

Skinning Mesh Algorithm Improvement Based on Double Weights

Liang Li, Zhong Li, Jiao Lin, Fengyuan Zou, Hongyao Xu, Junliang Wang

Department of Mathematical Sciences, Zhejiang Sci-Tech University, Hangzhou, 310018, China

Received: 12 Oct. 2012, Revised: 22 Dec. 2012, Accepted: 25 Dec. 2012

Published online: 1 Jun. 2013

Abstract: Skinning animation is one of the important contents in computer animation. This paper presents a skinning mesh algorithm improvement based on double weights for the bone bending and rotating movements. When the bone bends, we use the original weight; when the bone rotates, we set the new weight according to the Gaussian curvature in the influencing area. This method can effectively solve the "candy-wrapper" problem. The experimental results show that our method is simple and practical, especially for the realization of the human arm and leg animations.

Keywords: Skinning Animation, Double Weights, Gaussian Curvature, Skeleton.

1 Introduction

With the development of digital media and entertainment industry, computer animation has recently made great progress and it has been widely used in the field of game, movie, advertising, etc^[1]. Skinning animation has become the most widely used technology in all kinds of real-time animation applications. But traditional skinning algorithm has some shortcomings. One common problem is the "candy-wrapper" effect^[2], which produces some non-natural postures for some movements. When the skeleton rotates by itself, the local area of the joint will appear the deformation and cause serious twisting distortion, as shown in Fig 1. Many scholars made some improvements later. For example, Kavan and Kaza^[3] provided the blending algorithm based on the bone. This method needs more manual adjusting parameters and it can only be used for the area which has just two joints. Lewis et al^[4] regarded the skinning deformation as the interpolation process and they proposed the radical basis functions for interpolating skinning frames. Yang and Zhang^[5,6] put forward a geometric method, although the method can simulate real results, but it greatly increases the computation and computer hardware requirements, then they gave an animation algorithm based on the skin stretch. This algorithm is simple, but it increases the degree of freedom and there are still some drawbacks for the realistic rendering.

This paper presents an improved skinning animation method for solving the "candy-wrapper" problem. In the traditional method, the child bone rotating by itself does not affect the father bone. For instance, when the below arm rotates, the upper arm will not move. Therefore, the elbow will twist together because the weights of vertices surrounded by the bounding box in the upper bone are set small according to the below bone. However, the rotating of the below arm actually causes the spinning of the upper arm in the real movement. Based on this observation, we introduce a double-weight setting method which chooses different weights for the rotating and the bending of the bone. The experimental results show that our method is more in line with the real human body movements.

2 Basic Principle of Skinning Mesh Algorithm

Skinning mesh animation is a kind of surface deformation algorithm based on the local operation. It binds the bone and the skin and uses a series of bones to drive the skin of the mesh model, which produce the desired animation effect.

Every bone has two corresponding matrices: Local transformation matrix (*LMatrix*) and Combination transformation matrix (*CMatrix*). Local transformation matrix defines the bone's translating and rotating

* Corresponding author e-mail: lizhong@zstu.edu.cn



Fig. 1 The animation effect of the arm (The left is a standard arm. The right is the rotation effect by traditional weight, which is called as the "candy-wrapper" problem)

movements in its father coordinate system and the combination transformation matrix is used to determine the bone's position in the whole viewing space. Combination transformation matrix decides the rendering of the role animation in the viewing space. The bone's combining transformation matrix (*CMatrix*) is equal to the one by multiplying its local transformation matrix (*LMatrix*) and its father bone's combination transformation matrix (*PCMatrix*). The calculating formula is

$$CMatrix = LMatrix \cdot PCMatrix \quad (1)$$

In addition to the local transformation matrix and the combination transformation matrix, any bone has the final transformation matrix (*FMatrix*), which determines the position of vertices in the world coordinate system, the calculating formula is

$$FM_i = OM_i \cdot CM_i \quad (2)$$

Here OM_i is the initial offset transformation matrix (*OMatrix*) of each bone, CM_i is the combination transformation matrix (*CMatrix*) of each bone.

The binding of the bone and the skin often adopts the point binding algorithm. It regards the skin of a whole mesh and each vertex of the skin are controlled by several adjacent bones. The animation effect is decided by the weights between the bone and the vertex. The calculating formula is

$$V'_i = \sum_{i=1}^n V_i \cdot FM_i \cdot W_i \quad (3)$$

where V'_i represents the vertex's position in the world coordinate system, V_i is the vertex's coordinate in the

model space, FM_i denotes the bone's final transform matrix of the i^{th} bone, W_i is the weight between the vertex and the i^{th} bone and the sum of all W_i is 1.

3 Interpolation Animation Realization

We use the interpolation technique of key frames to realize continuous animation. The interpolation of translation can be realized by through the traditional linear interpolation. Supposing that $p(t_1)$ represents the position vector in one key frame of t_1 and $p(t_2)$ represents the position vector in another key frame of t_2 , the corresponding position vector $p(t)$ between two key frames in t can be calculated by following formula

$$p(t) = p(t_1) + \frac{t-t_1}{t_2-t_1} \cdot (p(t_2) - p(t_1)) \quad (4)$$

The rotation is normally realized by the quaternions' spherical linear interpolation (*slerp*). This method is also simple to solve the rotating interpolation between key frames. As shown in Fig 3, supposing corresponding quaternions of two vectors A and B are q_1 and q_2 , Ω represents the angle between q_1 and q_2 , U is the parameter of interpolation, we can conveniently get the middle quaternion $p = \text{slerp}(q_1, q_2, u)$ by the spherical linear interpolation equation.

$$p = \text{slerp}(q_1, q_2, u) = q_1 \cdot \frac{\sin[(1-u)\Omega]}{\sin\Omega} + q_2 \cdot \frac{\sin(\Omega u)}{\sin\Omega} \quad (0 \leq u \leq 1) \quad (5)$$

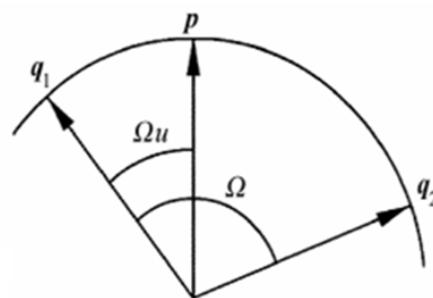


Fig. 2 Spherical linear interpolation

4 Simulation Results and Discussio

Based on the defect, we add the new weight setting for the arm movement of the human body. When the below arm rotates, we reset the weights of vertices between the

below arm and the upper arm, this can make the vertices in the upper arm have more influence from the below arm. So it can reduce the distortion when the below bone rotates. However, if the below bone bends, the vertices in the upper arm will also bend and cause the distortion too. At this time, we need to choose original weight for vertices in the upper bone. The new weight's setting can be obtained through the following process.

First of all, we use the pickup and feedback function in OpenGL to select several vertices from the upper arm, as shown in Fig 4.

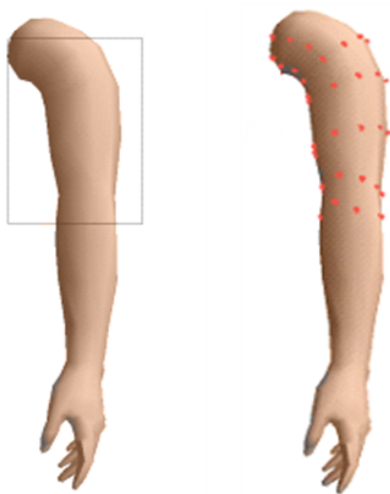


Fig. 3 The vertices selection in the upper arm (The left is a selected rectangular area and the right are the corresponding vertices)

Then, we need to estimate the curvature of each vertex. Gaussian curvature is one of the most important geometric properties in the surface theory. Supposing the vertex's two principal curvature are k_1, k_2 , then $k = k_1 \cdot k_2$ can be called as Gaussian curvature. When we map the vertex on the surface to the centre of the unit sphere, and project the point from the normal onto the surface of the unit sphere, we can build the relation for the point between on the surface and on the sphere, which is called as the spherical representation or Gaussian mapping. It explains that Gaussian curvature can reflect the local bending degree of the local surface. We use following formula to estimate Gaussian curvature^[7].

$$K(v) = \frac{1}{A(V)} (2\pi - \sum_{v_i \in N(v)} \theta_i) \quad (6)$$

where $\sum_{v_i \in N(v)} \theta_i$ is the total angle of the point's neighbor triangle, $A(V)$ represents the total area of Voronoi, as the dark area shown in Fig 4.

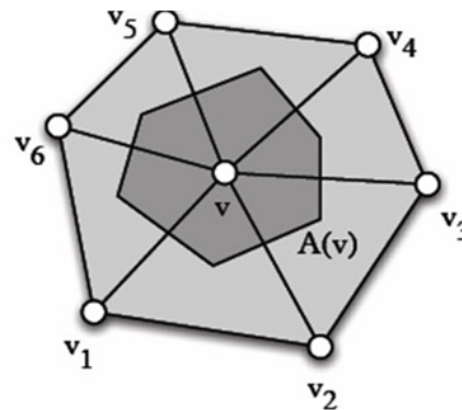


Fig. 4 The vertex with its Voronoi area

For the given vertex of the mesh model, when its Gaussian curvature is larger, it shows that the neighboring of its vertex bends much higher, which may have the larger influence to the bending motion during the mesh animation process. We set a threshold and choose the vertices whose Gaussian curvature is greater than it, then we select the weight influenced by the below arm and the upper arm through several experiments until it adjusts to the ideal animation effect. We finally save the value as the new weight.

When the bone's movement is interpolated for double weights, we need to judge whether there is an angle between the bone in current frame and the same bone in next frame. If there is the angle variance, we think the motion is bending. Whereas, we think the motion is rotating by itself. When the bone skeleton bends, we use the original weight. When the bone skeleton rotates, we choose the new weight. It is easy to judge whether there is the angle variance between two frames. For the human skinning animation, we just consider the bone movements of the arm and the leg.

5 Experimental Result

We implement above animation algorithm on a 2.40Ghz PC with 2GB of memory and a ATI 2400PRO graphics card and do some experiments to compare our results with the traditional skinning animation effect.

Fig 5 is the arm bending effect of human body. The left is the arm bending 45 degrees and the right is the bending arm with 135 degrees. We use the original weight for two cases and the results satisfy with the animation effect. The left of Fig 6 is the arm which is not rotated. The right is the arm effect by rotating 180 degrees with the new weight. It shows that the new weight can effectively avoid the "candy-wrapper" problem. In Fig 7,

the left is the male model whose arm does not rotate. The middle is the arm skinning animation effect by rotating 150 degrees which uses only one weight. We can see that it appears the "candy-wrapper" problem. The right uses the new weight for rotating the arm by 150 degrees. We find it can effectively solve the "candy-wrapper" problem. In Fig 8, the left is the female model whose leg does not rotate. The middle is the leg skinning animation by rotating 90 degrees which uses just one weight. At this time, we find that the thigh does not rotate when the below leg rotates. The right is the leg rotating effect with 90 degrees by using the new weight. We see that the thigh changes as the below leg rotates. It is more accordant with the leg movement of real human.



Fig. 7 The rotating effect of male model's arm

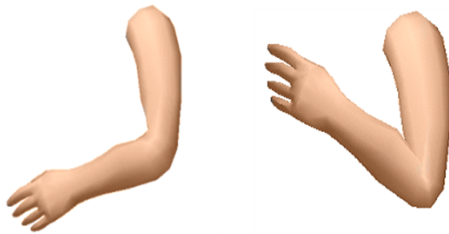


Fig. 5 The arm bending effect by using the original weight



Fig. 8 The rotating effect of female model's leg

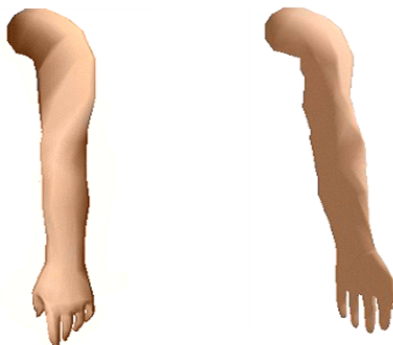


Fig. 6 The arm rotating effect by using the new weight

6 Summary and Future Work

The traditional skinning mesh animation only considers one weight, which could produce obvious distortion when rotating by itself. In this paper, two weights are set for rotating and bending moving cases. By using different

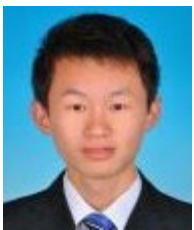
weights, the results are more in line with the real human body movements and guarantee the real-time and vivid requirements of human animations. Our future work is to realize more free movements for the human animations by through the interactive mesh operation.

Acknowledgement

This research was supported from the National Natural Science Foundation of China (No. 60903143 and 51075421); Natural Science Foundation of Zhejiang Province of China (No. Y1110504); Qianjiang Talent Project of Zhejiang Province of China (No. QJD0902006); Science and Technology Project of Zhejiang Province of China (No. 2012C21035); Project of National Research Center for College Teaching of China (No. 2009IM010400-1-11) and Project of Zhejiang Provincial Research Center of Clothing Engineering Technology.

References

- [1] D. L. James, C.D. Twigg, Skinning mesh animations [J]. ACM Trans on Graphics. (2005), 24(3), 399-407.
- [2] T.N. Magnenat, R. Laperrire, D. Thalmann, Joint-dependent local deformations for hand animation and object grasping [C]. Proceedings of Graphics Interface. (1988), 49(7), 26-33.
- [3] L. Kavan, J. Zra, Real time skin deformation with bones Blending [C]. Proceedings of WSCG, Plzen, Czech Republic. (2003), 17-22.
- [4] J.P. Lewis, M. Cordner, N. Fong, Pose space deformations: a unified approach to shape interpolation and skeleton-driven deformation [C]. Proceedings of ACM SIGGRAPH 2000, Annual Conference Series. (2000), 165-172.
- [5] X. Yang, J.J. Zhang, Realistic skeleton driven skin deformation [C]. Proceedings of the Third Technical Session on Computer Graphics and Geometric Modeling. (2005), 1109-1118
- [6] X. Yang, J.J. Zhang. Stretch it-realistic smooth skinning [C]. Proceedings of the International Conference on Computer Graphics, Imaging and Visualization. (2006), 323-328
- [7] M. Meyer, M. Desbrun, P.H. Schroder, et al, Discrete differential-geometry operators for triangulated 2-manifolds [A]. Visualization and Mathematics. (2002), 52-58



Liang Li, born in 1988, Master of Zhejiang Sci-Tech University. His research interests include computer animation, mesh deformation and mesh cutting.



Zhong Li, born in 1975, Professor of Zhejiang Sci-Tech University. His research interests include digital geometric processing, garment of CAD. Corresponding author of this paper. E-mail: lizhong@zstu.edu.cn.



Jiao Lin, born in 1989, Master of Zhejiang Sci-Tech University. His research interests include skeleton extraction and non-photorealistic rendering.