentium 5 GHz CPU and 1 GB RAM. The motion files are downloaded from CMU [5].

The motion clips typically contain more than one action within each clip. To obtain more accurate performance results, we manually break each of the clips down into basic motion clips with a single action. We use the basic motion clips as input and we take the first 50 frames of the basic motion clips as the query for scale computation. Our objective in the experiments is to find the most similar motion clips within the motion database. For comparison, we have implemented Dynamic Time Warping [8] and the Uniform Scaling method [10].

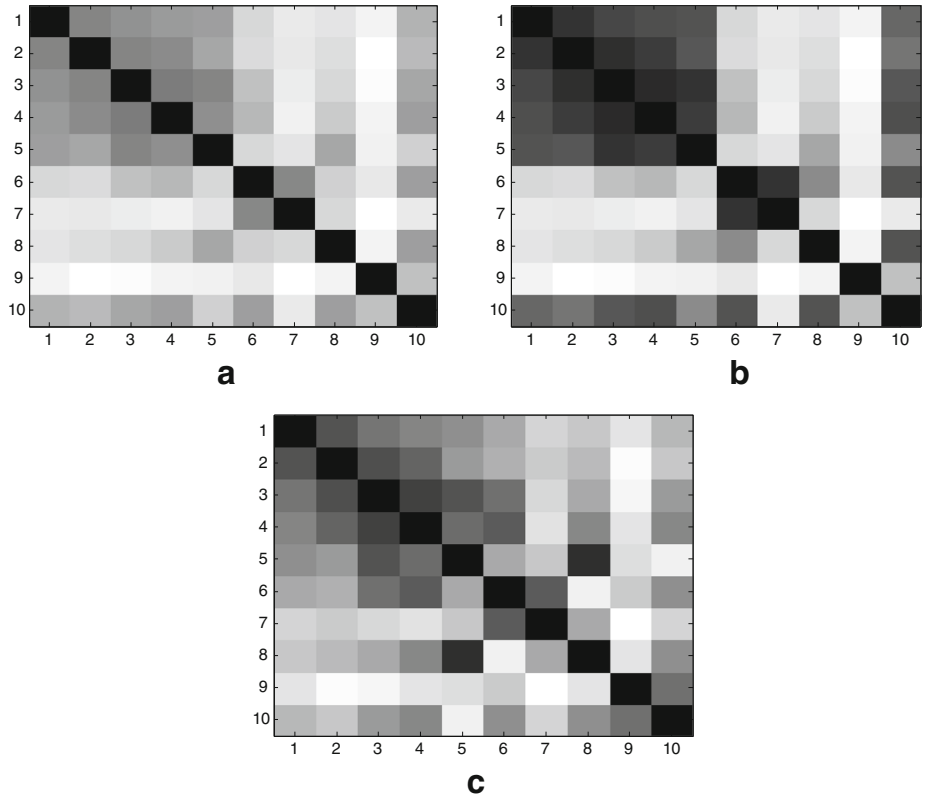


Fig. 7 Similarity score between every motion pair

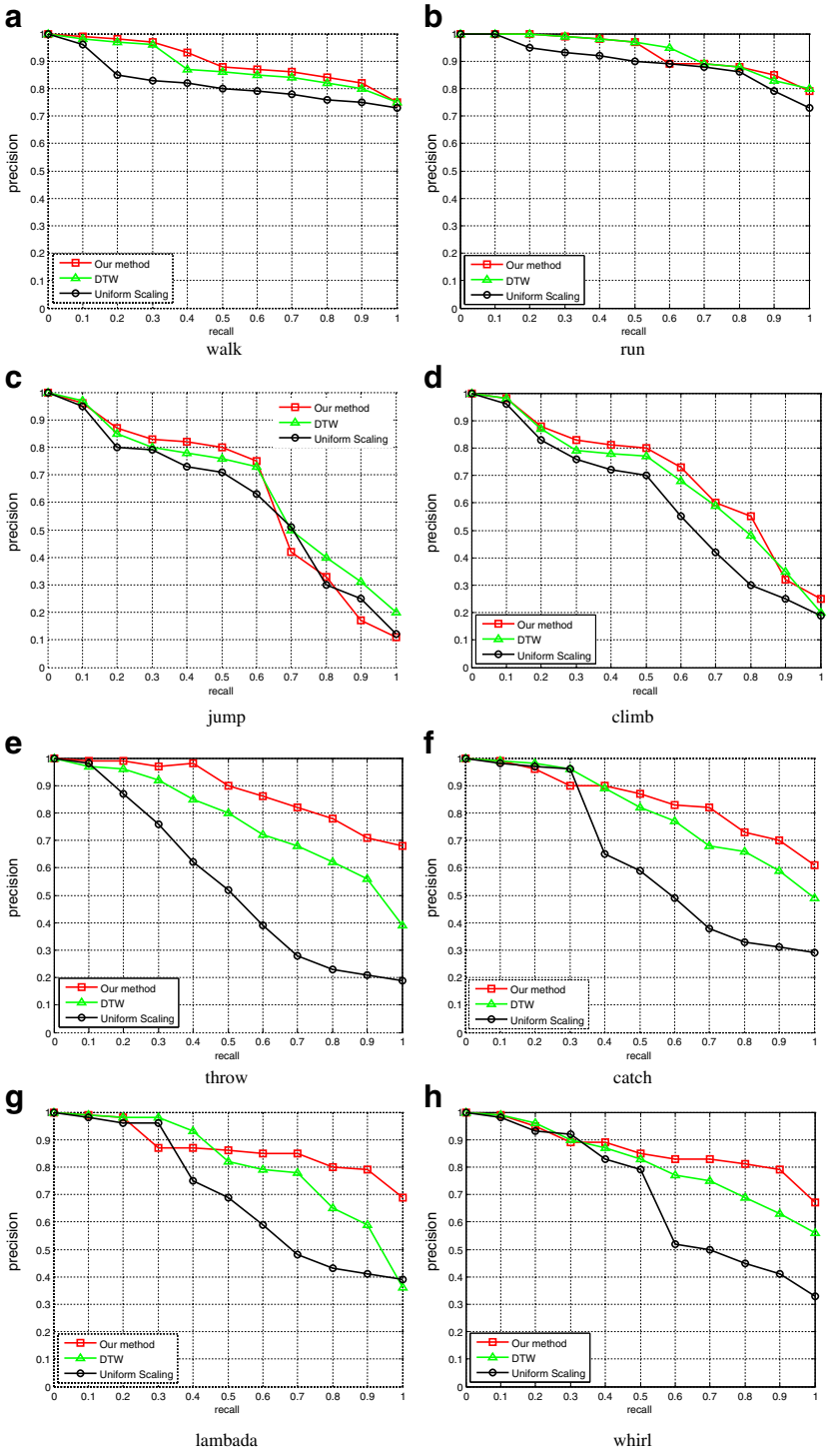


Fig. 8 Some precision-recall curves for motion retrieval in database

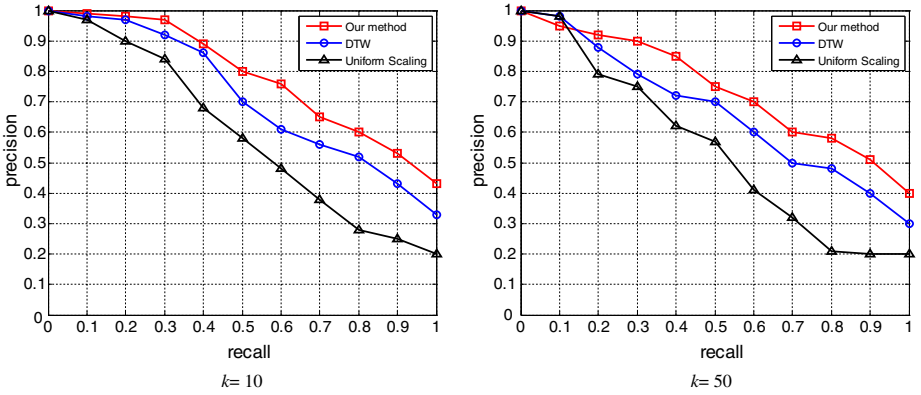


Fig. 9 Comparison precision-recall curves of three methods. The k is feedback number of query results

4.1 Performance on motion discrimination

In the first experiment, we compare retrieval performance of the three methods, Uniform Scaling, DTW, and our method, using the similarity matrix. To generate matrix, we first compute similarity score between every motion pair in our database. We then normalize the results with the maximum and minimum of the corresponding matrices to show the contrasts. The darker the color, the more similar the two motions are. Figure 7 shows the similarity matrices of the three methods. From the similarity matrices, we have the following observations: on the one hand, the diagonal lines of the matrices give the darkest color. This means our method performs well in identifying the same motion. This means that proposed method is able to give high similarity scores for similar motions. On the other hand, our method also gives a larger similarity contrast when comparing two motions from different groups. This may suggest that our method can distinguish different motion groups relatively easier.

4.2 Performance on motion retrieval

In the second experiment, we calculate average precision and recall value (as shown in Fig. 8). Those diagrams are generated by taking each of the motions in the database as query, searching similar motions from the same database and averaging all the precision and recall values.

The Fig. 8 shows part of our precision and recall results. From the diagram, our method performs well. This finding confirms our similarity analysis that our method can distinguish dissimilar motions. The Fig. 8 plots the precision and recall curves for 8 motions in our database, from retrieval results, we observe that: for simple motions, such as run, walk, and so on, the compared 3 methods all have the best performance. On the other hand, for some

Table 1 Running time comparison (per query)

Measurement approach	Class of motion		
	Fast walk	Slow walk	Jump-kick
Uniform scaling	250 s	230 s	110 s
DTW	220 s	370 s	135 s
Our method	9.72 s	28.3 s	12.8 s

motions with up or down direction, such as jump, climb and so on, the compared 3 methods have better performance, our method still has better performance than other approaches. The proposed retrieval algorithm always have low effective to motion with up to down direction, the main reason is that there is lesser discrimination for up and down movement based on proposed view-based motion coding method. It is just the weakness of this proposed algorithm, we will improve that in future works. However, for complex motions, such as dance, throw, and so on, our method has very good performance than other methods, that means, our retrieval frame is not only suited for simple motion retrieval, but also suited for complex motion discrimination.

In Fig. 9, we compare the 3 methods (our method, DTW and US) based on proposed retrieval frame and averaging precision and recall value, we can see that all three methods perform very well, whilst our method performs better.

4.3 Speed comparison

In the third experiment, we would like to compare the performance of the 3 methods according to retrieval speed. The parts of experimental results are shown in Table. 1, which reveal that our method actually performs better than DTW and Uniform Scaling. This is because DTW or Uniform Scaling computes all the motion frames, but our method, by applying the graph model and coding algorithm, involves only a matrix. This explains why the computation time consumed by our method is far less than other methods.

Based on the computational complexity, we can also explain why our method outperforms existed methods. In Uniform Scaling, we try to find the best scaled match between the query and the candidate. So, the time complexity is $O(p \times (m\text{-}n))$, where p, m and n represent the lengths of a scaled time series, the candidate and the query, respectively. The time complexity of DTW is roughly $O(m \times n)$. The time complexity of our method is harder to analyze because it is based on the simplex algorithm. However, if the algorithm is modeled as a matrix matching problem, the complexity is $O(n)$.

5 Conclusion

We have introduced a novel and efficient method for retrieving human motion data. Unlike other approaches, our method applies the graph model to describe motion, through inferring and coding, a small size and robust motion descriptor can be obtained. Our experiments show encouraging results.

Acknowledgment This work is partly supported by the National Basic Research Project of China (No. 2010CB731800) and the China National Foundation (No. 60972095, 61271362).

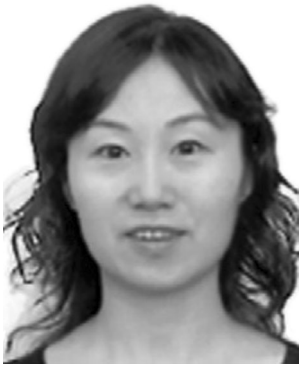
References

1. Bashir FI, Khokhar AA, Schonfeld D (2007) Real-time motion trajectory-based indexing and retrieval of video sequences. IEEE Trans Multimed 9:58–65
2. Chakrabarti K, Keogh E, Mehrotra S, Pazzani M (2002) Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans Database Syst 27(2):188–228
3. Chao M-W, Lin C-H, Assa J, Lee T-Y (2012) Human motion retrieval from hand-drawn sketch. IEEE Trans Vis Comput Graph 18(5):729–740
4. Feng L, Yueting Z, Fei W, Yunhe P (2003) 3D motion retrieval with motion index tree. Comp Vision Image Underst 92(2):265–284
5. Gao Y, Tang J, Hong R, Yan S, Dai Q, Zhang N, Chua T-S (2012) Camera constraint-free view-based 3-D object retrieval. IEEE Trans Image Process 21(4):2269–2281

6. Gao Y, Wang M, Zha Z-J, Tian Q, Dai Q, Zhang N (2011) Less is more: efficient 3-D object retrieval with query view selection. IEEE Trans Multimed 13(5):1007–1018
7. Graphics Lab. "Motion capture database". Carnegie Mellon University, http://mocap.cs.cmu.edu/
8. Keogh E, Palpanas T, Zordan V, Gunopulos D, Cardle M (2004) "Indexing large human-motion databases." Proc VLDB: 780–791
9. Kovar L, Gleicher M, Pighin F (2002) "Motion graphs." Proc ACM SIGGRAPH: 473–482
10. Lin Y (2006) "Efficient human motion retrieval in large databases." Proc ACM GRAPHITE: 31–37
11. Müller M, Röder T (2006) "Motion templates for automatic classification and retrieval of motion capture data." Proc ACM SCA
12. Nixon MS, Aguado AS (2008) "Feature extraction and image processing", Second edition, published by Elsevier, pp 135–140
13. Pavlovíc V, Rehg JM, Murphy KP, Cham T-J (1999) "A dynamic Bayesian network approach to figure tracking using learned dynamic models". Intl. Conf. Computer Vision: 94–101
14. Qian H, Debin Z, Siwei M, Wen G, Huifang S (2010) Deinterlacing using hierarchical motion analysis. IEEE Trans Circ Syst Video Technol 20(5):673–686
15. Qinkun X, Haiyun W, Fei L, Yue G (2011) 3D object retrieval based on a graph model descriptor. Neurocomputing 74(17):2340–2348
16. Russell S, Norvig P (2004) "Artificial intelligence: a modern approach", Second edition, published by Pearson Education Asia Limited, pp 430–436
17. Shah VP, Younan NH, King RL (2008) An efficient pan-sharpening method via a combined adaptive PCA approach and contourlets. IEEE Trans Geosci Remote Sens 46(5):1323–1335
18. Tam GKL, Lau RWH (2007) Deformable model retrieval based on topological and geometric signatures. IEEE Trans Vis Comput Graph 13(3):470–482
19. Tang Jeff KT, Leung H (2012) Retrieval of logically relevant 3D human motions by adaptive feature selection with graded relevance feedback. Pattern Recognit Lett 33(4):420–430
20. Tian J, Qi W, Liu X (2011) Retrieving deep web data through multi-attributes interfaces with structured queries. Int J Softw Eng Knowl Eng 21(4):523–542
21. Xiao QK, Dai QH, Wang HY (2008) 3D object retrieval approach based on directed acyclic graph lightfield feature. Electron Lett 44(14):847–849
22. Xiaobai L, Liang L, Shuicheng Y, Hai J (2011) Adaptive object tracking by learning hybrid template online. IEEE Trans Circ Syst Video Technol 21(11):1588–1599
23. Xiaohua D, Liang L, Hongyang C (2013) Discovering video shot categories by unsupervised stochastic graph partition. IEEE Trans Multimed (TMM) 15(1):167–180
24. Yang Y, Nie F, Xu D, Luo J, Zhuang Y, Pan Y (2012) A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. IEEE Trans Pattern Anal Mach Intell 34(4):723–742
25. Zhang Z, Tao D (2012) Slow feature analysis for human action recognition. IEEE Trans Pattern Anal Mach Intell 34(3):436–450

Qinkun Xiao was born in 1974. He is a Ph.D. and an associate professor in Xi'an technological University. He obtained doctor degree from Northwestern Polytechnic University in 2007, and from 2007 to 2009, he is postdoctoral in Tsinghua University. His research interests include 3D object retrieval, dynamic Bayesian network and image processing. E-mail: xiaoqinkun10000@163.com.

Yichuang Luo was born in 1987 and he is currently a graduated student in Xi'an technological university. His research interests include motion retrieval and video information processing. E-mail: 623634511@qq.com.



Haiyun Wang was born in 1974 and she received the B.S. degree from Northwestern Polytechnic University in 1996, and Ph.D. in Nanyang Technological University in 2003, she is currently the senior engineer in STMicroelectronics R&D of Asia-Pacificin Singapore. Her research interests include image and video information processing. E-mail: haiyun_w@gmail.com.