KEYFRAME REDUCTION TECHNIQUES FOR MOTION CAPTURE DATA*

Onur Önder[1], *Uğur Güdükbay*[1], *Bülent Özgüç*[1], *Tanju Erdem*[2], *Çiğdem Erdem*[2] *and Mehmet Özkan*[2]

[1]Bilkent University, Department of Computer Engineering, Bilkent, Ankara, Turkey
[2]Momentum DMT, TUBITAK MAM TEKSEB, Gebze, Kocaeli, Turkey

ABSTRACT

Two methods for keyframe reduction of motion capture data are presented. Keyframe reduction of motion capture data enables animators to easily edit motion data with smaller number of keyframes. One of the approaches achieves keyframe reduction and noise removal simultaneously by fitting a curve to the motion information using dynamic programming. The other approach uses curve simplification algorithms on the motion capture data until a predefined threshold of number of keyframes is reached. Although the error rate varies with different motions, the results show that curve fitting with dynamic programming performs as good as curve simplification methods.

Index Terms— Motion capture, keyframe reduction, curve fitting, curve simplification, noise filtering.

1. INTRODUCTION

Motion capture systems enable the animators to create realistic animations. These systems replicate the real world in a virtual environment. When a motion capture data is gathered, animators apply it on a virtual character. This step may produce two common problems, the action should be controlled by the animator, and the motion information should not contain any noise. This paper presents approaches to solve these problems by using two different techniques: curve fitting with dynamic programming and curve simplification.

Motion capturing is a costly process where the hardware and setting requirements play an important role. Thus there are motion capture data libraries where some common motions (such as running, jumping, walking, etc.) are available to the animators. Animators can use those predefined data for custom animations. If the animator needs to change any part of the motion, he/she would need to modify joint information on every frame. This process takes a lot of time since many joint information has to be modified for every frame of the animation. If a small number of keyframes could be used to represent the motion capture data, then the animator would have an easier control over the motion.

*This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV.

Keyframes can be selected to represent the motion capture data by curve fitting, where keyframes are located and modified as necessary using a dynamic programming approach. Keyframes can also be selected by curve simplification, where they are found by simplifying the motion curves.

The rest of this paper is organized as follows: Section 2 gives background information and previous works, Section 3 describes two different approaches for the given problems, Section 4 states the results of the approaches and compares them, and Section 5 discusses the conclusions.

2. BACKGROUND

There are some approaches for finding keyframes of the motion capture data. In [1], Terra and Metroyer proposed a solution to find the timings of the keyframes. They have used a performance based approach. Huang et al. propose an iterative approach, called key-probing, for keyframe extraction [2]. Also in [3], Park defined a method to extract key-postures from a motion.

Curve fitting algorithms are available in various topics. Using these algorithms, animation control is covered in [4, 5]. There is no generic research that uses dynamic programming to optimize the curve fitting process.

Curve simplification algorithms are generally used in motion summarization. The main algorithm is defined by Lowe [6] that is used in many other approaches [7, 8].

A curve fitting with dynamic programming approach is initially defined in [9]. This paper extends this study, presents its results and compares it with a curve simplification algorithm.

3. CONTRIBUTIONS

This section defines two different approaches to the keyframe reduction problem for motion capture data: curve fitting with dynamic programming and curve simplification.

3.1. Curve fitting with dynamic programming

In this approach, a Hermite curve is fit onto the motion graph of the joints in the motion capture data. Initially, some keyframes are predicted on the motion data. Then these keyframes

978-1-4244-1755-1/08/$25.00 ©2008 IEEE

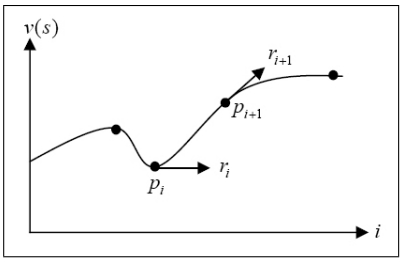

Fig. 1. Hermite curve is defined by its control points and tangent vectors.

are moved around their initial positions to find the best possible path that results in the minimum error value. Error value is calculated as the mean square of the difference between the found motion curve and the initial motion capture data curve. Two successive keyframes and their tangents are shown in Figure 1.

A Hermite curve can be represented as

$$v(s) = \begin{bmatrix} (s-i)^3 & (s-i)^2 & (s-i) & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_i \\ p_{i+1} \\ r_i \\ r_{i+1} \end{bmatrix}$$

where p_k represent the positions and r_k represent the tangents at keyframe k.

The algorithm runs as follows:

1. Find initial keyframe estimates, by predicting p_i, p_{i+1}, r_i and r_{i+1} for every i, choosing the frames where the first or second derivatives of the motion is zero.
2. Determine search spaces around the keyframes that are predicted. Nine possible positions in the search space is checked for every keyframe.
3. Compute the mean square error for each possible combination in the search space for the keyframes. For finding the value of the curve at certain point, the curve is converted into a Bézier curve and a Bézier subdivision algorithm is used until the required resolution is reached. For each combination of p_{i+1} and r_{i+1}, the best values of p_i and r_i are saved.
4. On the last segment of the curve, find the total mean squared error for each combination of p_{i+1} and r_{i+1} by traversing the curve backwards while using the minimum saved values.
5. Set the new curve resulting from the minimum cost path of keyframes that is calculated in step 4.
6. Assume that the values p_i, p_{i+1}, r_i and r_{i+1} for all i, obtained in step 5 are the updated initial values and repeat steps 2-6 until a predetermined accuracy in the overall cost is reached.

3.2. Curve simplification

The curve simplification algorithm generally uses the basics of Lowe's algorithm [6]. Initially, the motion will have two keyframes, one at the beginning and one at the end. Then the frame that produces the highest error value will be turned into a keyframe, and the algorithm will continue iteratively.

The algorithm is defined as follows:

1. Set the first and the last frames as keyframes, creating 2 keyframes.
2. Find and store the minimum distance between the line, defined by the 2 keyframes, and the value of the curve for every frame. Find the highest distance point, which would have the distance called as the error distance.
3. If the ratio of error distance to the length of the line between 2 keyframes is higher than a specified threshold, stop further subdividing this interval. Otherwise, create a new keyframe at the point with the highest error distance.
4. Subdivide the current state into two smaller segments, a segment between the beginning keyframe and the newly created middle keyframe, and another segment between the middle keyframe and the ending keyframe. Assume that the new segments now have only 2 keyframes and restart the algorithm from step 2 for both segments.

A sample figure showing the steps of the algorithm is presented in Figure 2.

4. RESULTS

The two proposed algorithms are used to simplify two different motion capture data. The first motion is a running motion with 22 frames. The second motion is a drinking tea motion with 1409 frames.

By using curve fitting with dynamic programming, the first motion is simplified into 8 keyframes per channel on the average from the initial 22 frames. For the second motion, the simplified motion resulted in 459 keyframes per channel on the average from the initial 1409 frames. The motion graphs for a sample joint is presented in Figure 3 for both the original and the simplified motion.

While using the curve simplification method, the number of average keyframes per channel is shown in Table 1 for some different threshold values.

Table 2 compares the number of keyframes in the resulting motion data in two algorithms.

The previous approaches can achieve up to 80% decrease in the size of animations, while the new algorithms described in this paper can achieve similar results with decrease percent around 75-80%. The curve simplification method provides better results both in terms of the error ratio and the number of keyframes.

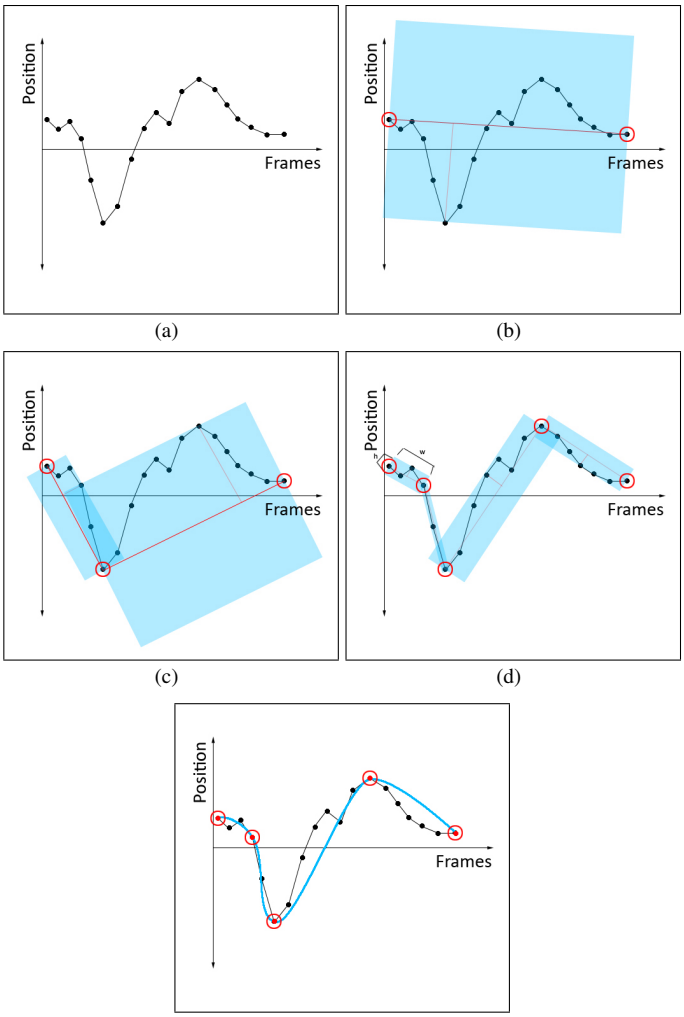

Fig. 2. The application of the curve simplification algorithm: (a) the initial curve; (b) the first step; (c) the second step; (d) the third step; (e) the final result.

The dynamic programming approach cannot be used over and over again since the found keyframes are the optimal ones. Also the total error is directly related to the initial predicted keyframes. On the other hand, the number of keyframes found in curve simplification method can be reduced by changing the threshold value or applying the algorithm multiple times. The error value would be expected to change depending on the threshold value. The error ratio is inversely proportional to the frame reduction ratio. Both approaches eliminate the noise factor since a curve is fitted on the motion data.

Considering the subjective evaluations, visual results obtained using the curve simplification method is slightly better than its counterpart. Dynamic programming approach's inability to decrease the error value after some point limits its usage. Sample frames from both motions can be seen in Figures 4 and 5.

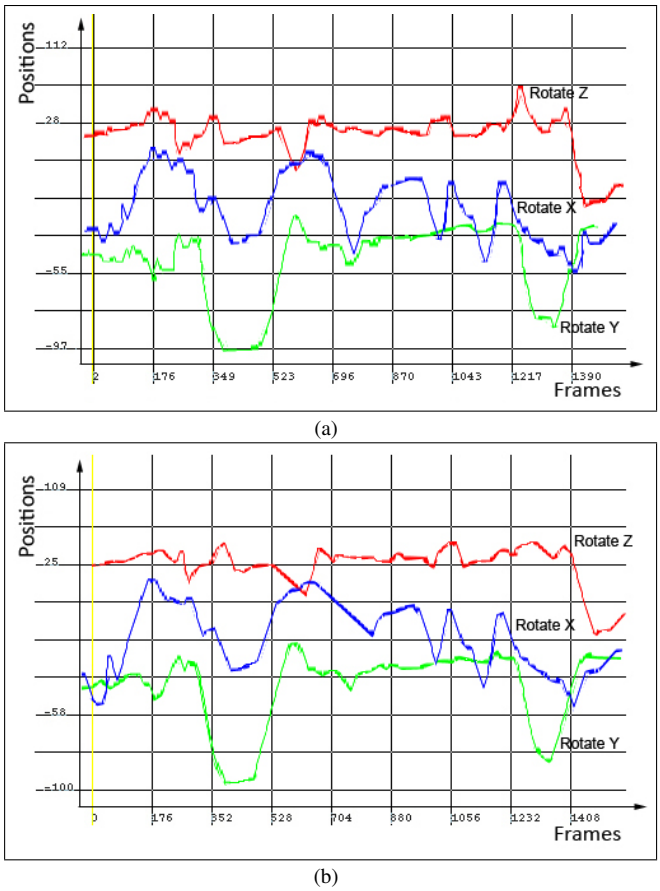

Fig. 3. Sample joint rotation graph for drinking tea motion: (a) original; (b) simplified.

5. CONCLUSION

This paper presented two different and new approaches for keyframe reduction and filtering of motion capture data. The algorithms help the animator to easily edit and modify the motion capture data by using the keyframes only. They solve the defined problems as well as having some disadvantages. Since the motion capture data is represented with smaller number of keyframes, the error value will never be zero in a real motion capture data.

The advantages of the proposed methods include the benefit of less storage space and less noisy data. If the algorithm thresholds are not defined properly, an over smoothed motion may be produced.

Although dynamic programming with curve fitting was never used before, it produces promising results. Curve simplification approach produces slightly better results (less number of keyframes) than the dynamic programming approach for comparable motion quality. Both methods are successful on decreasing the number of keyframes for a motion capture animation and filtering any jitter that may be produced by the motion capture system.

Table 1. Results of number of average keyframes per channel on several runs with Curve Simplification algorithm.

Threshold	Curve Simplification Running	Drinking
0.001	608	20
0.005	246	16
0.010	85	13
0.020	11	8
0.030	3	6
0.100	2	2

Table 2. Summary of the two approaches based on empirical results. Threshold is selected to have similar visual quality between two algorithms.

	Curve Fitting & Dyn. Prog.		Curve Simplification	
Properties	Running	Drinking	Running	Drinking
# of frames	22	1409	22	1409
Channels	84	84	84	84
# of Keyframes	8	459	6	246
Reduction Ratio	36 %	32 %	27 %	17 %

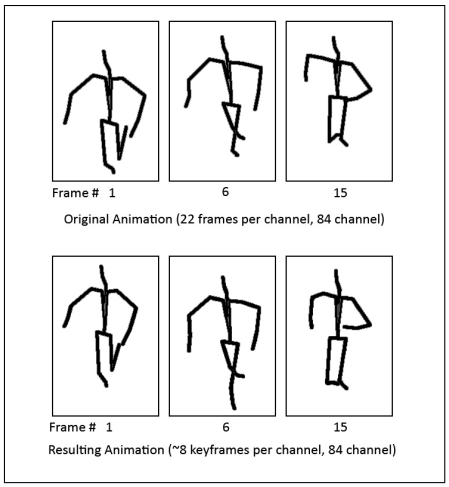

Fig. 4. Sample frames from original running motion and the resulting running motion.

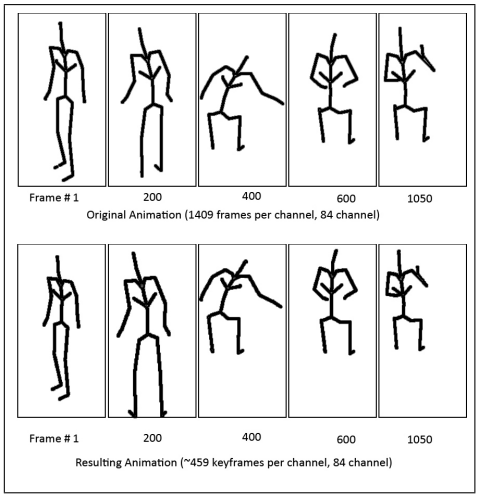

Fig. 5. Sample frames from original drinking tea motion and the resulting drinking tea motion.

6. REFERENCES

[1] S. C. L. Terra and R. A. Metroyer, "Performance timing for keyframe animation," *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2004.

[2] Y. Hsu K. Huang, C. Chang and S. Yang, "Key probe: A technique for animation keyframe extraction," *The Visual Computer*, vol. 21, no. 8-10, pp. 532–541, September 2005.

[3] M. J. Park and S. Y. Shin, "Example-based motion cloning," *Computer Animation and Virtual Worlds*, vol. 15, 2004.

[4] P. J. Schneider, "An algorithm for automatically fitting digitized curves," *Graphics Gems, Academic Press, San Diego, CA*, 1990.

[5] S. S. Snibbe, "A direct manipulation interface for 3D computer animation," *Computer Graphics Forum*, vol. 14, no. 3, pp. 271–283, 1995.

[6] D. G. Lowe, "Three-dimensional object recognition from single two dimensional images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, March 1987.

[7] M. Kraus, "Human motion and emotion parameterization," *Proceedings of the Conference of the Central European Seminar on Computer Graphics*, 2004.

[8] L. S. Lim and D. Thalmann, "Key-posture extraction out of human motion data by curve simplification," *Proceedings of the 23rd Annual EMBS International Conference, Istanbul, Turkey*, 2001.

[9] T. Erdem U. Gudukbay O. Onder, C. Erdem and B. Ozguc, "Combined filtering and key-frame reduction of motion capture data with application to 3dtv," *WSCG Poster Proceedings, Plzen, Czech Republic*, 2006.