EMBEDDED KEY-FRAME EXTRACTION FOR CG ANIMATION BY FRAME DECIMATION

Shiyu Li, Masahiro Okuda, Shin-ichi Takahashi

Faculty of Environmental Engineering, The University of Kitakyushu, Japan

ABSTRACT

This paper proposes a method for key-frame selection of captured motion data. In many cases, it is desirable to obtain a compact representation of the human motion. Key-framing is often used to express CG animation with a set of frames. In general, the animation is described by a set of curves that give the value of the rotation of all joints in each frame. Our method automatically detects the key-frames in captured motion data by using frame decimation. We decimate less important frames one by one, and then rank them by their importance. Our method has an embedded property, that is, all the frames are ranked by their importance, and thus users can specify any number of key-frames from one data set. We demonstrate the validity of our method in the experimental section by several typical motions such as walking and throwing.

1. INTRODUCTION

Motion capture systems have been widely used in CG amusement and human motion analysis such as games and athlete training. To reuse motion capture data, users need the motion library that stores existing motion data with human like character. Large motion databases do not accept the uncompressed forms, since the motion data are often huge. Moreover, motion data retrieval and transmission often require the compact representation of the motion [1]. For manipulation and rendering, a compact motion format is also required to reduce tasks. Key-framing is often used to express CG animation with a fraction of frames. Using interpolation and extrapolation techniques, general motion can be approximated by several key-frames. The accuracy of retrieved motion depends very much on the key-frame selection. In [3], Lim et al. treat the motion data as trajectory curves in a high-dimensional space and employ a curve simplification algorithm for key-posture extraction. This simplification method is also used in motion and emotion parameterization by M. Kraus [4]. Their method uses all joints of a human figure model which contains more than 20 joints. For each joint, the motion data include 3 angles of rotation about x, y and z coordinates respectively. Obviously, their curve simplification is time consuming when the total number of dimensions becomes greater than 60. To reduce its computational complexity, Etou, et al [2] use only five joints assuming that they are significant to characterize the motion. To achieve higher accuracy, we suggest a novel method which combines the characteristics of all joints and employ key-frame selection by iteratively decimating one point in each curve. Our method ranks all the frames by their importance, and thus specified number of key-frames can be obtained easily. This embedded property of our method leads into more accuracy and more feasibility than previous methods. According to the forward kinematics, in a motion chain, the transformation of a parent joint causes a change of its child joint position. The change of this child joint in turn affects its grand child joint, and so on. That is, the angles of parent joint are more important than its child joints. Thus we introduce a weight function for each joint for more precise key-frame selection.

The remainder of the paper is organized as follows. After a brief description about the motion data representation, we explain the algorithm of the generation of the integrate key-frames with characteristics of all curves in section 3. In the experimental section, some typical motions are employed to demonstrate the advantage of our approach compared with conventional methods. Finally, we draw a conclusion and refer to some future works.

2. PRELIMINARY

2.1. Motion Data

In CG application, a human figure is modeled by a hierarchical tree (Fig.1), in which the connections between two neighboring joints are rigid, for example, the joints of a shoulder and an elbow move, but the distance between them is not changed. In this framework, the motion data in the 3D space, *x, y, z*, are converted to a series of rotations. Fig. 1 shows a human figure with multiple joints and links. In different animations, the human figures with different numbers of joints are utilized.

2.2. Motion Curves

In our work, the animation is represented by a set of curves $g_j(k)$ of length *N*, where $j = 1, 2, \ldots, m$ are the indices of the curves and $k = 1, 2, \ldots, N$ are the frame numbers. These give the value, describing the rotation of all joints as well as the position of the joint of the root, represented as a function of the

0-7803-9332-5/05/$20.00 ©2005 IEEE

frame. Since for each joint the angle rotation about x, y, z axes respectively is adopted, and besides the rotation angles, for the root joint the position in 3D space is extra informed, the number of the joints is m/3-1.

We denote a set of the key-frames by

$$F^n = (f_1, f_2, ..., f_{n-1}, f_n),$$

where f_l, $l = 1,2,\Lambda\ n$ are frame indices of the key-frames. We treat the original curve as a set of N key-frames,

$$F^N = (f_1, f_2, ..., f_{N-1}, f_N),$$

that is, all the frames are key-frames. An example of the motion curves and their approximations are given in Fig.2.

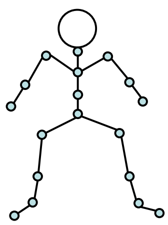


Fig.1 Human figure with multiple joints and links

3. CHARACTERISTIC COMBINED KEY-FRAME SELECTION ALGORITHM

3.1. Frame Decimation of Motion Curves

Suppose we start the decimation of vertices from the initial curves F^N. The decimating process is iteratively applied until only two frames, first and N-th frames, remain. When the k-tth frame f_k is decimated, it results in,

$$F^{N-1} = (f_1, f_2, ..., f_{k-1}, f_{k+1}, ..., f_N).$$

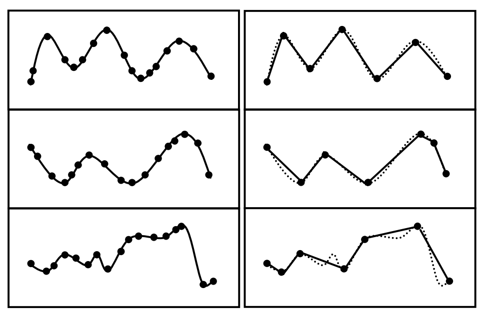
Meanwhile, the decimating algorithm generates a sequence of sets of frames, $F^N, F^{N-1}, ..., F^2$. We denote the k-th frame of F^n by $F^n(k)$, k=1, 2, 3,…, n, e.g., if current $F^n = (1, 5, 7, 15, 21, ..., N)$, then $F^n(3) = 7$.

To ensure that less important frames are decimated before more important ones, we need a cost function to measure the importance of the frames. The cost function $D^n(j,k)$, k=1, 2, …, n, for j-th curve (joint) in F^n, is defined by the difference of the distance between before and after the decimation of the k-th frame correspondingly. At the beginning of the algorithm $D^N(j,k)$ is calculated as follows. Assume that one of the frames $F^N(k)$ is being decimated. After the decimation, the previous $g_j\left(F^N(k-1)\right)$ and next key-frames $g_j\left(F^N(k+1)\right)$ are interpolated by a straight line. Thus the new values at the frame k are calculated by these two frames. The absolute difference between the original and the new values is set to $D^N(j,k)$. Then we define the total sum of $D^N(j,k)$ for all j as a cost function at the first iteration

$$D^N(k) = \sum_j D^N(j,k) \qquad (1)$$

For example of Fig. 3, the length of the arrow is assigned to $D^n(j,k)$. The frame with the least cost is considered least important and then is decimated.

After the decimation of the frame $F^n(k)$ in F^n at a iteration, we need to update the cost of neighbors of the frame, $F^n(k-1)$ ($= F^{n-1}(k-1)$) and $F^n(k+1)\,(= F^{n-1}(k))$. First, we assume that $F^{n-1}(k-1)$ would be decimated at the next iteration and then similarly $g_j\left(F^{n-1}(k-2)\right)$ and $g_j\left(F^{n-1}(k)\right)$ are interpolated by the straight line. This is illustrated in Fig.4(a) and (b). This procedure changes values at frames between $F^{n-1}(k-2)$ and $F^{n-1}(k)$. In Fig.4(a) the arrows illustrate the errors between original curves and the curve at the level n-1, which is denoted by $e(j, k-1, u)$, where $F^{n-1}(k-2) < u < F^{n-1}(k)$. The arrows of Fig.4(b) shows the errors $e'(j, k-1, u)$ after the decimation of $F^{n-1}(k-1)$.



(a) (b)

Fig.2 (a) Motion curves with 17 frames, from up to down, head, right should, left ankle joints (b) Approximations with 7 vertices after decimation of frames

Hence, for each joint, we calculate the difference of the sum of the error

$$D^{n-1}(j,k) = \sum_u e(j, k-1, u) - e'(j, k-1, u) \qquad (2)$$

where the range of the summation is $F^{n-1}(k-2) < u < F^{n-1}(k)$. The same procedure is done for $F^{n-1}(k)$ to calculate

$$D^{n-1}(j,k) = \sum_{u} e(j,k,u) - e'(j,k,u) \qquad (3)$$

where $F^{n-1}(k-1) < u < F^{n-1}(k+1)$.

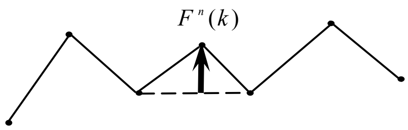


Fig.3 Cost of frame $F^{n}(k)$.

3.2. Generation of the Intergrative Key-frames

To generate the key-frames with the combined properties of all the motion curves of a whole body, we should pay attention to the characteristics of all joints in each motion chain. We first calculate the error $D^{n}(j,k)$ for each *j*th curve. Moreover, according to the hierarchical tree, the transformation of a parent joint causes a change of the position of its child joint. This change in turn affects its grand child joint location, and so on. Finally the changes are accumulated to the end- effector. Motion is inherited down the hierarchy from the terminator to the end-effector, which means the error of the lower nodes in the hierarchy affects the whole motion more than the higher nodes. Thus we introduce a weight function *W(j)*, *j*=1, 2,…*m*. Then assigning different *W(j)* for different *j*th curve, we can get the sum $E^{n}(k)$:

$$E^{n}(k) = \sum_{j} W(j) D(j,k)$$

We decimate the frame $F^{n}(k)$ with the minimum error $E^{n}(k)$.

In the end, our algorithm is stated as follow:

Step1. For all the frames F^{N}, calculate the cost function $D^{N}(k)$ in (1) and set $n = N$.

Step2. The frame k with minimum $D^{n}(k)$ is decimated and its index is removed from F^{n}. And then set $F^{n-1} = F^{n}$.

Step3. The cost function is updated by (2) and (3).

Step4. If the specified number of key-frames is remained, then stop. Otherwise go to Step.3.

4. EXPERIMENTS

We apply our key-frame selection algorithm to motion data. The motion is described by BVH format [4].

Here we show the results of two typical motions, “swagger”, “ballet”, “throw” and “kick”. Table.1 shows the description of these motions.

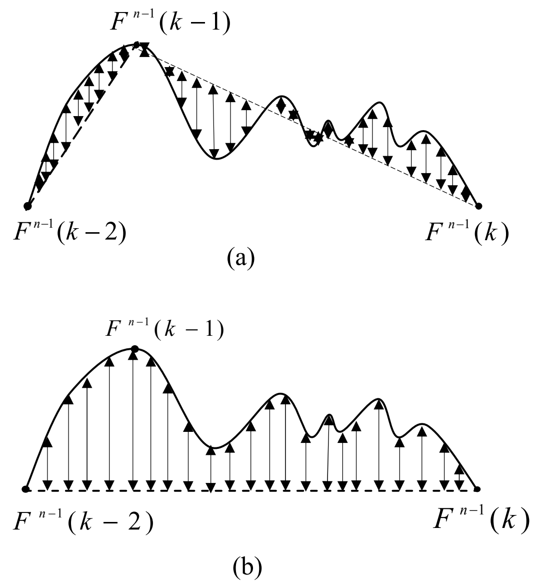


Fig.4 Solide and dotted lines are original curves and lines connected between key-frames, respectively, (a) level n-1, (b) curve after the decimation. The arrows represent (a) $e(j,k-1,u)$ and (b) $e'(j,k-1,u)$.

To evaluate the efficiency, we define the error as position differences of all the joints between the original motion and the retrieved one. To retrieve the motion from the selected key-frames, we adopt the piecewise cubic Hermite interpolation. The error of *j*-th joint is defined by

$$E_{j} = \frac{1}{N}\sum_{k=1}^{N}\left|g_{j}(k) - \hat{g}_{j}(k)\right|$$

and the total error of all joints in all frames by:

$$E = \frac{1}{m/3-1}\sum_{j=1}^{m/3-1} E_{j}$$

where $g_{j}(k)$ and $\hat{g}_{j}(k)$ are 3D rotation angles of the joint *i* in the frame *j* of original and retrieved motions respectively. N is the number of frames and $m/3-1$ is the number of total joints.

Fig.5 shows the difference of the error of angle between E^{p} generated by previous method in [3] and E^{o} generated by our algorithm when combining the characteristics of all joints of each motion. In this experiment, the number of the key-frames retained to retrieve the data is same in both methods. Moreover, to demonstrate the feasibility of our algorithm we also show the difference of error of position of throwing motion between the previous method and ours in Fig.6.

The error of the position in the 3D space of joint *i* between the original motion and the retrieved one by equation:

$$E_j = \frac{1}{N}\sum_{k=1}^{N}\left\|P_j^O(k) - P_j^r(k)\right\|$$

and these are summed for all the joints, where $P_j^O(k)$ and $P_j^r(k)$ are position of joint *i* in frame *j* of original and compressed motions respectively in world coordinate.

Table 1 Experimental results

	Data size	Number of joints	Sampling rate	Number of frames
Swagger	324k	23	0.00833s	580
Ballet	183k	20	0.0400s	388
Throw	70k	17	0.033333s	179
Kick	56k	19	0.033333s	147

5. CONCLUSION AND FUTURE WORK

We described a method for the key-frame selection of the captured human motion data with motion style in this paper. Due to the redundancy of huge database and discursion in motion index, a compact motion format is required. Key-framing based motion compression is considered a direct and effective compressing method to solve this problem. Treating animation as a set of curves which give the value of the rotation of all joints in each frame, our algorithm combines motion characteristics of all joints as a whole to detect a number of style-based key-frames. Moreover, the number of key-frames can be specified by user according to the motion characteristics. The embedded property of our method leads into more accuracy and more feasibility than conventional methods.

Finally, we have studied that considering motion characteristics, we have to pay more attention to some special joints such as the end effectors than other general types of joints. Now we are going to add some kinematics constraints to our method. The inverse kinematics algorithm has been successfully used in some other previous works of ours, and will be applied again into the proposed method introduced in this paper in our future work.

6. ACKNOWLEDGEMENT

This work was partly supported by a Grant-in-Aid for Young Sciences (#14750305) of Japan Society for the Promotion of Science, fund from MEXT via Kitakyushu innovative cluster project, and Kitakyushu IT Open Laboratory of National Institute of Information and Communications Technology (NiCT).

7. REFERENCES

[1] Armin Bruderlin and Lance Williams, "Motion signal processing," *Proceedings of SIGGRAPH 95. ComputerGraphics*, pp. 97—104, Aug. 1995.

[2] H. Etou, Y. Okada and K. Niijma, " Feature Preserving Motion Compression Based on Hierarchical Curve Simplification", CD-ROM Proc. of The 2004 IEEE International Conference on Multimedia and Expo (ICME2004), Taipei, Taiwan, June 2004.

[3] Ik Soo Lim, Daniel Thalmann, "Key-posture extraction out of human motion data", Proceeding of 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC2001), vol.2, pp. 1167-1169, 2001.

[4] Michal Kraus, "Human motion and emotion parameterization", Central European Seminar on Computer Graphics (CESCG2004).

[5] M. Meredith and S.Maddock, "Motion capture file formats explained," Department of Computer Science, University of Sheffielf.

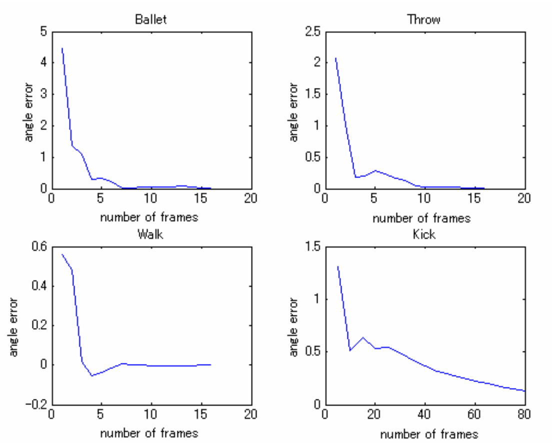


Fig 5. The difference of angle error E^p-E^o over the number of frames when combining the characteristics of all joints of each motion.

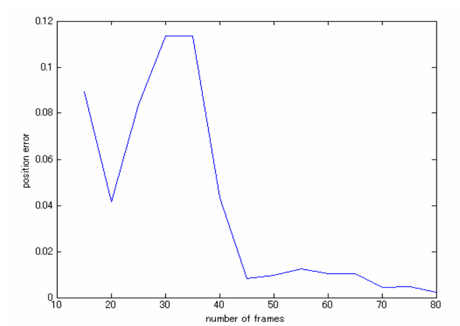


Fig 6. The difference of position error of throwing motion between previous method and our method over the number of frames