Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Meshless simulation for skeleton driven elastic deformation*

SONG Chao, ZHANG Hong-xin†‡, HUANG Jin, BAO Hu-jun
(*State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, China*)
†E-mail: zhx@cad.zju.edu.cn
Received May 20, 2006; revision accepted June 21, 2006

Abstract: A meshless simulation system is presented for elastic deformation driven by skeleton in this paper. In this system, we propose a new method for calculating node rotation while applying a similar technique with stiffness warping to tackle the nonlinear large deformation. In our method, all node rotations are evaluated from sampling points in attached skeleton by constructing and solving the diffusion partial differential equation. The experiments indicated that the method can enhance the stability of the dynamics and avoid fussy sub-step calculation in static deformation edition. Moreover, rational deformation results for the area around the skeleton joints can be simulated without user interaction by adopting the simplified technique.

Key words: Physically-based deformation, Skeleton driven, Meshless methods, Rotation field
doi:10.1631/jzus.2006.A1596 **Document code:** A **CLC number:** TP391

INTRODUCTION

Physical simulation is an important area for creating realistic character animations. In the film industry, animators require detailed control of the motion of their characters. Many techniques have been developed for automatically simulating characters' motion. The skeleton driven elastic deformation is paid much attention in these techniques, since many motions and deformations in character animations are mainly driven by skeleton. In general, however, it is not easy to get fast and realistic effects of deformation because of large nonlinear deformations in many applications.

It is a well-established approach to simulate deformation as continuums and solve their governing equations numerically. When adopting an object, it is necessary to choose the measure of strain. Green strain tensor, which consists of linear terms and nonlinear terms, is a common choice for large deformations. But the corresponding computation cost is expensive. An alternative choice is adopting Cauchy strain which has only linear terms. However, this simplified method can produce quite unnatural results when it is simply applied to bending or deformation of relatively large magnitude. For example, the volume of the deformed shape can increase unrealistically (Müller *et al.*, 2002; Guo and Qin, 2005). In order to tackle the problem, many techniques were proposed, with adoption of stiffness warping being representative. Stiffness warping technique compensates the stiffness matrix for rotational parts of deformation and yields visually plausible result.

As for solving the governing differential equations of motion, many methods such as finite difference, finite element and meshless methods are introduced. Furthermore, with the development of data acquisition power and point sampled geometry, meshless methods have attracted much attention because of its many advantages in simulating large deformation and material failure.

In this paper, we present a system which simulates the nonlinear large deformation driven by skeleton in a meshless framework. It is convenient for simulating both dynamics and statics. Moreover, constructing finite element discretization can be

‡ Corresponding author
* Project supported by the National Basic Research Program (973) of China (No. 2002CB312102) and the National Natural Science Foundation of China (Nos. 60021201, 60333010 and 60505001)

avoided for simulating complex model which can be obtained by scan devices.

An implementation procedure applying warping stiffness technique in meshless methods framework is given. The innovative aspect lies in the way of getting the rotation tensor field for compensating stiffness. We calculate all nodes rotation based on the rotation of the sample skeleton points by constructing and solving the rotation diffusion equation. Compared with some methods in literature, our method of rotation calculation is non-iterative and can avoid error accumulation. So, in statics simulation, fussy sub-step calculation is not necessary for calculating rotation. In dynamics, it can increase the stability without decreasing the simulating rate. In the area of skeleton joint, our system does not require users' interaction and can be automatically implemented.

RELATED WORK

Physically based deformation has two decades of history in Computer Graphics (Terzopoulos *et al*., 1987). Previous work in this field had two central aims: to speed up the simulation and increase the realism of the result. Mesh based methods, such as the finite element method and finite difference method, are prevailing. Comprehensive surveys on this subject can be found in (Gibon and Mirtich, 1997; Nealen *et al*., 2005).

Skeleton driven deformation is implemented by two types of methods which are free-form deformation (FFD) and physical simulation. FFD introduced by Thomas and Scott (1986), is based on geometry and many techniques were involved. In physical simulation area, Capell *et al*.(2002; 2005) did some outstanding work and obtained realistic and fast results in finite element method framework. As for nonlinear deformation, Capell's technique can apply stiffness warping technology which is proposed to avoid unnatural result (Müller *et al*., 2002; Müller and Gross, 2004). In detail, to address large relative rotational deformations with a single object, stiffness warping technique tracks the rotation of each node and warps the stiffness. It must be indicated that user must set up the weight value for every part in joint area in Capell's simulation.

Recently, meshless methods are attracting much attention and were introduced to Computer Graphics by Desbrun and Cani (1995). Guo and Qin (2005) and Guo *et al*.(2006) applied element-free Galerk in methods in modal analysis. For the advantages without mesh topology, Pauly *et al*.(2005) succeeded in simulating fracture solids.

Our method is similar to Müller's approach in that rotations are handled separately to reduce or eliminate artifacts. The difference is that, whereas their rotations were obtained by an iterative method, our method of rotation calculation is based on diffusion without iteration. Compared with Capell's work, our system can decrease users' interaction in the joint area. Moreover, our system is implemented in original space, instead of modal space allowing for statics simulation application. In addition, our work has focused on a deformable object driven by skeleton, thus the ghost force effects can be suppressed by constraint force. The method of rotation calculation in this paper can be used in finite element frame.

In the following, deformation simulation problem and meshless method based on moving least square approximation are briefly described first. Then, the method of rotation calculation is discussed in the areas of geometry discretization, rotation diffuse equation and solution. At the last section, several simulation results are provided.

MESHLESS METHODS AND SKELETON DRIVEN DEFORMATION

In general, an object is represented as a domain $\Omega\subset\mathbb{R}^3$ and the skeleton in the object is represented as a graph $S\subset\Omega$. As any point $p\in\Omega$, the movement is represented by a time dependent function p: $\Omega\times\mathbb{R}\rightarrow\mathbb{R}^3$: $(\boldsymbol{X},0)\mapsto\boldsymbol{x}(\boldsymbol{X},t)$, where $\boldsymbol{X}$ is the material coordinate (i.e. initial location coordinate), $\boldsymbol{x}(\boldsymbol{X},t)=\boldsymbol{X}+\boldsymbol{u}(\boldsymbol{X},t)$ is the location coordinate of node p at time t and $\boldsymbol{u}(\boldsymbol{X},t)$ is the displacement. The simulation target is getting all points coordinate after the object deforming. It is well known that deformation follows the governing equation

$$\boldsymbol{M}\ddot{\boldsymbol{x}}+\boldsymbol{C}\dot{\boldsymbol{x}}+\boldsymbol{K}(\boldsymbol{x}-\boldsymbol{X})=\boldsymbol{f}_{\text{ext}}, \text{ and } \forall\boldsymbol{x}^*\in S \text{ s.t. } \boldsymbol{x}^*=\text{bone}. \quad (1)$$

If the rate effect is negligible for deformation, Eq.(1) changes into statics equation

$$\boldsymbol{K}(\boldsymbol{x}-\boldsymbol{X})=\boldsymbol{f}_{\text{ext}}, \text{ and } \forall \boldsymbol{x}^* \in S \text{ s.t. } \boldsymbol{x}^*=\text{bone}. \quad (2)$$

In Eqs.(1) and (2), $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{f}_{\text{ext}}$ denote mass matrix, lamp matrix and node force vector respectively. The stiffness matrix $\boldsymbol{K}$ is correlated with the deformation energy and we call the item of $\boldsymbol{K}(\boldsymbol{x}-\boldsymbol{X})$ elastic forces. The Green strain tensor is represented by using matrix $\boldsymbol{\varepsilon}$:

$$\begin{aligned}\boldsymbol{\varepsilon}_{ij} &= \frac{1}{2}\left(\delta_{lm}\frac{\partial \boldsymbol{x}_l}{\partial \boldsymbol{X}_i}\frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{X}_j} - \delta_{ij}\right) \\ &= \frac{1}{2}\left(\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{X}_j} + \frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{X}_i} + \frac{\partial \boldsymbol{u}_\alpha}{\partial \boldsymbol{X}_i}\frac{\partial \boldsymbol{u}_\alpha}{\partial \boldsymbol{X}_j}\right).\end{aligned} \quad (3)$$

where $\boldsymbol{u}=\boldsymbol{x}-\boldsymbol{X}$ is a displacement vector. Omitting the nonlinear terms in Eq.(3), the problem becomes linear and easy. The simplified strain is called Cauchy strain and is familiar in engineering analysis. In this paper, we use simplification and compensate the result by using stiffness warping technique.

We apply the meshless methods to discrete the governing Eq.(1). The advantages can be summarized as follows: (1) they can easily handle very large deformations, since connectivity among nodes does not exist; (2) in areas where more refinement is needed, nodes can be added quite flexibly; (3) data management overhead can be minimized during simulation.

Moving least square (MLS) shape functions

In this paper, the shape functions are constructed by using the MLS technique. The MLS method can provide continuous and smooth field approximation throughout the analysis domain with any desirable order of consistency. We associate each node I with a positive weight function ω_I of compact support. So we define the influent domain of the node: $\Omega_I=\{\boldsymbol{x}\in\mathbb{R}^3: \omega_I(\boldsymbol{x})=\omega(\boldsymbol{x},\boldsymbol{x}_I)>0\}$ where $\omega(\boldsymbol{x},\boldsymbol{x}_I)$ is the weight function associated with node I evaluated at position $\boldsymbol{x}$. The approximation of the field function f at a position $\boldsymbol{x}$ is only affected by the nodes whose weights are non-zero at $\boldsymbol{x}$. We denote the set of such nodes as the set L. Considering a field function defined in the analysis domain $\Omega\subset\mathbb{R}^3$, we can construct its approximation as follows:

$$f(\boldsymbol{x}) \approx f^h = \sum_{i=1}^{4} p_i(\boldsymbol{x})a_i(\boldsymbol{x}), \quad (4)$$

where $p_i(\boldsymbol{x})$ are linear basis function $\boldsymbol{p}^{\mathrm{T}}=\{1,x,y,z\}$, $a_i(\boldsymbol{x})$ are coefficients which can be derived by minimizing weighted errors at all nodes in the influent domain of $\boldsymbol{x}$

$$J = \sum_{I\in L} w(\boldsymbol{x}-\boldsymbol{x}_I)[(\boldsymbol{p}(\boldsymbol{x}_I))^{\mathrm{T}} a(\boldsymbol{x}) - f_I]^2, \quad (5)$$

where f_I is the real field value at the node I. If the field f is also a function depended on time, we can write

$$f(\boldsymbol{x},t) = f^h(\boldsymbol{x},t) = \sum_{I\in L}\phi_I(\boldsymbol{x})f_I(t), \quad (6)$$

where $\phi_I(\boldsymbol{x})=\omega_I(\boldsymbol{x})[\boldsymbol{p}(\boldsymbol{x})]^{\mathrm{T}}[\boldsymbol{M}(\boldsymbol{x})]^{-1}\boldsymbol{p}(\boldsymbol{x}_I)$, and $\boldsymbol{M}(\boldsymbol{x}) = \sum_{I\in L}\omega_I(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_I)[\boldsymbol{p}(\boldsymbol{x}_I)]^{\mathrm{T}}$ is called the moment matrix. Especially, if the displacement field is taken as a field function, we can write

$$u(\boldsymbol{x},t) \approx u^h(\boldsymbol{x},t) = \sum_{I\in L}\phi_I(\boldsymbol{x})u_I(t). \quad (7)$$

The above description shows that the weight function plays an important role in MLS approximation. We use the compactly supported radial spline function

$$\omega(\boldsymbol{x},\boldsymbol{x}_I) = \omega_I(r) = \begin{cases} 1-6r^2+8r^3-3r^4, & r\le 1; \\ 0, & r>1, \end{cases} \quad (8)$$

where $r=(\|\boldsymbol{x}-\boldsymbol{x}_I\|)/h_i$. The support radius h_i is determined adaptively depending on the local density of nodes.

Stiffness warping

The main difference between meshless methods and finite element methods is model discretization and the shape function (i.e. approximation method). Hence, after finishing the two processes, our simulation is the same as that in finite element framework. Gauss integration method is utilized on the integral cells which were detailedly discussed in (Belytschko *et al*., 1994).

In this paper, stiffness warping strategy in meshless methods is described. As pre-computing process, we calculate the initial stiffness matrix in linear elasticity model

$$\boldsymbol{f}_{\text{elastic}} = \boldsymbol{K}\cdot(\boldsymbol{x}-\boldsymbol{X}), \tag{9}$$

where $\boldsymbol{K}$ is the whole stiffness matrix which is expressed by many 3×3 sub-matrixes. Let $\boldsymbol{k}_{ij}$ be the 3×3 sub-matrix of $\boldsymbol{K}$ containing entries $\boldsymbol{K}_{vw}$, with $3i-2\le v\le 3i$ and $3j-2\le w\le 3j$. We get the elastic force $\boldsymbol{f}_i$ at node i

$$\boldsymbol{f}_i = \sum_{j=1}^{n}\boldsymbol{k}_{ij}(\boldsymbol{x}_j-\boldsymbol{X}_j). \tag{10}$$

During the simulation, at every time step, stiffness warping is implemented as follows:

$$\boldsymbol{f}_i = \boldsymbol{R}_i\sum_{j=1}^{n}\boldsymbol{k}_{ij}(\boldsymbol{R}_j^{-1}\boldsymbol{x}_j-\boldsymbol{X}_j), \tag{11}$$

where $\boldsymbol{R}_i$ is the rotation of node i, $\boldsymbol{R}_j$ is the rotation of node j. The node rotation can be calculated with the method described in later sections. We update the elastic forces in Eq.(1) and solve the motion governing equation in every time-step. Moreover, the same strategy can be used to deal with Eq.(2) for simulating the statics deformation.

SKELETON CONSTRUCTION AND DEFORMATION

Constructing skeleton of a given object is necessary in our system. As for the object without skeleton, we have two methods to construct skeleton according to requirement. One method is to take interactively the centre shaft of the object as a skeleton after obtaining the Voronoi mesh (Lin *et al*., 2005). The other is Capell's method, where the user creates a sample point by clicking on the object with the mouse. If the ray through the mouse point (from the camera projection center) intersects the object at least twice, a sample point is placed midway between the first two intersections. After connecting all sample points with lines, a skeleton structure is constructed. Of course, we can select more points in the skeleton as sampling points during the calculation. Moreover, according to simulation demand, we also select some joint points which have different material property or weak rotation confinements.

We calculate the skeleton deformation by applying standard nonlinear finite element method. Because of having much less nodes than the whole object, the skeleton discretization and calculation is not expensive even though it is a strict mechanics computation.

DIFFUSION BASED ROTATION CALCULATION

Directly and non-iteratively calculation of nodes rotation is critical in our simulation system. The previous methods for obtaining node rotation are all iterative. That is, calculating node rotation must depend on the result of the last sub-step calculation. We take Müller's method for instance. For getting the rotation of a node or element at time t_{k+1}, translation matrix from initial location to the location at time t_k must be obtained. The method make polar-decompose to the matrix which is gotten by deleting the scale part from the translation matrix and taking the rotation part as the node or element rotation. It is obvious that the iterative methods sometimes decrease the simulation stability and rate. Specially, fussy sub-steps calculation must be done and this increases the calculation cost greatly. Our system applies a solution method in which rotation calculation is based on diffusion from the sample point rotation in the skeleton. The method has more stability and can directly solve statics simulation in one step.

Rotation of the sample skeleton points

The node rotation is calculated based on diffusion from the rotation of sample points in the skeleton. The sample point rotation can be obtained from the skeleton location after deforming. In detail, it can be calculated with the method given by Yu *et al*.(2004). In detail, the sample point rotation is obtained by calculating the local coordinate frame rotation after deforming and denoted by corresponding quaternion.

Rotation diffusion equation

During simulating skeleton driven elastic deformation, we make an analogy between the node rotation distribution and the steady-state temperature field. We assume a sample skeleton point p and a constraint set $T\subset\Omega$. If the temperature is 1 at point p and is 0 at all points in T, we can get a steady-state temperature or diffusion scalar field function $\mathbb{R}$: $\Omega \mapsto [0,1]$ which has maximal value at point p. The values are in inverse proportion with the distance away from point p and close to zero near the constraint nodes. Analogously, we take the point p as rotation diffusion resource and construct all the nodes rotation in the simulating object. In detail, we individually construct one scalar field R_i (i=1, 2, 3, 4) for each item in the rotation quaternion. Each of four scalar fields follows the steady-state temperature equation (i.e. Laplace equation) and the boundary condition which is the corresponding item in the quaternion denoting the sample skeleton rotation.

$$\nabla^2 R = \frac{\partial^2 R_i}{\partial x^2} + \frac{\partial^2 R_i}{\partial y^2} + \frac{\partial^2 R_i}{\partial z^2} = 0,$$

$$\text{s.t. } R_i(p) = \overline{r}_{pi}, R_i(p') = 0, \forall p' \in T, \ i = 1,\ 2,\ 3,\ 4. \quad (12)$$

As for any one node s in the calculation domain, we get the four scalar values r_{s1}, r_{s2}, r_{s3}, r_{s4} by solving the above equations. The quaternion q_s=(r_{s1}, r_{s2}, r_{s3}, r_{s4}) expresses the rotation of the node s with normalization being necessary.

Diffusion equation solving

In order to solve the diffusion Eq.(12), we even tried to use several methods. For example, in previous experiment, we applied the volume Laplacian method which puts the nodes and creates volume graph without finite element discretization. But it was found that the diffusion effect of rotation from inner nodes to exterior nodes was not good. The reason is that the Laplacian coordinate of exterior nodes does not depend on the inner nodes.

Concerning the existence of discretization for solving motion equation, we turn into applying the meshless method to solve the diffusion equation. It is easy to be implemented and the calculation cost is small. Generally, we think of the functional

$$\Pi = \int_V \left\{ \frac{1}{2} \left[\left(\frac{\partial R_i}{\partial x} \right)^2 + \left(\frac{\partial R_i}{\partial y} \right)^2 + \left(\frac{\partial R_i}{\partial z} \right)^2 \right] \right\} \mathrm{d}V. \quad (13)$$

Solving Eq.(12) is equivalent to the problem of solving the minimal value of the functional Π (i.e. $\delta\Pi$=0). Applying MLS approximation, whose scalar value is $R_i(\boldsymbol{x},t) \approx \sum_{I\in L} \phi_I(\boldsymbol{x}) R_{iI}(t) = \boldsymbol{N}(\boldsymbol{x}) \boldsymbol{R}_i(t)$ in which $\boldsymbol{R}_i(t)$ is a $m\times1$ column vector which indicates the nodal field value at time t, where m is the node number in L. Then, we calculate

$$\left\{ \frac{\partial R_i}{\partial x_i} \right\} = [\boldsymbol{B}]\{R\} = \begin{bmatrix} \frac{\partial \boldsymbol{N}}{\partial x} & \frac{\partial \boldsymbol{N}}{\partial y} & \frac{\partial \boldsymbol{N}}{\partial z} \end{bmatrix}^{\mathrm{T}} \{\boldsymbol{R}_i\}$$

and substitute it into Eq.(13)

$$\begin{aligned} \Pi &= \sum_{e=1}^{n} \int_{V_e} (\{\boldsymbol{R}_i\}[\boldsymbol{B}])^{\mathrm{T}} (\{\boldsymbol{R}_i\}[\boldsymbol{B}])/2\mathrm{d}V \\ &= \sum_{e=1}^{n} (\{\boldsymbol{R}_i\})^{\mathrm{T}} [\boldsymbol{K}]^e \{\boldsymbol{R}_i\}/2, \end{aligned} \quad (14)$$

where V_e is integration cell in the object and $[\boldsymbol{K}]^e = \int_{V_e} [\boldsymbol{B}]^{\mathrm{T}} [\boldsymbol{B}] \mathrm{d}V$ is cell stiffness matrix. By calculating the differential to Eq.(14) and making it equal to zero, we get the linear system $\boldsymbol{K}\{\boldsymbol{R}_i\}$=0 with its boundary being equivalent to that of Eq.(12). Because $\boldsymbol{K}$ is sparse matrix in the linear system, solving the linear system is not expensive and many methods can be used, such as LU-decomposition of sparse matrix.

IMPLEMENTATION AND DISCUSSION

As for simulating skeleton driven elastic deformation, our rotation calculation method can be applied in dynamics and statics simulation according to user's application. In dynamics simulation, implicit integral method presented by Baraff and Witkin (1998) is applied. In summary, simulation is implemented as follows:

(1) Create the skeleton for the given model. If the model has skeleton itself, the step can be omitted.

(2) Select the sampling skeleton points interactively. Create the meshless discretization.

(3) As pre-computing process, calculate linear elastic stiffness matrix $\boldsymbol{K}$.

(4) Calculate the skeleton location based on standard nonlinear FEM and get the sample points rotations presented by corresponding quaternion.

(5) Following our diffusion method, calculate

rotation quaternion of every node, about which its detail is in Section 5. Then, implement Step 6 in statics simulation or Step 7 in dynamics simulation.

(6) As for statics simulation, solve the governing Eq.(2) while apply the stiffness warping.

(7) As for dynamics simulation, update the whole stiffness matrix $\boldsymbol{K}$ according to Eq.(11) and rotation correlative force $\boldsymbol{F}_0 = -\sum \boldsymbol{R}_i \boldsymbol{K}_{ij} \boldsymbol{X}_j$. Then solve the following equation derived from Eq.(4)

$$(\boldsymbol{M} + \Delta t\boldsymbol{C} + \Delta t^2 \boldsymbol{K})\boldsymbol{v}^{i+1} = \boldsymbol{M}\boldsymbol{v}^i - \Delta t(\boldsymbol{K}\boldsymbol{x}^i + \boldsymbol{F}_0 - \boldsymbol{f}_{\text{ext}}), \tag{15}$$

update $\boldsymbol{x}^{i+1}=\boldsymbol{x}^i+\Delta t\boldsymbol{v}^{i+1}$. Unless the simulation is over, go to Step 4, or else, finish.

In order to test and verify our method, we implement all simulating programs in Visual C++. Fig.1 shows deformation of a Cube model with one joint where (a) is an initial model, (b) to (d) is respectively the result of rotating 45°, 90° and 110°. The results in Fig.1 are similar to those of Müller's method and show the feasibility of our method. Compared with Capell's simulation, which requires the user to set weight values for every part in the joint area and blend them, our method can be automatically implemented without users' intervention. We also edit the Cube model twisted around the skeleton as shown in Fig.2.

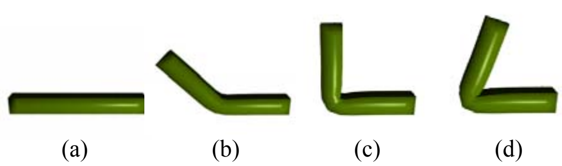

Fig.1 The bend deformation of Cube model. (a) The initial model; (b)~(d) The result of rotating 45°, 90° and 110° respectively

Fig.2 The twist deformation editing of Cube model. (a) The initial model; (b)~(d) The result of twisting 45°, 90° and 110° respectively

Fig.3 is a dynamics deformation simulation of the Dino model, where (a) is the initial model, (b) expresses the skeleton and sample points, (c) to (d) are a series of results. In the simulation, the Dino model comprises 2949 nodes in which 64 nodes are sample skeleton points. Every step costs 55 ms with our methods on the PC with PIV 2.4 G CPU and 512 RAM. Thirty-five ms are required to solve the dynamics equation in it. Moreover, we also do the same experiment applying Müller's method of calculating rotation in finite element methods with the results being similar to ours even though the simulation needs smaller time-steps.

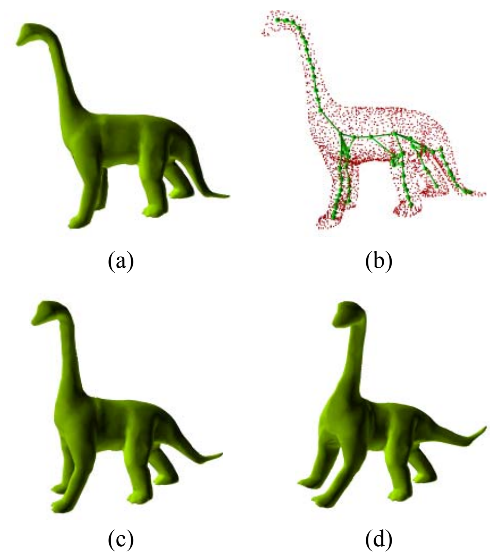

Fig.3 The deformation of Dino model. (a) The initial model; (b) expresses the skeleton and sample points; (c) and (d) are two deformation results

Fig.4 is an example in which our method is applied to simulation of complex model deformation. The Amodilio model has 21943 nodes and 89 sample skeleton points. Every time-step in average costs 1450 ms. Compared with the method applied by Zhou *et al.*(2005), our simulation is based on physical rule and has a more realistic effect. Moreover, our simulation method can easily control material property.

CONCLUSION AND FUTURE WORK

In this paper, an implementation for simulating skeleton driven elastic deformation is provided in meshless methods framework. In the course of simulation, a diffusion based method of calculating nodes rotation is proposed. Experiments showed it has three advantages. First, compared with the previous method,

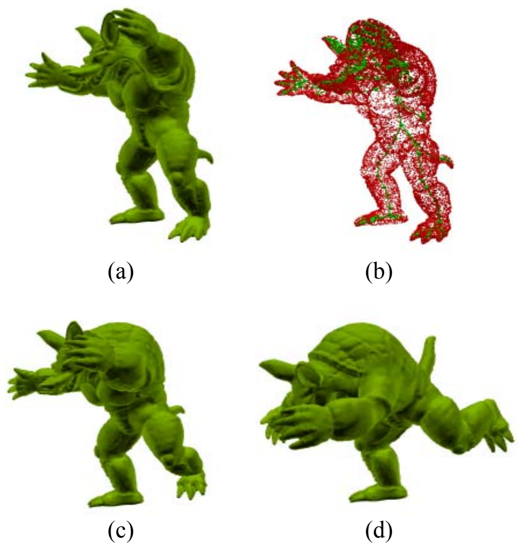

Fig.4 Dynamics deformation of Amodilio model. (a) The initial model; (b) expresses the skeleton and sample points; (c) to (d) are two deformation results

such as Müller's, the method can avoid errors accumulation in the dynamics and fussy sub-steps calculation in statics simulation by iterative calculation. Second, compared with Capell's method for joint area deformation, the method does not need users' intervention; finally, the method can increase stability whether in dynamics and in statics.

It must be pointed out that our system for large deformation is a simple nonlinear calculation based small stain (i.e. Cauchy strain) as that of Müller's method. So in some areas, such as joint area, the results are not perfect or sensitive for various material property. The problem will be discussed in future work so that more realistic results can be obtained.

Collision detection and handling is not involved in this paper. In fact, we can use the method described in (Capell *et al*., 2005) to deal with collision problem because more statics prediction can be given with our rotation calculation method. As for this problem, we will make further researches.

References

Baraff, D., Witkin, A., 1998. Large Steps in Cloth Simulation. Proc. of ACM SIGGRAH 1998, p.43-54.

Belytschko, T., Lu, Y.Y., Gu, L., 1994. Element free Galerkin methods. *Int'l J. Numerical Methods in Eng.*, **37**(2):229-256. [doi:10.1002/nme.1620370205]

Capell, S., Green, S., Gurless, B., Duchamp, T., Poovic, Z., 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. on Graph.*, **21**(3):586-593.

Capell, S., Burkhart, M., Curless, B., Duchamp, T., Popovic, Z., 2005. Physically Based Rigging for Deformable Characters. SCA'05: Pro. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, New York, USA, p.301-310.

Desbrun, M., Cani, M.P., 1995. Animating Soft Substances with Implicit Surfaces. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM SIGGRIPH 1995, p.287-290. [doi:10.1145/218380.218456]

Gibson, S.F., Mirtich, B., 1997. A Survey of Deformable Models in Computer Graphics. Technical Report TR-97-19, MERL—A Mitsubishi Electric Research Laboratory, Cambridge, Massachusetts.

Guo, X.H., Qin, H., 2005. Real-time meshless deformation. *Computer Animation and Virtual Worlds*, **16**(3-4):189-200. [doi:10.1002/cav.98]

Guo, X.H., Li, X., Bao, Y.F., Gu, X.F., Qin, H., 2006. Meshless thin-shell simulation based on global conformal parameterization. *IEEE Trans. on Visualization and Computer Graphics*, **12**(3):375-385. [doi:10.1109/TVCG.2006.52]

Lin, H., Zhang, H.X., Bao, H.J., 2005. Navigation Path Extraction for Virtual Endoscopy by Shape-sensitive Voronoi-diagram Filtering. Proceedings of CGIV 2005.

Müller, M., Gross, M., 2004. Interactive Virtual Materials. Proceedings of the 2004 Conference on Graphics Interface, p.239-246.

Müller, M., Dorsey, J., McMillan, L., Jagnow, R., Cutler, B., 2002. Stable Real-time Deformations. Proceeding of the 2002 ACM SIGGRAPH Symposium on Computer Animation, p.54-59,189.

Nealen, A., Müller, M., Keiser, R., Boxeman, E., Carlson, M., 2005. Physically Based Deformable Models in Computer Graphics. EUROGRAPHICS 2005 State of the Art Report (STAR), Dublin, Ireland, p.289-301.

Pauly, M., Keiser, R., Adams, B., Dutré, P., Gross, M.H., Guibas, L.J., 2005. Meshless animation of fracturing solids. *ACM Trans. Graph.*, **24**(3):957-964. [doi:10.1145/1073204.1073296]

Terzopoulos, D., Platt, J., Barr, A., Fleischer, K., 1987. Elastically deformable models. *Computer Graphics*, **21**(4):205-214.

Thomas, S., Scott, P., 1986. Free form deformation of solid geometric models. *SIGGRAPH, Association of Computing Machinery*, **20**(4):151-159.

Yu, Y.Z., Zhou, K., Xu, D., Shi, X.H., Bao, H.J., Guo, B.N., Shum, H.Y., 2004. Mesh editing with Poisson-based gradient field manipulation. *ACM Trans. Graph.*, **23**(3):644-651. [doi:10.1145/1015706.1015774]

Zhou, K., Huang, J., Snyder, J., Liu, X.G., Bao, H.J., Guo, B.N., Shum, H.Y., 2005. Large mesh deformation using the volumetric graph Laplacian. *ACM Trans. Graph.*, **24**(3):496-503. [doi:10.1145/1073204.1073219]