FEATURE PRESERVING MOTION COMPRESSION BASED ON
HIERARCHICAL CURVE SIMPLIFICATION

Hiroaki Etou, Yoshihiro Okada1,2, Koichi Niijima

1Graduate School of Information Science and Electrical Engineering, Kyushu University,
6-1, Kasuga-koen, Kasuga, Fukuoka, 816-8580, JAPAN,
{h-etou, okada, niijima}@i.kyushu-u.ac.jp
2Intelligent Cooperation and Control, PRESTO, JST

ABSTRACT
The authors have been studying on motion database systems. When entering an example motion as the query for the similarity search of motion data, it is natural way to enter it as a semantic primitive motion, i.e., "walk", "jump", "run" and so on. Mostly one motion data consists of several primitive motions. It is necessary to divide a composite motion into primitive motions. There are no algorithms able to automatically divide a composite motion into semantic primitive motions perfectly because the semantic meanings of primitive motions are strongly depending upon the human sense. A curve simplification algorithm is used for the key-posture extraction from motion data. This helps us to divide a composite motion into its primitive motions. The key-posture extraction is also used for the motion compression. In this paper, the authors propose new efficient key-posture extraction method that hierarchically applies the curve simplification algorithm to the feature joints of a human figure model.

1. INTRODUCTION
Recently the demand of human motion has been growing quickly in the movie production and the game industry. Even using CG animation systems, motion design is really laborious work and it needs much time. Motion capture systems have become available as commercial products and many motion data were created and stored using a motion capture system. It is better to create required motions by composing them from already existing primitive motions [6] [1] [4]. In this situation, we need a motion database management system that allows us to retrieve required primitive motions. Hence, we have been studying on motion database systems [2].

Generally each motion is composed from two or more semantic primitive motions. For example, Figure 1 shows a composite motion consisting of three primitive motions, i.e., "walk", "jump" and "walk". So, we have to divide such a composite motion into semantic primitive motions. Especially, when entering the query for motion database systems, the user enters it as a semantic primitive motion, e.g., "walk", "jump" and "run" motion. It is necessary to change a motion database into a collection of only semantic primitive motions. Osaki, et al [5] proposed an algorithm to divide a motion into its primitive motions for the motion recognition by matching those symbolized sequential values. However, generated primitive motions are too small segments and are not semantic primitives. Actually, there are no algorithms able to divide a composite motion into multiple semantic primitive motions accurately because the semantic meanings of primitive motions are strongly depending upon the human sense. We found a paper that treats a key-posture extraction method from human motion data based on a curve simplification algorithm [3]. This key-posture extraction method is able to divide a composite motion into multiple sub-motions. Lim’s method uses all joints angle data of a human figure model to be applied the curve simplification. In this paper, we propose that only a few joints are enough for the key-posture extraction by applying the curve simplification algorithm hierarchically to them. The key-posture extraction is also used for the motion compression. We experimented on several motions and obtained better results as compared with the results of Lim’s method.

The remainder of this paper is organized as follows. First of all, section 2 briefly describes the motion data format of our motion database. In section 3, we explain a curve simplification algorithm and our motion compression method. Section 4 shows experimental results. Finally we conclude the paper in section 5.
2. MOTION DATA

We use motion data, whose format is BVH because BVH format is very standard, originally proposed by BioVision Incorporated. Figure 2 shows the example of a BVH format file. BVH format data consists of two parts, HIERARCHY part and MOTION part. HIERARCHY part defines the hierarchy of the joints of a skeleton model and informs their initial positions. Usually, Hips is the root of the hierarchy, used as the center of mass of the skeleton model. The positions of other joints are specified by the relative position to their parent joint. OFFSET value, the set of x, y and z values, means such a relative position. Since the root joint does not have its parent, its OFFSET value is \{0, 0, 0\}. CHANNELS section specifies the number of available DOF of the corresponding joint. MOTION part describes DOF values, their types are specified by CHANNELS section, of each joint at each frame time. These values are also relative values to the parent joint.

HIERARCHY
ROOT Hips
{
OFFSET 0 0 0
CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation

JOINT LeftHip
{
OFFSET 3.43 0 0
CHANNELS 3 Zrotation Xrotation Yrotation

JOINT LeftKnee
{
OFFSET -18.47 0 0
CHANNELS 3 Zrotation Xrotation Yrotation

JOINT LeftAnkle
{
OFFSET -17.95 0 0
CHANNELS 3 Zrotation Xrotation Yrotation

End Site
{
OFFSET 0 -3.12 0
}
}

MOTION
Frames: 131
Frame Time: 0.033333

0.32 39.50 -0.80 3.63 0.48 -7.85 -3.21 ...
0.42 39.87 0.98 2.90 0.71 -7.99 -3.38 ...
...

Fig. 1. Motion "walk→jump→walk"

Fig. 2. BVH format

3. HIERARCHICAL CURVE SIMPLIFICATION

Given a curve as the chain of line segments, a curve simplification algorithm generates an approximation of a curve with a smaller number of vertices. Its concrete algorithm is as follows.

1. Two endpoints of a curve are connected by a straight line, (1) & (2) in Figure 3.

2. The point in the curve, which has the longest distance from the above line, is connected to each of the two above endpoints by new straight line respectively, (3) & (4) in Figure 3.

3. [2] is repeated until the user specified criterion is not satisfied, (5) & (6) .. in Figure 3.

Fig. 3. Curve simplification
Step-by-step visualization of the simplification.

One example criterion to stop the above process is the distance of a longest distant point from a current line. The user can specify it as a threshold value. The curve simplification algorithm does not depend upon the number of dimensions of a curve. This is an important feature of the curve simplification algorithm. Each motion data is represented as one \( N \) dimensions curve. One point on the curve means one posture in the motion. \( N \) is calculated by the following equation,

\[
N = LD
\]

(1)

Here, \( L \) is the number of joints of a human figure model, \( D \) is the number of dimensions of each joint.
Our motion data includes x-, y-, and z-angle values for each joint. Then \( D = 3 \). The method proposed in the paper [3] uses all joints of a human figure model. Our human figure model has 18 joints, so \( L = 18 \). Totally, \( N \) becomes 54. Indeed, one motion data is considered to be one \( N + 1 \) dimensional curve. One extra dimension is the time because motion data consists of time-sequential posture data. In the case of Lim’s method, \( N \) becomes 54 and the total number of dimensions becomes 55.

In this research, we consider the importance of joints. We propose the use of only five joints, i.e., two hands and two feet, and a root joint (the center of mass) to reduce the time consuming of the key-posture extraction process and to enhance the efficiency of the motion compression. We assume that these five joints seem to be significant to characterize the motion. We also propose the hierarchical use of the curve simplification algorithm. Our method applies the curve simplification by two steps. In the first step, for the key-posture extraction to coarsely divide a motion into its primitive motions, the curve simplification is applied to only the root joint (the center of mass). In the second step, for the key-postures extraction to finely divide the already generated primitive motions into their fine primitive motions, the curve simplification is applied to four significant joints, i.e., two hands and two feet. We experimented on several motions.

4. EXPERIMENTS AND COMPARISON

In this experiment, motion data was compressed using the above curve simplification algorithm. The compressed data consists of extracted key-postures. Uncompress process is to interpolate such key-postures. We used the linear interpolation for simplicity. For the evaluation of efficiency, we had to calculate the error between an original motion data and its uncompressed data. Furthermore, we compared the two errors of Lim’s method and our method. The error is calculated using the following equation,

\[
E = \frac{1}{M} \sum_{j=1}^{M} E_j
\]

This means the difference between the position of all joints for all postures included in an original motion and those included in its uncompressed motion. \( A \) and \( B \) mean an original motion and its uncompressed motion respectively. \( M \) is the total number of frames. \( E_j \) is the error between the posture of \( A \) and the posture of \( B \) at \( j \)-th frame. This is calculated by the following equation,

\[
E_j = \sqrt{\sum_{i=1}^{L} \left( (x_i^A - x_i^B)^2 + (y_i^A - y_i^B)^2 + (z_i^A - z_i^B)^2 \right)}
\]

Here, \( L \) is the number of joints. \( x_i^A, y_i^A \) and \( z_i^A \) are the x-, y- and z-coordinate value of the \( i \)-th joint of the posture at \( j \)-th frame of an original motion \( A \). \( x_i^B, y_i^B \) and \( z_i^B \) are the x-, y- and z-coordinate value of the \( i \)-th joint of the posture at \( j \)-th frame of the uncompressed motion \( B \) of \( A \). In Lim’s method, the curve simplification is applied to all joints, a \( (54 + 1) \) dimensional curve. In this experiment, we applied six times recursively and we obtained \( 64(2^6) \) divided sub-motions and 65 key-postures per each motion. In our method, the curve simplification is applied hierarchically by two steps. In the first step, the simplification algorithm is applied to a center of mass joint, a \( (3 + 1) \) dimensional curve. In this experiment, we applied \( n(<7) \) times recursively and we obtained \( 2^n \) divided sub-motions per each original motion. In the second step, the simplification algorithm is applied to four significant joints, a \( (12 + 1) \) dimensions curve generated from each of the above divided sub-motions. We applied \( 6 - n \) times recursively and consequently we obtained \( 64(2^n) \) primitive motions per each original motion. Each original motion was compressed into 65 key-postures.

Experimental results for 300 motion data is shown in Table 1. They are average error for 300 motions in the left row, time spent for the compression in the middle row, and joints and the number of times that the curve simplification algorithm was applied to the joints. Each COM and H&F means a center of mass joint, hands and feet. The SPEC of the machine used for experiments is Pentium4 2.6GHz CPU, 512MB memory.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
<th>Time(sec)</th>
<th>Joints and n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lim’s method</td>
<td>2.83</td>
<td>1.502</td>
<td>all joints: 6</td>
</tr>
<tr>
<td>Proposed method</td>
<td>3.14</td>
<td>0.125</td>
<td>COM:6, H&amp;F:0</td>
</tr>
<tr>
<td></td>
<td>2.84</td>
<td>0.266</td>
<td>COM:5, H&amp;F:1</td>
</tr>
<tr>
<td></td>
<td>2.77</td>
<td>0.298</td>
<td>COM:4, H&amp;F:2</td>
</tr>
<tr>
<td></td>
<td>2.68</td>
<td>0.472</td>
<td>COM:3, H&amp;F:3</td>
</tr>
<tr>
<td></td>
<td>3.02</td>
<td>0.486</td>
<td>COM:2, H&amp;F:4</td>
</tr>
<tr>
<td></td>
<td>3.23</td>
<td>0.639</td>
<td>COM:1, H&amp;F:5</td>
</tr>
<tr>
<td></td>
<td>4.34</td>
<td>0.698</td>
<td>COM:0, H&amp;F:6</td>
</tr>
</tbody>
</table>

Table 1. Experimental results

As for the accuracy, COM: 3 H&F: 3 case is the best. This means that it is effective to focus on both the center of mass and the hands and feet rather than only the center of mass or only the hands and feet. Of course, the result is dependent upon the motion database that
the user uses. For motion data that include gestures expressed by hands and feet movements, COM: 2 H&F: 4 and COM: 1 H&F: 5 would become better. In our method, the user can select which joints are important by varying $n$. The calculation time for the compression of our method is less than Lim’s method because the dimension of a curve represents a motion in our method is less than that of Lim’s method. The compressed motion (the set of extracted key-postures) in each of the above some cases generated from an original motion "walk→jump→walk" is shown in Figure 4.

![Lim’s method](image)

COM: 6 H&F: 0

COM: 5 H&F: 1

COM: 4 H&F: 2

COM: 3 H&F: 3

Proposed method

**Fig. 4.** Key-postures of motion "walk→jump→walk"

5. CONCLUSION

In this paper, we proposed hierarchical use of the curve simplification algorithm to extract key-postures from motion data for the help us to generate semantic primitive motions. This is considered to be the modification of the method Lim et al proposed in the paper[3]. This method is also available for the motion compression. Lim’s method applies the curve simplification algorithm to all joints of a human figure model, contrarily, our method applies it only the small number of joints to reduce the time consuming of the key-posture extraction process and to enhance the efficiency of the motion compression. We experimented on several motions and obtained better results as compared with the results of Lim’s method.

As future works, we will develop a system that helps us to generate semantic primitive motions from composite motions. Furthermore, we will develop a motion design system that allows us to generate required motions by composing them from such semantic primitive motions.

6. REFERENCES


