Journal of Korea Multimedia Society Vol. 11, No. 6, JUNE 2008(pp. 869-874)

3D Animation Authoring Tool Based On Whole Body IK and Motion Editing

Woosuk Ju[†], Choongjae Im[††]

ABSTRACT

The work of creating character motion needs the higher professional technology and sense and the creating work of realistic and natural motion possess the most part of production term. In this paper we introduce the easy and convenient 3D animation authoring tool which makes the motion based on whole-body inverse kinematics and motion editing function. The proposed 3D animation authoring tool uses the forward kinematics using quaternion and whole-body inverse kinematics to determine the rotation and displacement of skeleton. Also, it provides the motion editing function using multi-level B-spline with quasi-interpolant. By using the proposed tool, we can make 3D animation easily and conveniently.

Key words: whole-body inverse kinematics, hierarchical structure, animation, motion editing, signal processing, multi-level B-spline

1. INTRODUCTION

Recently, the animation industry has been growing as higher value-added business, so the research of multi-joint character animation is accomplished briskly. The work of creating character motion needs the higher professional technology and sense and the creating work of realistic and natural motion possess the most part of production term. To solve this problem, motion capture system is introduced. But it needs the much money and professional actors. Therefore the anyone who needs his own animation cannot use this wonderful system easily. Also the captured motions must pass through the retouching process using animation production tools. But, also, these have the difficulty to use because of complex functions and interfaces.

※ Corresponding Author : Choongjae Im, Address : (705-701) Department of Game & Mobile Contents, Keimyung University, Deagu, Korea, TEL : +82-53-620-2170, FAX : +82-53-620-2175, E-mail : dooly@gw.kmu.ac.kr
Receipt date : Apr. 31, 2008, Approval date : June 10, 2008
† Dept. of Game, Dongseo University
(E-mail : savrang@dongseo.ac.kr)
†† Dept. of Game & Mobile Contents, Keimyung University

The proposed tool is the easy and convenient but professional animation tool, which can create the multi-joint character motion and edit the existing motions. We make it to interconvert with 3D MAX. Its motion creation and editing use many algorithms, for example, forward kinematics, inverse kinematics, multi-level B-spline, signal processing and offer the fast and instinctive interface.

This paper consists as follows. Chapter 2 show the work process of this tool using plug-in with 3D MAX[1] and its file format and introduce the algorithms for motion creation such as forward kinematics and whole-body inverse kinematics. The motion editing algorithm using multi-level B-spline, is introduced in chapter 3. The conclusion and further studies is in chapter 4.

2. MOTION CREATION PROCESS

The goal of the animation authoring tool is to create animation motions of 3D object modeled using 3D rendering tools, for example, 3D MAX, MAYA etc, more easily and instinctively. The animation data created by the animation authoring tool are imported to 3D rendering tools and then

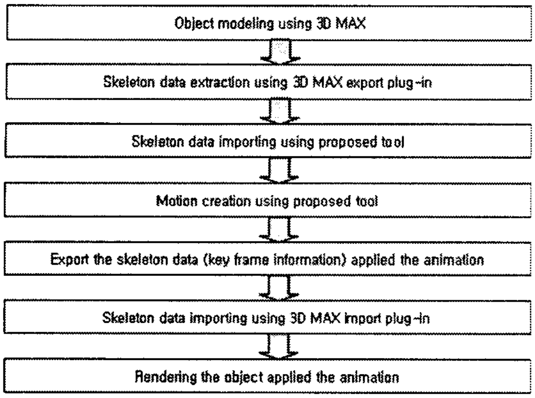

Fig. 1. Working process using proposed tool

we can get the final results. In this case we must unify the data formats between 3D rendering tools and proposed tool. The working process for motion creation using proposed tool is like Fig. 1.

2.1 Skeleton data extraction using 3D MAX export plug-in

3D MAX export plug-in is the one of program module working under 3D MAX environment. 3D MAX plug-in can not only access all information of the modeled object but also it can amend. Any of the motion creation tools does not make the 3D objects but give the motion to 3D object modeled by the 3D rendering tools. To give motion to 3D object, first of all, we must model the 3D object in 3D MAX.

The skeleton in Fig. 2 is the framework composed of bones. For example, in the case of human, it is the aggregate of bone structure formed human's body. Mesh is the aggregate of vertices. In the case of human, it is same as skins. Every part of meshes is connected with arbitrary bone which forms skeleton, and if the bone has the movement then mesh connected this moves along bone at once.[2]

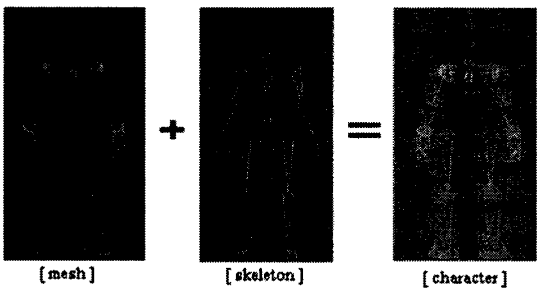

Fig. 2. 3D character modeling

The object modeled like this must be stored in the form of usable data format, like .ikf or .ikm, in proposed tool using 3D MAX export plug-in. The proposed tool needs skeleton data file (.ikf) and mesh data file (.ikm) basically.

Fig. 3 is the structure of skeleton data file. SKELETON HEADER can store the basic information about skeleton data, which are file version information, the number of bones formed skeleton and world coordinate information of skeleton. BONE HEADER can store the basic information about bone formed skeleton, which are index of parent bone, index of child bone, transformation matrix deciding the early position and local coordinate information of bone. CHILD BONE INDEX LIST can store the child bones of applied bone and the already created animation information is stored in ANIMATION KEY FRAME INFORMATION, which are the time that the key frame is established, displacement value of bone in applied key frame and rotation value of bone in applied key frame. These are usable to interpolate key frames.

The proposed tool provides the interface to represent skeleton data from the skeleton data file (.ikf) as Fig. 4.

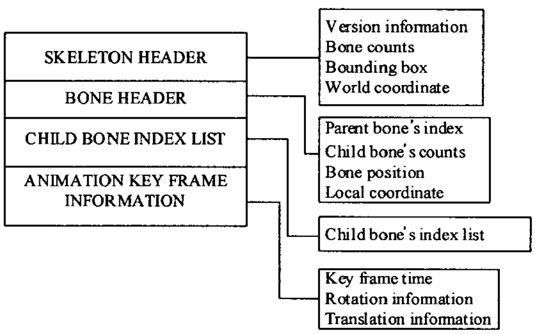

Fig. 3. Skeleton data file format (.ikf)

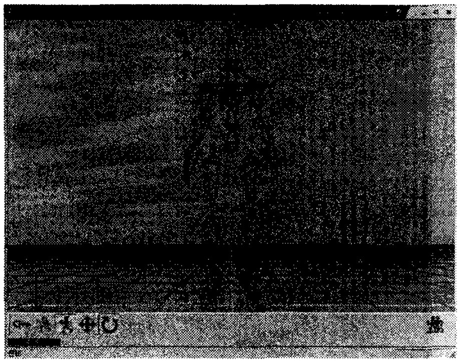

Fig. 4. Skeleton data interface

Fig. 5 is the structure of mesh data file. MESH COUNT store the number of meshes in the case that object is constituted with several meshes. LINKED BONE INFORMATION stores the information about bones connected with applied mesh and MESH INFORMATION stores the information about vertices and faces formed applied mesh.

The proposed tool provides the interface to represent mesh data from the mesh data file (.ikm) as Fig. 6.

2.2 Hierarchical structure of skeleton data

Every bones constituted skeleton forms hierarchical structure. There are two ways to decide the bone position in world space. One is the absolute coordinate system and the other is the relative coordinate system. Each bone has one transformation matrix, which express the transformation information. In the case of absolute coordinate system, it expresses the transformation

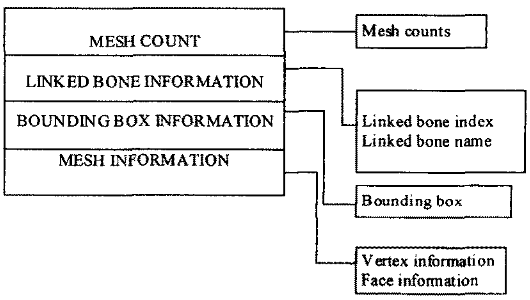

Fig. 5. Mesh data file format (.ikm)

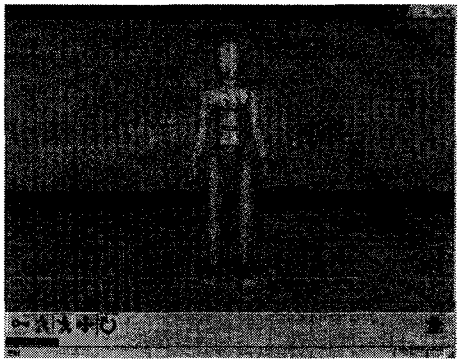

Fig. 6. Mesh data interface

information as the reference point is the origin. In the case of relative coordinate system, it expresses the transformation information from parent bone. Our system uses the relative coordinate system and the transformation matrix of the applied bone is multiplied to the transformation matrix of parent bone to calculate the final position in parent-child hierarchical structure.

Fig. 7 shows the hierarchical structure of skeleton. For example, to calculate the final position of S_L_Foot1 bone, the transformation matrices of F_BODY bone, S_L_Leg1 bone, R_L_Leg2 bone and S_L_Foot1 bone. [3]

2.3 Forward kinematics using quaternion

Our system uses the quaternion to determine the rotation and displacement. This prevents the gimbal lock phenomenon. The motion data is expressed by

$$m(t) = (q(t), q^1(t), \cdots, q^n(t))^T \qquad (1)$$

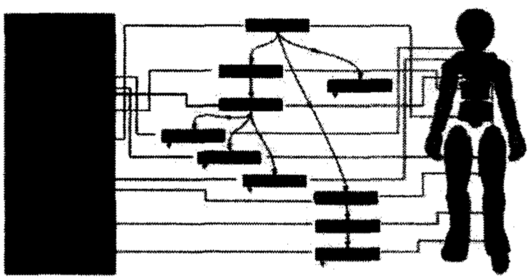

Fig. 7. Hierarchical structure of skeleton data

Here $p(t) \in R^3$ and $q^1(t) \in S^3$ is the displacement and rotation of root and $q^i \in S^3$ is the rotation information of each $(i-1)$ joint and n is the number of all joints. [4]

2.4 Whole-body inverse kinematics

We combine the jacobian transpose method [5] and analytic method for linkage with 3 joints of [6]. According to the least-square means,

$$\begin{pmatrix} v_0^n \\ w_0^n \end{pmatrix} = \begin{pmatrix} J_v \\ J_w \end{pmatrix} \Delta\theta \tag{2}$$

is solved by the Moore-Penrose inverse matrix

$$J^+ = \begin{cases} (J^T J)^{-1} J^T & 6 > n = r \\ J^T (J J^T)^{-1} & r = 6 < n \end{cases} \tag{3}$$

Here is the number of joints and is the rank of jacobian matrix. But this solution is also numerical solution, so the errors happen in the course of nature. Therefore, to reduce the time-complexity, we use the jacobian transpose method. And we use the analytic method of [6] for limb linkage to reduce the errors. The explicit form is

$$\phi = \cos^{-1}\left(\frac{l_1^2 + l_2^2 + 2\sqrt{l_1^2 - r_1^2}\sqrt{l_2^2 - r_2^2} - L^2}{2 r_2 r_2}\right) \tag{4}$$

l_1 = the length of the upper linkage,
l_2 = the length of the lower linkage,
r_1 = the distance from the middle one rotation axis to the upper one,
r_2 = the distance from the middle one rotation axis to the lower one,
L= the distance between the upper on and the lower one.

From this, we reduce the DOFs form 37 to 13 DOF. For the remaining DOFs, we can use the pseudo-inverse jacobian method using the above angular velocity and its explicit form is $dX_k = Jd\theta_k$. Then we can find the $d\theta$ -term by using the pseudo-inverse jacobian and it is same notion of the minimality for displacement mapping. From this we can find the quaternion term according to relation between quaternion and axis-angle rotation.

Our system uses the lock function on bone which one does not want to move. This is corresponding to spartial constraints.

Fig. 8. Animation based on inverse kinematics

The red part in Fig. 8 is the bone which is locked. In this case, the rotation and displacement values calculated from IK solver does not be applicable. So if one does not want to apply the transformation then one must use the lock function.

The proposed tool has one inverse kinematics mode and two forward kinematics modes. They have two kinds of key frame information values. One uses to establish key frame information fixed in real-time, and the other uses to reuse motion data which are fixed previously. When the animation is created, these two key frame values are added. After then we can get the one motion.

3. MOTION EDITING FUNCTION

In this section we will explain about the motion editing function using multi-level B-spline with quasi-interpolant. It is similar to multi-level B-spline fitting algorithm of [6]. In [6], they used the multi-level B-spline algorithm. But we use the multi-level B-spline with quasi-interpolant. The cubic B-spline basis function in Fig. 9 is expressed by the piecewise continuous function as follows.

By the equation $B_{j,3} = B_{0,3}(t-j)$, the interpolation function is $f(t) = \sum_{i=-1}^{m} \phi_i B_{i,3}(t)$. In this case, the knot

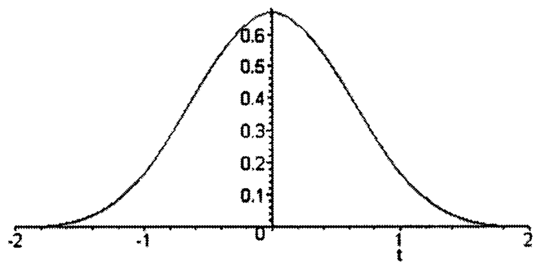

Fig. 9. Cubic B-spline basis function

vector is $\{-1, 0, 1, \cdots, n+1\}$. The control point is determined by the linear system $B\Phi = X$ and, by the least-square means,

$$\begin{pmatrix} B_{-1,3}(0) & B_{0,3}(0) & \cdots & B_{m+1,3}(0) \\ B_{-1,3}(1) & B_{0,3}(1) & \cdots & B_{m+1,3}(1) \\ \vdots & \vdots & \cdots & \vdots \\ B_{-1,3}(n) & B_{0,3}(n) & \cdots & B_{m+1,3}(n) \end{pmatrix} \begin{pmatrix} \phi_{-1} \\ \phi_0 \\ \vdots \\ \phi_{m+1} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \tag{5}$$

But this method depends on the number of data and the real time processing is impossible. The above linear system $\Phi = (B^T B)^{-1} B^T X$ can be changed as the local linear system by the use of compact support of B-spline. (Refer Fig. 10).

From this, $B_{i,3}(t) = B_{0,3}(t-i)$ has support $[(i-2)h, (i+2)h]$. Therefore the proximation set is defined same as [7] and the partial linear system $B_i \Phi_i = X_i$ is defined as follows.

$$\begin{pmatrix} B_{i-3,3}((i-2)h) & B_{i-2,3}((i-2)h) & \cdots & B_{i+3,3}((i+2)h) \\ B_{i-3,3}((i-1)h+1 & B_{i-2,3}((i-1)h+1) & \cdots & B_{i+3,3}((i-1)h+1) \\ \vdots & \vdots & \cdots & \vdots \\ B_{i-3,3}((i+2)h) & B_{i-2,3}((i+2)h) & \cdots & B_{i+3,3}((i+2)h) \end{pmatrix} \begin{pmatrix} \phi_{i-3} \\ \phi_{i-2} \\ \vdots \\ \phi_{i+3} \end{pmatrix} = \begin{pmatrix} x_{(i-2)h} \\ x_{(i-1)h+1} \\ \vdots \\ x_{(i+2)h} \end{pmatrix} \tag{6}$$

After then, $\Phi_i = (B_i^T B_i)^{-1} B_i^T X_i$ is solved by the least-square method. In this case we get 5 solutions but we take only the middle value ϕ_i and the rest of control points are solved by changing the intervals.

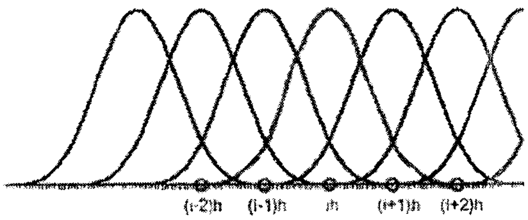

Fig. 10. Compact support of cubic B-spline function

The error is made because it is found by the least-square method. So we solve it by using multi-level method as [7] and [8]. Fig. 11 shows the original motion data and Fig. 12 shows the result of motion editing using multi-level B-spline with quasi-interpolant.

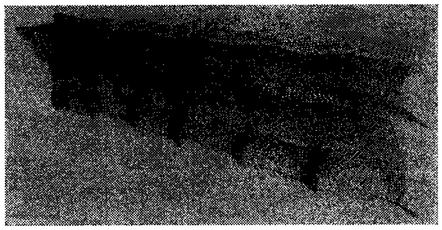

Fig. 11. Original motion data

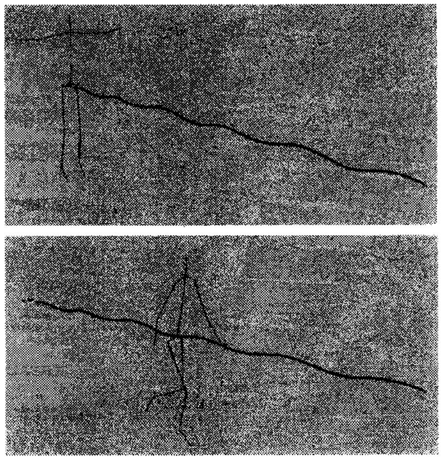

Fig. 12. Motion editing result

6. CONCLUSIONS

The proposed tool is made to create 3D motion animation easily and conveniently. It has plug-in with 3D MAX as yet. But we will make the plug-ins with other 3D rendering tool. Also this is similar to the Animanium of SEGA. But the editing function is big difference with it.

REFERENCES

[1] A. Bicallo, and S. Feltman, *MAXScript & the SDK for 3D Studio MAX*, SYBEX, Indianapolis, 2000.

[2] M. Girard, and A. A. Maciejewski, "Computational Modeling for the Computer Animation of Legged Fiqures," *ACM*, Vol.19, No.3, pp. 263-270, 1985.

[3] D. B. Paull, *Programming Dynamic Character Animation*, Charles River Media,

Boston, MA, 2002.

[4] M. W. Spong, and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons. Inc, Hoboken, NJ, 1989.

[5] C. Welman, "Inverse kinematics and geometric constraints for articulated figure manipulation," *Thesis for Master Degree of Simon Fraser University*, 1993.

[6] J.Lee and S. Shin, "A hierarchical approach to Interactive Motion Editing for Human-Like figures," *SIGGRAPH 99*, pp. 39-48, 1999.

[7] R. Parent, *Computer Animation , Algorithms and Techniques*, Morgan Kaufmann Publisher, San Francisco, Calif., 2002.

[8] Byung-Gook Lee, Jae-chil Yoo, and Joon-Jae Lee, "An Efficient Scattered Data Approximation Using Multilevel B-splines Based on Quasi-Interpolants," *3-D Digital Imaging and Modeling 2005*, pp. 110-117, 2005.

Woosuk Ju

He received his B.S. degree from Kyungnam University, Masan, Korea, in 1998 and his M.S. degrees from Dongseo University, Busan, Korea, in 2005. He is currently a Ph.D. degree student of the Pusan National University, Korea. He is currently a professor of the Dept. of Game in Dongseo University, Pusan, Korea. His research interests include online game development and image processing.

Choongjae Im

He received his B.S. and M.S. degrees from Chungnam National University, Deajeon, Korea, in 1991 and 1993, respectively. He is currently a professor of the Dept. of Game & Mobile Contents in Keimyung University, Daegu, Korea. His research interests include multi-platform game development and computer graphics.