한국시뮬레이션학회 '98추계학술대회 논문집 1998.11.7 청주대학교

모션 캡처 데이타 향상 기법

°최광진, 박상현, 고형석

서울대학교 휴먼 애니메이션 연구단

Enhancing Motion Capture Data

°Kwangjin Choi, Sanghyun Park, Hyeongseok Ko

Human Animation Center, Seoul National University

Abstract

In animating an articulated entity with motion capture data, especially when the reconstruction is based on forward kinematics, there could be large discrepancies at the end effector. The small errors in joint angles tend to be amplified as the forward kinematics positioning progresses toward the end effector. In this paper, we present an algorithm that enhances the motion capture data to reduce positional errors at the end effector. The process is optimized so that the characteristics of the original joint angle data is preserved in the resulting motion. The frames at which the end-effector position needs to be accurate are designated as "keyframes" (e.g. starting and ending frames). In the algorithm, corrections by inverse kinematics are performed at sparse keyframes and they are interpolated with a cubic spline which produces a curve best approximating the measured joint angles. The experiment proves that our algorithm is a valuable tool to improve measured motion especially when end-effector trajectory contains a special goal.

1. Introduction

Animating an articulated entity requires the technique to control highly redundant degrees of freedom[1,2,5,7,8,9,11,20]. To obtain realistic motion of complex character, copying is more effective than computational synthesis, and recently, motion capturing is emerging as a powerful technique for producing realistic animation of complex articulated characters. However, the motion capture technique by itself has very poor generality and automatism. Thus it is best utilized in the application in which highly realistic motion is required but is never used again, as in movies. The new ideas such as motion composition, warping[12], blending[4], motion signal processing[3] and motion editing with spacetime constraints[6] contributed in overcoming the above limitations.

Another fundamental problem of motion capturing is that the measurement errors and noises greatly demotes the value of captured data. Without an elaborate manual processing of the captured data, the resulting animation often looks shaky and unrealistic. Furthermore, the errors in joint angle data tend to be amplified and accumulated during the forward kinematic reconstruction. Therefore the resulting end-effector position might have a large discrepancy when compared with the actual performance.

In this paper, we presents an algorithm that automatically processes the motion capture data to produce another data set that promises more accurate and smooth animation.

2. Overview

Whether it is directly or indirectly collected, the joint angles contain important features of a motion, and the forward kinematic reconstruction

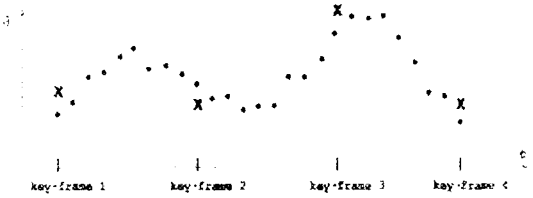

Figure 1: conflicts between keyframes and joint angle measurements

is quite effective in restoring such features. When a motion is reconstructed from joint angle measurements only, however, the end-effector position in the resulting animation can be different from the original motion due mainly to joint angle errors and inaccurate link lengths. In some motion, the pattern of end-effector movement contains important characteristics. Therefore animators may want the end-effector to follow the measured trajectory, or at least agree with the trajectory at several keyframes. In this paper, we develop an algorithm that is faithful to both joint angle and end-effector data.

Figure 1 shows the situation. The dotted points in the graph represent the measured joint angles. The points marked with X are the joint angles suggested by the keyframes. If a joint angle curve satisfies the following two conditions,

1. the curve should pass the joint angles suggested from the keyframes

2. at the same time, the curve should be a least square fit of the measured joint angles.

The end-effector position will agree with the measured end-effector trajectory at the keyframes, and at the same time, the curve will preserve the pattern of the measured joint angles in an optimal sense.

This paper presents an algorithm to find the curve that satisfies the above two conditions, and it involves two major steps as shown in Figure 2. The first step is to solve inverse kinematics at the keyframes, which will produce the joint angles at the keyframes. The second step is to interpolate the joint angles at keyframes and at the same time perform a least square fit on the measured joint angles.

3. Inverse Kinematics

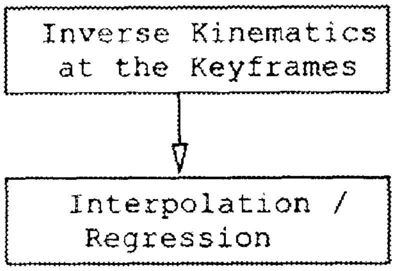

Figure 2: the two major steps of our algorithm

Even though there exist measurement errors in both position and angle data, the position data is much more dependable since the error is not amplified nor accumulated. The angle data is relative, while the position data is absolute. In our motion capture data enhancement, the position data of the end-effector is utilized in inverse kinematics to correct the angle data so that the amount of accumulated error is limited by the errors in the position data.

An inverse kinematics problem can be solved by minimizing a nonlinear objective function [13]. Beside of minimizing end-effector position error, we have to maintain the original joint angle pattern. i.e. we need to impose that the joint angles should resemble the measurements. Consequently, we obtain the objective function as follows.

$$G(\theta_1,\theta_2,\ldots,\theta_n) = w_1 \cdot \|\vec{e} - \vec{e}'\|^2 + (1-w_1) \cdot \sum_i^n (\theta_i - \theta_i')^2 \qquad (1)$$

G shown above consists of two error terms: the end-effector position error and the sum of joint angle errors, where $\vec{e}$ is a function of the joint angles $(\theta_1,\theta_2,\ldots,\theta_n)$ that give the end-effector position, $\vec{e}'$ and θ_i' are the measured end-effector position and the measured joint angles respectively. With a large ω_1, the first term in (1) can be interpreted as a soft constraints. Provided that the number of degree of freedom of the model is large enough, as is true for human models, the constraint can be easily met in a few iterations.

4. Interpolation/Regression

We interpolate the keyframe points coming from inverse kinematics with a smooth piecewise cubic spline curve. By steering the derivatives at the extreme points of the whole curve, we can control the interpolating curve so that it also optimally approximates the original joint angle data.

We call the fixed points . $P_1, P_2, \cdots, P_n$[1)]

---

[1] In the algorithm, the control points correspond to the joint angle values obtained by inverse kinematics at the

The two derivatives at the ends P_1' and P_n', and the controllability of the two end derivatives is illustrated in Figure 3.

To apply regression, the cubic spline curve P(t) had to be expressed in a linear combination of P_1', P_n', and other known parameters. We could obtain following equation after proper manipulation of the cubic spline equations [10].

$$P(t) = f_1(t) + f_2(t) \cdot P_1' + f_3(t) \cdot P_n' \qquad (2)$$

In (2), the terms in $f_1(t)$, $f_2(t)$, and $f_3(t)$ are completely determined by $P_1, P_2, \ldots, P_n$. Now, we have to choose P_1' and P_n' that make the interpolating curve be also a least square fit of the measured joint angle data, $m(t)$. The least square solution is given by

$$\begin{bmatrix} P_1' \\ P_n' \end{bmatrix} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{m} - \mathbf{f}_1), \qquad \mathbf{X} = [\, \mathbf{f}_2 \ \mathbf{f}_3]. \qquad (3)$$

The bold symbols in (3) represent matrices or column vectors. That is, the i-th elements of vector **f1**, **f2**, **f3**, and **m** are the values of $f_1(t_i)$, $f2(t_i)$, $f3(t_i)$, and $m(t_i)$ repectively.

5. Result

Our algorithm was implemented on Silicon Graphics workstations. In this experiment, a subject was asked to swing his arms and make his fists meet three times as shown in Figure 4, and the motion was reproduced by (1) the forward kinematic reconstruction and (2) by our algorithm, and these two results were compared. Eleven 6DOF sensors were placed on the head, chest, waist, shoulders, upper arms, lower arms, and hands. The sensors placed at the back of the hands were used to extract the positional information, and the other sensors were used to extract the joint angle information. The captured position of the subject's hand was mapped to the palm center of the animated figure. Therefore in Figure 6, the distance between the small cube and palm center can be interpreted as an approximation of the end-effector error.

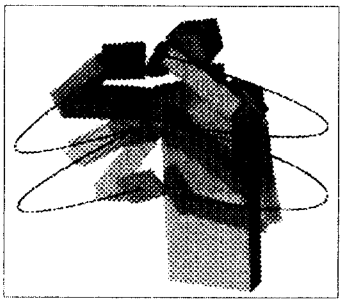

Figure 4: Fist matching motion

As expected from the previous discussion, the two fists didn't match in the forward kinematic reconstruction due to various kinds of errors. Figure 6 (a) shows the top view of a snapshot during the motion reconstructed by forward kinematics. The small boxes near the hands represent the measured position of the end-effectors. The black curve in the figure represents end-effector trajectory acquired by forward kinematic reconstruction and the grey curve shows directly measured end-effector positions. When the keyframes are corrected by inverse kinematics, and joint angle interpolation/regression is done subsequently, we could get a smooth motion which showed exact fist matches at the keyframes. The resulting end-effector trajectory is compared with the directly measured trajectory in Figure 6 (b). In the above analysis we can see that our algorithm effectively reduces the end-effector errors and preserves the joint motion pattern as well.

6. Conclusion

In this paper, we proposed an algorithm for reducing the end-effector errors in motion capture data. A few points (keyframes) along the measured end-effector trajectory were selected, and inverse kinematics was applied to find the joint angles at those keyframes. A smooth curve that interpolates the above joint angles was constructed so that it also gives the best approximation to the measured joint angles. Our method can be useful to people in character animation and game industries. It will save animators from laborious manual editing of motion capture data especially when the end-effector trajectory contains a special goal.

Acknowledgment

This paper is supported by Creative Research Initiatives of the Korean Ministry of Science and Technology.

Bibliography

[1] S. Rosenthal B. Bodenheimer, C. Rose and J. Pella.. The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97*, pages 3-18, 1997.

[2] N. I. Badler, B. Brian, and D. Zeltzer. *Making Them Move: Mechanics, Control, and Animation of Articulated Figures.* Morgan Kaufmann

---

keyframes.

Publishers, 1990.

[3] Armin Bruderlin and Lance Williams. Motion signal processing. In Robert Cook, editor, *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 97-104, August 1995. ACM-0-89791-701-4.

[4] Bobby Bodenheimer Charles Rose, Brian Guenter and Michael F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 147-154, August 1996. ACM-0-89791-746-4.

[5] Baker D. and Wampler C. On the inverse kinematics of redundant manipulators. *The International Journal of Robotics Research*, 7(2):3-21, March/April 1988.

[6] Michael Gleisher. Motion editing with spacetime constraints. In *1997 Symposium on Interactive 3D Graphics*, 1997.

[7] Cary B. Phillips. *Interactive Postural Control Of Articulated Geometric Figures.* PhD thesis, University of Pennsylvania, 1991. MS-CIS-91-82.

[8] Cary B. Phillips and Norman I. Badler. Interactive behaviors for bipedal articulated figures. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 359-362, July 1991.

[9] David R. Pratt, Paul T. Barham, John Locke, SARCOS Inc; Micheal Hollick Micheal J. Zyda, Naval Postgraduate School; Bryant Eastman, John Granieri, Hyeongseok Ko, and University of Pennsylvania Norman I. Badler. Insertion of an articulated human into a networked virtual environment. In *Distributed Interactive Simulation Environments Conference*, University of Florida, December 1994.

[10] David F. Rogers and J. Alan Adams. *Mathematical Elements for Computer Graphics.* McGraw-HILL International Editions, second edition, 1990.

[11] Douglas J. Wiley and James K. Hahn. Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications*, pages 39-45, November/December 1997.

[12] Andrew Witkin and Zoran Popovic. Motion warping. In Robert Cook, editor, *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 105-108, August 1995. ACM-089791-701-4.

[13] Jianmin Zhao and Badler N. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13(4):313-336, October 1994.