AUTOMATIC KEY-FRAMES EXTRACTION OF HUMANOID MOTIONS

Chin-Hung Ko   Jia-Yi Li   Tain-Chi Lu*

Department of Computer Science and Information Engineering
National Chiayi University
Chiayi, Taiwan 60004, R.O.C.

Key Words: character animation, principal component analysis, motion saliency, key-frame extraction.

ABSTRACT

3D authoring tools provide a wide variety of motion editing functions to users to make motions appear more plausible and natural. Animators obviously need to spend much time and invest considerable efforts to edit great amounts of frames for the sake of obtaining their desired animation results. Therefore, animators try to reduce their burdens by picking up the appropriate key-frames, editing these chosen key-frames, and interpolating them for their desired results. Even animators with sufficient experience in animation may need to use several number of times to gain the appropriate key-frames. In this paper, we present a method to overcome aforementioned key-frame problem by automatically picking up the most significant key-frames from a continuous motion clip. Users only need to input a target motion clip in Biovision Hierarchy (BVH) format and provide a series of parameters, and the proposed method will extract the most significant key-frames from target motion clips automatically. In order to verify the quality of extracted key-frames, we directly interpolate the extracted key-frames and compare them with the input motion clip. The experimental results suggest that the proposed method is less labor-exhaustive, and it decreases the operation time without sacrificing original motion features.

I. INTRODUCTION

Nowadays, people are familiar with a wide range of human motions including entertaining animations, computer games, movies, human computer interactions, and so on. Motion captures (MoCap) are exerted to provide highly precise representations of actual human movements in real time. However, MoCap data can be inflexible because it requires recapturing motions in case the motions need to be modified for changings with certain purposes. Otherwise, animators had better make use of motion editors to refine motions manually. Instead of modifying each frame to accomplish motion editing, modifying key-frames and interpolating the in-between key-frames are a cost-saving way to implement. Unfortunately, key-frame selection is a time-consuming task for a motion editor. If key-frame selection is not performed well and is less accurate, the interpolation of key-frames leads to an unsmooth motion results. In this paper, we propose an automatic key-frame extraction method by means of motion saliency calculating and k-means clustering, which can extract representative key-frames from a clip of motion sequence. In practice, a motion sequence is recorded with variations of all articulated joints for each frame. It is complicated to deal with the original motion data directly. The first step is to reduce the dimensionality of the input motion sequence. We treat the input motion sequence as a set of motion curves and perform the dimensionality reduction by utilizing the principal components analysis (PCA) [1]. We also regard a frame in a motion sequence as a key-point on a motion curve, and calculate curvatures at each key-point. After obtaining the curvatures of the reduced motion curves, we extend them into the Gaussian-

* Corresponding author: Tain-Chi Lu, e-mail: tclu@mail.nchu.edu.tw
weighted mean curvature and combine the center-surrounding operation to calculate motion saliency for each frame. Subsequently, we choose candidate key-frames according to the saliency of the frames. In the last phase of our method, we apply k-means clustering to obtain the most important key-frames of the input motion sequence. We also interpolate these extracted key-frames to recover whole motion sequence for verifying the validity and efficiency. Our experimental results show that the proposed method can lead to provide lesser key-frames with satisfactory visual quality.

The remainder of this paper is structured as follows: Section II briefly surveys the related works. Section III presents the framework of the proposed key-frame extraction method and elaboration on the proposed method in detail. In Section IV, we show the experimental results in comparison with the relevant schemes. Finally, we present our conclusion and along with suggestions for future research in Section V.

II. RELATED WORK

Motion editing is an ongoing and challenging topic in many areas. It has received a great deal of prior attention to make it more flexible and less cumbersome. For example, 3DS Max [2] let users modify the joint data directly in any frame of a motion sequence, and it performs the interpolation between key-frames which specified by users. In this section, we survey related methods about numerous key-frame extracting approaches, such as mathematical-model-based methods, clustering-based approaches, and curve-simplification-based approaches. Kim and Kim [3] defined five clustering orders to classify the candidate key-frames, and extracted key-frames from the most effective clusters. Key-frames also can be extracted by employing the mathematical models to analyze a motion sequence. Qi et al. [4] utilized the Gaussian mixture model (GMM) to train a probability distribution with a key-pose of a motion sequence, and extracted key-frames from well-trained probability distribution model. Tong et al. [5] analyzed and predicted the motion cycle of a motion sequence, and then they extracted key-frames through dominant local motion region (DLMR). Ratsamee et al. [6] took advantage of Electro Dermal Activity (EDA) to extract key-frames. The values of EDA are proportional to the activation of the target. They assigned each frame a weighted EDA value, and applied the filter to accomplish the extraction. Clustering-based approaches are also widely used in key-frame extraction [7-10]. Before applying the clustering approaches, the similarity between frames should be calculated first. Then we applied the clustering approaches and selected the center of each cluster as the extracted key-frames. Instead of performing clustering methods on joint angles of each frame, Zeppelzauer et al. [11] concentrated on the motion trajectories. They split and merged motion trajectories iteratively until the key-frames were decided. Curve-simplification-based approaches focus on simplifying motion curves as straight lines or intervals to obtain key-frames [12-14]. The purpose on the curve simplification is to reduce the complexity of motion curves and to make the calculation more efficient for the remaining process. Principal component analysis (PCA) is a commonly used approach, which reduces high dimension data into low dimension data by calculating the eigenvalue and eigenvector of the high dimension one. A number of approaches [15-16] utilize PCA to compress or remove redundancies of high dimension and complex sources, and then they focus on those remaining parts which reserved the attributes. Li et al. [17] searched and weighted the inflection point on the complex motion curves first, and then applied the curve simplification to obtain key-frames. Some researchers [18-20] -have applied the concept of saliency [21], which usually is normally used on 3D mesh simplification, to key-frame extraction. Halit and Capin [19] treat the motion sequence as motion curves and finding the most salient parts of the motion curves for key framing. They also apply PCA to obtain the reduced motion curves at first, and then calculated the curvatures [22] of reduced curves. Furthermore, they extended these curvatures into Gaussian weighted mean curvature [10] and constructed the saliency map to split the center frame and surrounding frames. Finally, they utilized clustering method to extract desired key-frames.

In this paper, we refer to Halit and Capin’s method [19] by combining the curve simplification, motion saliency, and clustering method to propose a key-frame extraction method. In addition, we turn our attention to improve results by utilizing the k-means clustering [23] on the clustering stage.

III. METHODOLOGY

Fig. 1 shows the overview of the proposed method. The input of our key-frame extraction framework is a human motion clip in BVH format, a standard deviation constant for Gaussian-weighted mean curvatures, and a weighted constant for k-means clustering. After finishing the input stage, there are two major processes in our method, including sa-
liency calculating, and candidate key-frames clustering. In the first process, we start with exploiting PCA to reduce the dimensionality of input motion data for obtaining the reduced curves. In the following, we calculate the Gaussian-weighted mean curvatures for each reduced curve with several different standard deviations. Furthermore, we construct the saliency map by utilizing the center-surround operator to calculate the saliency values with the results of the previous step. After normalizing and mapping the saliency map onto the saliency curves, we select the candidate key-frames from the inflection points of the saliency curves. The second process is to cluster these candidate key-frames with k-means clustering. Finally, we extract the candidate key-frames to treat as the eventual key-frames from the center of each cluster.

1. Saliency Calculation

Human motion sequence data generally comprise a series of motion frames. Each frame is the records of the joints’ rotations and translations in terms of an articulated skeleton. These joints comprise real moving parameters, which always hold the explicit temporal and spatial information, such as positions and orientations. However, it is a bit tedious to manipulate such data directly because the large number of frames. Given a sequence of 3D motion \( M \in \mathbb{R}^{n \times m} \) with \( n \) frames and a \( m \)-joints-skeleton, the first step of our method is to utilize PCA to reduce the dimensionality of the input motion sequence. After accomplishing the dimensionality reduction, we can obtain \( \alpha \) reduced curves \( D \in \mathbb{R}^{\alpha \times l} \) with \( n \) key-points.

\[
M = \left[ f_1 = \{ j_{i1}, j_{i2}, \ldots, j_{im} \} \right] \\
\vdots \\
\left[ f_\alpha = \{ j_{i1}, j_{i2}, \ldots, j_{im} \} \right]
\]

(1)

Fig. 2 shows the eigenvalues and the representing ratio of the reduced curves, which stand for the significant components of the input motion sequence.

\[
D_i = \{ k_{i1}, k_{i2}, \ldots, k_{im} \}, i = 1 to \alpha
\]

(2)

According to the definition of curvature [22], the curvature on each key-point from the reduced curves can be defined as

\[
C(D_j) = \frac{|D'_j|}{(1 + D_j)'^2}
\]

(3)

where \( i \) is the key-point and \( j \) is the reduced curve. In addition, we referred to Lee et al. [21] to utilize the center surround operation for extending the reduced curve to
2. Candidate Key-Frame Clustering

In order to acquire appropriate amount of key-frames, the number of clusters is required to be decided first. Support that each saliency curve provides \( n \) candidate key-frames, the number of clusters \( N \) can be decided by:

\[
N = \frac{\omega \sum_{i=1}^{\text{CurveNum}} (n_i \times \lambda_i)}{\text{CurveNum}}
\]  

(7)

\[
\lambda_i = \frac{\lambda_i}{\lambda_{\text{sum}}}
\]  

(8)

where \( \lambda_i \) is the eigenvalue of the saliency curve \( S_i \), and \( \lambda_{\text{sum}} \) is the sum of the eigenvalues of all saliency curves. Note that \( \omega \) is a weighted value which controls the output amount of key-frames. We will discuss the possible effect on this weighted value in section 4. Within all the candidate key-frames and the number of clusters, we employ the k-means clustering to acquire the required key-frames.

IV. EXPERIMENTS

In this section, we experiment on the proposed method for showing the implementation results of extracting key-frames from human motion sequences. We implement the proposed method by means of MATLAB. We utilize the human MoCap database from CMU MoCap Library [25] as the motion input. Fig. 4 shows the extracted key-frames from the 5th motion sequence of the subject #7 (denoted as 07_05) in the CMU motion database. Since key-frame extraction can be regarded as a form of data compression, we employ the peak signal-to-noise ratio (PSNR) [24] to evaluate the accuracy of our method. We reconstruct the whole character motion by linear interpolation with the extracted key-frames, and calculate the PSNR between the reconstructed motion and the original CMU motion to measure the quality of the reconstructed motion. Note that the value of PSNR is positive proportional to the quality of the reconstructed motion, i.e., a higher PSNR value leads to a higher similarity between the reconstructed motion and original motion. Typically, the value for PSNR in image and video compression are between 30 and 50 dB.

In the proposed method, there are three parameters, \( \sigma \), \( \lambda \), and \( \omega \) which can be assigned to control the quality and amount of the extracted key-frames. The first parameter \( \alpha \) is utilized for dimension reduction which represent the number of reduced curves. The number of reduced curves

Gaussian-weighted mean curvature. Let \( f \) denote the key-point \( j \) on the reduced curve \( i \), and \( N(f, \sigma) \) denote the neighborhood set of \( f \) with a distance \( \sigma \) which is specified by users, i.e., \( \{ x \mid \|x - f\| < \sigma, x \text{ is a point on the reduced curve } i \} \), the Gaussian-weighted mean of the mean curvature can be defined as:

\[
G(f, \sigma) = \frac{\sum_{x \in N(f, \sigma)} m(f) e^{-\frac{1}{2} \left( \frac{\|x - f\|}{\sigma} \right)^2}}{\sum_{x \in N(f, \sigma)} e^{-\frac{1}{2} \left( \frac{\|x - f\|}{\sigma} \right)^2}}
\]  

(4)

where \( m(f) \) denotes the mean curvature of \( f \). Furthermore, we use four scales of \( \sigma \) to vary the coverage of the Gaussian-weighted mean of the mean curvature. Thus, we can calculate the saliency maps of reduced curves for each scale of \( \sigma \) by Equation (5):

\[
S_i(f) = |G(f, \sigma_i) - G(f, 2\sigma_i)|
\]  

(5)

To ensure each saliency map \( V \) has the corresponding influence on all scales of saliency map, we utilize the normalization [24] defined as follows:

\[
V' = \frac{V - \text{Min}}{\text{Max} - \text{Min}} \times (\text{Max} - \text{Min}) + \text{Min}
\]  

(6)

where \( \text{Max} \) and \( \text{Min} \) are the global maximum and minimum saliency values, \( \text{Max} \) and \( \text{Min} \) are the local maximum and minimum in the same scale of \( \sigma \). We also employ the non-linear normalization on the normalized saliency map by multiplying a coefficient \( c = (\text{Max}_i - \text{Min}_i)^2 \), and we sum up all scales of the saliency map to obtain the saliency curve. As shown in Fig. 3, we choose the inflection points of the saliency curves as the candidate key-frames.
The key-frames extracted by the proposed method with #07_05 describes the capacity of representing the high-dimensional motion sequence. With the higher the value of $\alpha$, the representing percentage of the reduced curves is much higher. To avoid fierce distortion, the value of $\alpha$ we set in this paper represents over 90 percent component of original motion data. Due to the fact that human motions could variate widely, we divide the human motion data into two types, low dynamic type and high dynamic type. For the motion in the low dynamic type which are moderate motions, such as a slow walk or a walk with a slight variation, we set $\alpha = 7$. On the contrary, motions in the type of high dynamic, such as boxing and dribbling basketball, which are more complex than the ones belonging to the low dynamic type, we set $\alpha = 10$. Table 1 shows the details of extracting key-frames in both categories of low and high dynamic motions. In this case, we set $\sigma = 1$ and $\omega = 2$ for four motions.

The results of above tests show that proposed method can produce a satisfactory visual quality no matter what type motion is or how many frames exist. However, the motions of low dynamic lead to higher PSNR than those in high dynamic due to the variation of motion. That is, fiercer motion will result in lower PSNR. The following parameter $\sigma$ is used to control the number of neighboring frame sets during the calculation of Gaussian weighted mean curvature. Due to larger $\sigma$ increasing larger clusters, it leads to a smaller number of key-frames in result. That is, larger $\sigma$ can lead to fewer key-frames, and decrease the accuracy of the reconstructed motion. However, the last parameter $\omega$ also affect the amount of clusters.
Table 1  PSNR values and compression ratio for reconstructed motions

<table>
<thead>
<tr>
<th>Motions</th>
<th>Slow Walking</th>
<th>Walking</th>
<th>Basketball</th>
<th>Boxing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount of frames</td>
<td>449</td>
<td>4273</td>
<td>721</td>
<td>3000</td>
</tr>
<tr>
<td>Amount of key-frames</td>
<td>39</td>
<td>415</td>
<td>64</td>
<td>308</td>
</tr>
<tr>
<td>Reconstructed Motion PSNR</td>
<td>51.65</td>
<td>49.35</td>
<td>48.19</td>
<td>48.32</td>
</tr>
<tr>
<td>Compression Ratio</td>
<td>11.51</td>
<td>10.30</td>
<td>11.27</td>
<td>9.74</td>
</tr>
</tbody>
</table>

Fig. 5  Effect of parameter $\sigma$ and $\omega$ on PSRN for low dynamic ((a) and (b)) and high dynamic ((c) and (d)) motions

Fig. 6  Effect of parameter $\sigma$ and $\omega$ on execution time for low-dynamic ((a) and (b)) and high-dynamic ((c) and (d)) motions
before performing k-means clustering. In addition, k-means might lead to a local minimum, so that we have tested on a different range of $\sigma$ and $\omega$ a hundred time for each pair of parameters, and choose the average value as results. As shown in Fig. 5, our tests show that smaller $\sigma$ and larger $\omega$ will result in a satisfactory visual quality of reconstructed motion for both type of motions. Besides, low-dynamic motions have better visual quality than high dynamic motions when $\omega$ is small.

Fig. 6 shows the execution time for different motion type and frame numbers. The frame numbers of walk, slow walk, basketball, and boxing are 557, 449, 722, and 3000, which captured with 120 frame/second. It can be found that the execution time is positive proportional to frame numbers, $\sigma$, and $\omega$. From above experimental results, we can find out that the significant factors, which can affect PSNR, are the contents of motions and the designed parameters. The frame number of input motion affects slightly on PSNR but significantly on execution time.

We also compare our method with Halit and Capin’s method in stretching and boxing cases. We choose #114_10 as the low-dynamic motion and #13_18 as the high-dynamic motion. As shown in Table 2, our method has a close but smaller compression rate level and value of PSNR with the low-dynamic motion, but a better compression rate and PSNR with the one in the high-dynamic motion.

<table>
<thead>
<tr>
<th>Table 2: Comparison in the low-dynamic motion: Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halit and Capin’s method</td>
</tr>
<tr>
<td>Stretching Boxing</td>
</tr>
<tr>
<td>No. of frames</td>
</tr>
<tr>
<td>No. of key-frames</td>
</tr>
<tr>
<td>Compression Rate</td>
</tr>
<tr>
<td>PSNR</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have exploited PCA, motion saliency, and k-means clustering to develop an automatic key-frame extracting framework for human motion sequences. We have first utilized PCA to reduce the dimension of motion curves from input motion sequence. Second, we have calculated Gaussian-weighted mean curvatures with the center-surrounding operation for each reduced curve. With the scales variation of the center-surrounding operation, we can obtain saliency values for each motion curves. Continuously, we have normalized and merged all scales of the saliency values to acquire the saliency curves. The inflection points of the saliency curves can be selected as the candidate key-frames. Finally, we have employed the k-means clustering on the candidate key-frames and chosen the center of each clusters as our key-frames.

We have also reconstructed the motion sequence with the extracted key-frames, and evaluated the reconstructing results of the proposed method by calculating the PSNR value. The experimental results show that our approach provided a high compression ratio with 40-50 PSNR values. Due to the effective parameters $\sigma$ and $\omega$ are designed as the manual inputs, users can modify these two parameters to generate their desired number of key-frames. Note that PSNR is proportional to the amount of key-frames, users need to experiment with the above two parameters for obtaining their anticipated results. Therefore, automatic parameter optimization is our future research issue.

**NOTATION**

$M$ A matrix of a 3D motion sequences with $n$ frames and a skeleton of $m$ joints
$D$ A reduced curve in a vector form which consists of $n$ key points
$C(\cdot)$ The formula to calculate the curvature on a specific point of a curve
$N(\cdot)$ The neighborhood set of a reduced curve
$f$ A mean value of a reduced curve
$\sigma$ A distance which is specified by users
$x$ A point on a reduced curve
$G(\cdot)$ The formula to calculate the Gaussian-weighted value
$m(\cdot)$ A function to calculate mean curvature
$S(\cdot)$ A formula to calculate the saliency
$V$ A saliency map
$V'$ Normalization for saliency map $V$
$c$ A coefficient for non-linear normalization on the normalized saliency map
N  Numbers of clusters for classification
λ  The eigenvalue of the saliency curve
λ_sum  The sum of the eigenvalues of all saliency curves

ACKNOWLEDGEMENTS

This work is supported in part by the Ministry of Science and Technology at Taiwan, R.O.C., under the project grant number MOST104-2221-E-415-025.

REFERENCES


