1000-9825/2002/13(01)0015-09 ©2002 Journal of Software 软 件 学 报 Vol.13, No.1

Interactive Shape Deformation of *n*-Sided Surfaces*

ZHENG Jin-jin, ZHANG Jian-jun

(*National Centre for Computer Animation, Bournemouth University, Poole, Dorset BH*12 5*BB, UK*)

E-mail: jj_zheng@lycos.com; jzhang@bournemouth.ac.uk

http://ncca.bournemouth.ac.uk/main/staff/jian

Received March 1, 2001; accepted September 27, 2001

Abstract: An efficient physically based surface sculpting method is presented in this paper for the interactive deformation of *n*-sided surfaces. By minimizing an energy functional, the user is able to deform a surface by applying different forms of forces directly, acting as virtual sculpting tools. The user is also able to define necessary geometric constraints, so as to further control the surface shape. Compared with the traditional method that a surface is deformed by moving the control points, this method is much more intuitive and still is very efficient.

Key words: physically based deformation; *n*-sided surfaces; control points; virtual sculpting; geometric constraints

1 Introduction

Interactive sculpting of surfaces is an important subject of computer graphics and geometric modeling[1]. Due to the wide applications of parametric surfaces, such as Bézier and B-spline surfaces, shape control is conventionally performed through the use of the control points. However, deforming or sculpting a complex object is often not trivial, if the control points are used alone. Indeed it can be extremely tedious and time-consuming, if not impossible. Many researchers have devoted a great amount of effort to make such a task more intuitive. The free-form deformations (FFDs), developed by Sederberg and Parry[2] was one of the earliest attempts along this line. It still is the most widely used deformation tool implemented by many geometric modeling and animation packages. This method was later improved by Hsu *et al*[3]. to allow a direct manipulation of the geometry. Using this method, the user is able to pick up a point on a surface and move it to a new location directly. The manipulation of the control lattice is calculated automatically. Other approaches, such as those proposed by Fowler & Bartels[4], share the same idea that the user can deform the geometry directly rather than through the use of the control points. These direct manipulation methods, despite being useful in many applications, are purely geometric, i.e. no physical properties and their effects on the surfaces or objects are considered.

Physically based surface sculpting techniques are different in that the physical properties are closely linked to the outcome. Usually an energy functional is formulated on the assumption that a surface or an object reaches the

---

* Supported by the Startup Scientific Research Fund for Returned Scholars from the Chinese Education Ministry (国家教育部留学回国人员科研启动基金); the Research Fund for Excellent Returned Scholars from the Chinese Academy of Sciences (中国科学院留学经费择优支持回国工作基金); the Post-Doctoral Research Fund from Wang Kuang Cheng Educational Foundation (王宽诚博士后工作奖励基金)

ZHENG Jin-jin was born in 1963. He is a research fellow at the National Center for Computer Animation, Bournemouth University, UK. He received a Ph.D. degree in computer aided geometric modeling from the University of Birmingham, UK in 1998. His research interests include CAGD and computer animation. **ZHANG Jian-jun** was born in 1959. He is a professor in Computer Graphics and Director of Research Projects at the National Center for Computer Animation, Bournemouth University, UK. He received a Ph.D. degree from Chongqing University in 1987. His research interests include computer graphics, CAD and computer animation.

rest shape when its overall energy becomes minimum. Based on this understanding, a surface patch can be sculpted by applying synthetic forces directly on it, as if they were from some kind of sculpting tools. Below this intuitive user-interaction layer, the new positions of the control points of the surface patch are automatically computed. Several methods were proposed by a number of researchers including Chui & Lai[5], Essa *et al.*[6], and Fowler & Bartels[4]. Understandably, the introduction of physics laws makes the surface sculpting process more flexible and more intuitive.

Although there have been a number of physically based surface sculpting approaches reported in the literature, they are all concerned with surfaces composed of regular 3 or 4-sided patches, such as B-spline or NURBS surfaces[7~11]. Physically based modeling of irregular *n*-sided surfaces is still a gap, in spite of their indispensable position in geometric modeling, computer aided design and computer animation[12~17].

In this paper, we are proposing such a method, so that physical properties can be assigned to any irregular *n*-sided surface patches. With our method, the user will be able to deform an irregular patch using different forms of synthetic forces. The update of the control points of the patch will be calculated automatically by minimizing the overall surface energy. This method will also automatically satisfy the geometric constraints, which include the points and curves the surface patch has to pass through.

Both these constraints and the forces can be used as user-interface handles for surface sculpting. Compared with the conventional approach of directly moving the control points, the method presented here produces more natural deformations and are more intuitive to operate. It can move a number of control points simultaneously to achieve a desirable surface shape.

2 *n*-Sided Patches and Energy Functional

An *n*-sided control point surface of degree *m* can be defined as follows

$$\mathbf{r}(\boldsymbol{u}) = \sum_{j=0}^{[m/2]} \sum_{\min\boldsymbol{\lambda}=j} B_{\lambda}(\boldsymbol{u})\mathbf{r}_{\lambda} \tag{1}$$

where $\lambda=(\lambda_1,\lambda_2,...,\lambda_n)$ represents the *n*-ple subscripts. $\boldsymbol{u}=(u_1,u_2,...,u_n)$ represents the *n* parameters of which only two are independent. $\mathbf{r}_{\lambda}$ stands for a control point in 3D-space $\mathbb{R}^3$, as shown in Fig.1. $B_{\lambda}(\boldsymbol{u})$ is the associated basis function. The total number *l* of the control points is given by

$$l=\begin{cases}\dfrac{n(m+2)m}{4}+1, & m \text{ even}\\[2mm] \dfrac{n(m+1)^2}{4}, & m \text{ odd}\end{cases} \tag{2}$$

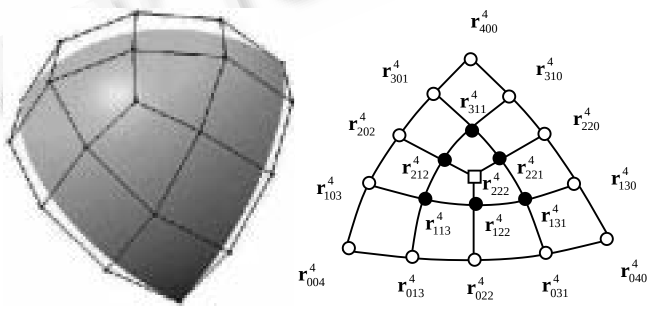

Fig.1 Control points and 3-sided surface *m*=4

For a detailed description of the *n*-sided control point surfaces, the reader is referred to Refs.[16,17].

The final deformed shape of a surface is the result of minimizing its potential energy. Based on the energy model of a tensioned thin plate[8,10,18], the following energy functional for an n-sided surface is suggested in this work

$$E = E_1 - E_2 \tag{3}$$

$$E_1 = \iint \left[\alpha\left(\left\|\mathbf{r}_u(\boldsymbol{u})\right\|^2 + \left\|\mathbf{r}_v(\boldsymbol{u})\right\|^2\right) + \beta\left(\left\|\mathbf{r}_{uu}(\boldsymbol{u})\right\|^2 + 2\left\|\mathbf{r}_{uv}(\boldsymbol{u})\right\|^2 + \left\|\mathbf{r}_{vv}(\boldsymbol{u})\right\|^2\right)\right]\mathrm{d}u\mathrm{d}v \tag{4a}$$

$$E_2 = \iint 2\mathbf{r}(\boldsymbol{u}) \cdot \boldsymbol{f}(\boldsymbol{u})\mathrm{d}u\mathrm{d}v \tag{4b}$$

where u and v are two independent variables. E_2 represents the energy of the surface itself and its natural resistance to deformations. It is expressed as a weighted sum of its stretching and bending terms. α and β are non-negative and otherwise freely chosen coefficients. E_2 represents the energy resulting from the applied forces. A force vector $\boldsymbol{f}(\boldsymbol{u})$ is introduced here as a sculpting tool for the deformation of the n-sided surface.

The surface expression (1) can be rephrased as

$$\mathbf{r}(\boldsymbol{u}) = \mathbf{V}^T\mathbf{Z}, \tag{5}$$

where $\mathbf{V}$ is the column vector of all control points and $\mathbf{Z}(\boldsymbol{u})$ is a column vector of all the Zheng & Ball[16] basis functions.

Inserting (5) into the integral (4a) results in the following quadratic form

$$E_1 = \mathbf{V}^T\mathbf{K}\mathbf{V}, \tag{6}$$

where E_1 is called the fairness norm. $\mathbf{K}$, an $l \times l$ symmetric matrix, is called the stiffness matrix, whose entries are calculated by evaluating the integral (4a). This is given by

$$\mathbf{K} = \alpha\,\mathbf{K}_1 + \beta\,\mathbf{K}_2 \tag{7}$$

$$\mathbf{K}_1 = \iint\left(\mathbf{Z}_u\mathbf{Z}_u{}^T + \mathbf{Z}_v\mathbf{Z}_v{}^T\right)\mathrm{d}u\mathrm{d}v \tag{8a}$$

$$\mathbf{K}_2 = \iint\left(\mathbf{Z}_{uu}\mathbf{Z}_{uu}^T + 2\mathbf{Z}_{uv}\mathbf{Z}_{uv}^T + \mathbf{Z}_{vv}\mathbf{Z}_{vv}^T\right)\mathrm{d}u\mathrm{d}v \tag{8b}$$

Once the degree, m, of an n-sided surface is given, the entries of the matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are constants evaluated from (8).

Substituting (5) into (4b), we have

$$E_2 = 2\mathbf{V}^T\mathbf{F} \tag{9}$$

where $\mathbf{V}$ is the vector of the control points. And

$$\mathbf{F} = \iint \mathbf{Z} \cdot \boldsymbol{f}(\boldsymbol{u})\mathrm{d}u\mathrm{d}v \tag{10}$$

is a column vector, called the force vector, whose entries are vectors in $\mathbb{R}^3$ taking the form $\iint B_\lambda(\boldsymbol{u})\boldsymbol{f}(\boldsymbol{u})\mathrm{d}u\mathrm{d}v$.

Combining (6) and (9), we have the following complete expression of the energy functional

$$E = \mathbf{V}^T\mathbf{K}\mathbf{V} - 2\mathbf{V}^T\mathbf{F} \tag{11}$$

Eq.(11) is a quadratic form with respect to the control point vectors $\mathbf{V}$. Minimising (11) gives the following equation:

$$\mathbf{K}\mathbf{V} = \mathbf{F} \tag{12}$$

The solution to (12) is a set of control point vectors $\mathbf{V}$ used to generate the deformed n-sided surface.

3 User Interaction Handles——Physical Forces

In the expression of the force vector

$$\mathbf{F}=\iint \mathbf{Z}\cdot \boldsymbol{f}(\boldsymbol{u})\mathrm{d}u\mathrm{d}v \tag{13}$$

$\mathbf{Z}(\boldsymbol{u})$ is a column vector of all the Zheng & Ball[25] basis functions. $\boldsymbol{f}(\boldsymbol{u})$ is the net effect of all the applied forces, which may be spring forces, repulsion forces, gravitational forces, inflation forces, etc. as discussed in Refs. [7,10,19,20]. The initial shape of the n-sided surface is defined by the control point vectors $\mathbf{V}_0=(\ldots,\mathbf{r}_\lambda^0,\ldots)^T$ and the force density $\boldsymbol{f}_0(\boldsymbol{u})$ of the distribution forces. Then by (12), we have

$$\mathbf{K}\mathbf{V}_0=\mathbf{F}_0 \tag{14}$$

where $\mathbf{F}_0=\iint \mathbf{Z}\cdot \boldsymbol{f}_0(\boldsymbol{u})\mathrm{d}u\mathrm{d}v$.

As a set of forces $\boldsymbol{f}_0(\boldsymbol{u})$ makes the n-sided surface take a particular shape in the space $\mathbb{R}^3$, when we change these forces to a new value $\boldsymbol{f}_1(\boldsymbol{u})=\boldsymbol{f}_0(\boldsymbol{u})+\Delta\boldsymbol{f}(\boldsymbol{u})$, the control point vectors will move to new locations $\mathbf{V}=(\ldots,\mathbf{r}_\lambda,\ldots)^T$. Inserting it into (13), we have

$$\mathbf{F}=\iint \mathbf{Z}\cdot\left(\boldsymbol{f}_0(\boldsymbol{u})+\Delta\boldsymbol{f}(\boldsymbol{u})\right)\mathrm{d}u\mathrm{d}v \tag{15}$$

Considering (13) and (15) gives

$$\mathbf{K}\mathbf{V}=\iint \mathbf{Z}\cdot\left(\boldsymbol{f}_0(\boldsymbol{u})+\Delta\boldsymbol{f}(\boldsymbol{u})\right)\mathrm{d}u\mathrm{d}v$$

or

$$\mathbf{K}\mathbf{V}=\mathbf{F}_0+\iint \mathbf{Z}\cdot\Delta\boldsymbol{f}(\boldsymbol{u})\mathrm{d}u\mathrm{d}v \tag{16}$$

$\Delta\boldsymbol{f}(\boldsymbol{u})$ is the density of the distribution forces to be applied to the initial n-sided surface. Once $\Delta\boldsymbol{f}(\boldsymbol{u})$ is given, a new set of control points $\mathbf{V}$ will arrive by the resolution of the linear system (16).

There are a number of different forms of forces we can use for sculpting a surface patch. In the following sections, we are discussing both the forms and their applications of the applied forces.

3.1 Discrete forces

For a force applied on a surface point $\mathbf{r}(\boldsymbol{u}_0)$, $\Delta\boldsymbol{f}(\boldsymbol{u})$ can be repressed as

$$\Delta\boldsymbol{f}(\boldsymbol{u})=\boldsymbol{f}(\boldsymbol{u}-\boldsymbol{u}_0)\delta(\boldsymbol{u}-\boldsymbol{u}_0) \tag{17}$$

where $\delta(u-u_0,v-v_0)$ is the delta function.

A simple example is to connect a surface point $\mathbf{r}(\boldsymbol{u}_0)$ to a space point $\mathbf{P}$ with an ideal Hookean spring whose stiffness is k. Then we have

$$\Delta\boldsymbol{f}(\boldsymbol{u})=k\left(\mathbf{P}-\mathbf{r}(\boldsymbol{u}_0)\right)\delta(\boldsymbol{u}-\boldsymbol{u}_0) \tag{18}$$

3.2 Continuous forces

For a force applied on an area of the n-sided surface $\mathbf{r}(\boldsymbol{u})$, $\Delta\boldsymbol{f}(\boldsymbol{u})$ is expressed as

$$\Delta\boldsymbol{f}(\boldsymbol{u})=\begin{cases}\boldsymbol{f}(\boldsymbol{u}) & (u,v)\in D\\ 0 & (u,v)\notin D\end{cases} \tag{19}$$

where $D\subset\mathbb{R}^2$ is an area in the domain of surface $\mathbf{r}(\boldsymbol{u})$.

In practice, a force as a sculpting tool can be applied on an area surrounding a surface point $\mathbf{r}(\boldsymbol{u}_0)$ with radius ρ. For such area forces, $\Delta\boldsymbol{f}(\boldsymbol{u})$ can be expressed by (see Fig.2):

$$\Delta\boldsymbol{f}(\boldsymbol{u})=\left(a_2\rho^2+a_1\rho+a_0\right)\mathbf{n}_0$$

$$(u,v)\in\{(u,v)\ \|\mathbf{r}(u,v)-\mathbf{r}(u_0,v_0)\|\le\rho\} \tag{20}$$

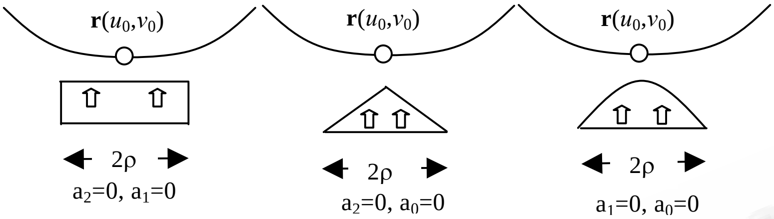

Fig.2 Alternative forces applied

4 Geometric Constraints

Using synthetic forces as a sculpting tool can be nicely complimented by linear constraints. Constraints specify conditions that the surface has to satisfy and therefore provide extra user-interface tools for surface deformation. Such constraints usually include interpolated points, curves and surface normal constraints[7,19].

The process of constraint satisfaction leads to the determination of some unknowns. In another word, this process reduces the degrees of freedom (DOF) of the whole system, and hence leaves fewer DOF for further manipulation. Considering the quadratic system (11), the solution to the constraints imposed on the surface will result in a smaller unconstrained system, of similar nature. Thus proper treatment of these constraints allows faster deformation to be produced.

We suppose the number of constraints is less than the number of variables. Linear constraints are generally expressed as

$$A\mathbf{V}=b \tag{21}$$

where $\mathbf{V}$ is the vector of l variables (degree of freedom). A is a $k\times l$ matrix of coefficients, k is the number of constraints ($k\le l$). If $k>l$, there are redundant constraints in (21). Each row of the matrix A represents a linear constraint on $\mathbf{V}$.

By solving system of constraints (21), the variable vector $\mathbf{V}$ will be transformed into a new one $\mathbf{W}$ where the number of variables is reduced from l to j. Here $j=l\text{-rank}(A)\le l$.

Suppose $\boldsymbol{g}_i$ ($i=1,2,\ldots,j$) is a set of basis vectors in the null space of Eq.(21). $\overline{\mathbf{V}}_0$ is a particular solution for Eq.(21). Then the general solution to (21) can be given by

$$\mathbf{V}=\sum_{i=1}^{j} w_i\boldsymbol{g}_i+\overline{\mathbf{V}}_0,\ \text{for an arbitrary real value } w_i$$

We denote $G=(\boldsymbol{g}_1,\boldsymbol{g}_2,\ldots,\boldsymbol{g}_j)$, an $l\times j$ matrix formed by all basis vectors $\boldsymbol{g}_i$. $\mathbf{W}$ is a vector with j unconstrained variables w_i. Then we have

$$\mathbf{V}=G\mathbf{W}+\overline{\mathbf{V}}_0 \tag{22}$$

As we have seen, redundant constraints in system (21) are removed in this process.

Inserting Eq.(22) into the system (11), a new energy function with respect to the unconstrained variables $\mathbf{W}$ can be obtained as the following.

$$E=\mathbf{W}^T(G^TKG)\mathbf{W}+2\mathbf{W}^T(G^TK\overline{\mathbf{V}}_0-G^TF)+(\overline{\mathbf{V}}_0{}^TK\overline{\mathbf{V}}_0-2\overline{\mathbf{V}}_0{}^TF)$$

This is also a quadratic form with respect to **W**. Minimising it leads to the following solution

$$G^T K G\mathbf{W} = G^T F - G^T K\overline{\mathbf{V}}_0$$

Inserting Eq.(15) into it, we have

$$G^T K G\mathbf{W} = G^T F_0 - G^T K\overline{\mathbf{V}}_0 + G^T \iint \mathbf{Z} \cdot \Delta f(\boldsymbol{u})\mathrm{d}u\mathrm{d}v \tag{23}$$

Solving Eq.(23) gives a new set of control points for the resulting deformed surface.

A special case: if no constraints exist, G is an identity matrix and therefore **V** = **W** in Eq.(22). System (23) is identical to system (16).

In practice, the constraints may be:

- Moving a surface point to a new position. Suppose a surface point $\mathbf{r}(\boldsymbol{u}_0) = \sum_{\lambda} B_{\lambda}(\boldsymbol{u}_0)\mathbf{r}_{\lambda}^0$ is to be moved to a new position $\mathbf{r}^1$, where $\mathbf{r}_{\lambda}^0$ are the initial control points of the surface. Then, the following constraint must be added into Eq.(21).

$$\sum_{\lambda} B_{\lambda}(\boldsymbol{u}_0)\mathbf{r}_{\lambda} = \mathbf{r}^1 \tag{24}$$

  That means a row of the matrix A corresponds to Eq.(24).

- Preserving surface points. In the above case, if the new position is identical to the original point $\mathbf{r}(\boldsymbol{u}_0) = \sum_{\lambda} B_{\lambda}(\boldsymbol{u}_0)\mathbf{r}_{\lambda}^0$, then the following constraint must be added into Eq.(21).

$$\sum_{\lambda} B_{\lambda}(\boldsymbol{u}_0)\mathbf{r}_{\lambda} = \mathbf{r}(\boldsymbol{u}_0) \tag{25}$$

- Preserving surface curves. Suppose $\mathbf{r}(\boldsymbol{t}(s)) = \sum_{\lambda} B_{\lambda}(\boldsymbol{t}(s))\mathbf{r}_{\lambda}^0$ is a curve on the initial surface, $\boldsymbol{t}(s) = \boldsymbol{t}(u(s), v(s))$. If the surface interpolates the curve, following Ref.[7], the following constraints are added into Eq.(21):

$$\tilde{A}\mathbf{V} = \tilde{b} \tag{26}$$

A row of $\tilde{A}$ is given by

$$A_{\eta} = \int \left(\sum_{\lambda} B_{\lambda}(\boldsymbol{t}(s))\mathbf{r}_{\lambda} \right) B_{\eta}(\boldsymbol{t}(s))\mathrm{d}s \tag{27a}$$

The corresponding entry in the vector $\tilde{b}$ is

$$b_{\eta} = \int \left(\sum_{\lambda} B_{\lambda}(\boldsymbol{t}(s))\mathbf{r}_{\lambda}^0 \right) B_{\eta}(\boldsymbol{t}(s))\mathrm{d}s \tag{27b}$$

For each control point, there is a constraint equation in (26). Typically, as mentioned in Ref.[7], most of these constraints are redundant. In the procedure of calculating the matrix G, the redundant constraints can be reduced.

5 Calculation of Control Points

In this section, let us discuss the solvability of system (23).

In the literature, it was assumed that there were enough degrees of freedom in **V** to satisfy a constrained Eq.(21). In fact, this may not be true if too many constraints are introduced into the system. One way to increase the degrees of freedom in **V** is by degree elevation, which leaves enough DOF to satisfy the constraints in Eq.(21).

The symmetric stiff matrix K is semi-positive definite rather than positive definite. It is possible that the linear system (23) is not solvable. In this case, a least square solution to system (23) should be used instead.

The calculation of the stiff matrix is a one-off task. It should not be considered as a sculpting cost. For each pair of n and m, we need only calculate it once. Therefore once the degree and number of sides of a surface are known, surface deformation can be achieved very quickly. In addition, by using the symmetry of the n-parameters as outlined below, the computation is even more efficient:

(1) Setting $u_1 = u_2 = \cdots = u_n$, we can get the central point in the parametric domain.

(2) In the domain of definition, subdivide the patch by connecting the central point with the middle point of each edge. This subdivides an n-sided patch into n 4-sided areas.

(3) In each area, we integrate (8) using Gaussian formulae.

6 Implementation

We have implemented the above-described method on a PC workstation using the OpenGL library to render the surfaces. Although we haven't calculated the exact sculpting time in the following examples, it won't take a second to sculpt the models in a PIII800 PC workstation due to the fact that the stiff matrix is already prepared before the sculpting procedure.

The procedure of surface deformation is shown as in Fig.3. Figure 3 (left) shows an initial 5-sided planar patch. In Fig.3 (middle) three upward forces and one downward force are applied on four surface points simultaneously as indicated by the arrows. Two circular points are preserved during the deformation of the surface. In Fig.3 (right), the forces are applied once more to make the surface deform further.

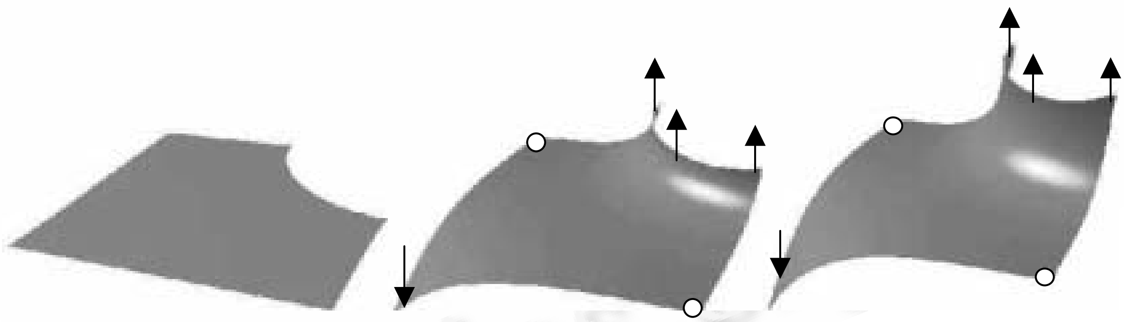

Fig.3 (Left) Initial 5-sided patch. (Middle and right) Deformed 5-sided patch. Arrows show the forces applied on surface points. Circular points kept fixed during deforming

The physically based method has also been applied to a model shown in Fig.4. Figure 4 (top) shows an initial model with 3- and 5-sided patches. As shown in Fig.4 (bottom), two forces are applied on two surface points (not control points) on the 5-sided surface patch. Arrows indicate the applied forces on surface points. While the 5-sided patch is deformed, its boundary curves and the surface normal along these curves are preserved.

In Fig.5, a diagram is given to illustrate the procedure of the algorithm of surface deformation.

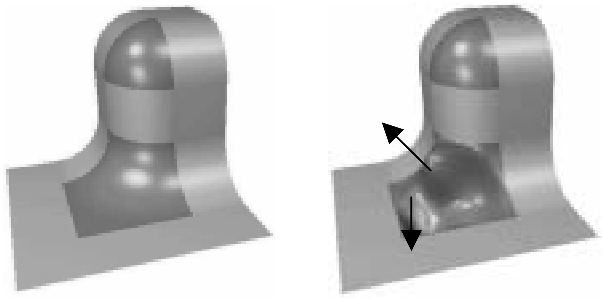

Fig.4 A model with 3- and 5-sided surface patches (left). Deformed model (right). Arrows show the forces applied on surface points

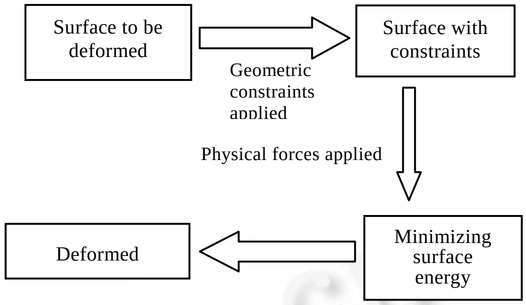

Fig.5 Procedure of surface deformation

7 Conclusions and Discussion

In this paper, we have presented an efficient method for interactively sculpting irregular *n*-sided surfaces. By minimizing an energy functional, the control points of an *n*-sided surface are automatically computed. The applied forces and necessary geometric constraints can be used as user interface handles, and it is therefore much more intuitive and meaningful than direct moving the control points.

Different types of forces are discussed and their expressions are given, which can be used as effective sculpting tools.

Further research could involve the investigation of an efficient algorithm for the computation of the force vectors when continuous forces are applied.

Acknowledgement The first author would like to gratefully acknowledge the financial support from the Startup Scientific Research Fund for Returned Scholars from the Chinese Education Ministry, the Research Fund for Excellent Returned Scholars from the Chinese Academy of Sciences, and the Post-doctoral Research Fund from Wang Kuang Cheng Educational Foundation.

References:

[1] Jeng, E., Xiang, E. Moving cursor plane for interactive sculpting. ACM Transactions on Graphics, 1996,15(3):211~223.

[2] Sederberg, T., Parry, S. Free-Form deformation of solid geometric models. In: Proceedings of the SIGGRAPH Annual Series. 1986. 151~160.

[3] Hsu W.M., Hughes J.F. Kaufman, H. Direct manipulation of free-form deformations. In: Proceedings of the SIGGRAPH, Annual Series. Chicago: ACM Press, 1992. 261~268.

[4] Fowler, B. Bartels, R. Constrained based curve manipulation. IEEE Computer Graphics & Applications, 1993,13(5):43~49.

[5] Chui, C.K., Lai, M.J. Filling polygonal holes using C^1 cubic triangular spline patches. CAGD, 2000,17:297~307.

[6] Essa, A., Sclarof, S., Pentland, A. A unified approach for physical & geometric modeling for graphics & animation. Eurographics'92, 1992,11(3):127~138.

[7] Celniker, G., Welch, W. Linear constraints for deformable B-spline surfaces. In: Proceedings of the 1992 Symposium on Interactive 3D Graphics. Massachusetts: ACM Press, 1992. 165~170.

[8] Qin, H., Terzopoulos, D. Dynamic NURBS Swung surfaces for physics-based shape design. Computer Aided Design, 1995,27(2): 111~127.

[9] Qin, H., Terzopoulos, D. Dynamic manipulation of triangular B-splines. In: Proceedings of the 3rd Symposium on Solid Modeling and Applications. Salt Lake City: ACM Press, 1995. 351~360.

[10] Terzopoulos, D., Qin, H. Dynamic NURBS with geometric constraints for interactive sculpting. ACM Transactions on Graphics, 1994,13(2):103~136.

[11] Vassilev, T. Interactive sculpting with deformable nonuniform B-spines. Computer Graphics Forum, 1997,16(4):191~199.

[12] Ball, A.A., Zheng, J.J. Degree elevation for *n*-sided surfaces. CAGD, 2000,18(2):135~147.

[13] Gregory, J.A., Zhou, J. Filling polygonal holes with bi-cubic patches. CAGD, 1994,11:391~410.

[14] Sabin M.A. A symmetric domain for a 6-sided patch. In: Bowyer, A. ed., Computer-Aided Surface Geometry and Design——The Mathematics of Surfaces IV. Oxford: Clarendon Press, 1994. 185~193.

[15] Volpin, O, Bercovier, M., Matskewich, T. A comparison of invariant energies for free-form surface construction. The Visual Computer, 1999,15(4):199~210.

[16] Zheng, J.J., Ball, A.A. Control point surfaces over non- four-sided areas. Computer Aided Geometric Design, 1997,14:807~820.

[17] Zheng, J.J. The *n*-sided control point surfaces without twist vectors. CAGD, 2001,18(2):129~134.

[18] Halstead, M., Kass, M., DeRose, T. Efficient fair interpolation using Catmull-Clark surfaces. In: Proceedings of the SIGGRAPH Annual Series. California: ACM Press, 1993. 35~44.

[19] Szeliski, R., Tonnesen, K. Surface modeling with oriented particle systems. Computer Graph, 1992,26(2):185~194.

[20] Terzopoulos, D., Fleischer, K. Deformable models. Visual Computer, 1988,4(6):306~331.

[21] Hu, S., Li, Y., Ju, T., *et al*. Modifying the shape of NURBS surface with geometric constraints. Computer Aided Design, 2001,33(12):903~912.

[22] Hu, S., Zhu, X., Sun, J. Shape modification of NURBS surfaces via constrained optimization. Journal of Software, 2000,11(12):1567~1571 (in Chinese).

[23] Lin, J., Ball, A.A., Zheng, J.J. Surface modeling and mesh generation for simulating superplastic forming. Journal of Materials Processing Technology, 1998,80~81:613~619.

[24] Piegl, L. Modifying the shape of rational B-splines (2 Surfaces). Computer Aided Design, 1989,21(9):538~546.

[25] Piegl, L. Modifying the shape of rational B-splines (1 Curves). Computer Aided Design, 1989,21(8):509~518.

[26] Zhang, P., Cheng, F. Constrained scaling of trimmed NURBS surfaces based on fix-and-stretch approach. Computer Aided Design, 2001,33(1):103~112.

[27] Yan, W., Larson, P. Eager aggregation and lazy aggregation. In: Dayal, U., Gray, P., Nishio, S., eds. Proceedings of the 21st International Conference of Very Large Data Bases. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1995. 345~357.

[28] Tan, S.T., Wang, T.N., Zhao, Y.F., *et al*. A constrained finite element method for modeling cloth deformation. Visual Computer, 1999,15(2):90~99.

附中文参考文献：

[22] 胡事民,朱翔,孙家广.基于约束优化的 NURBS 曲面形状修改.软件学报,200,11(12):1567~1571.

塑造 *n* 边形曲面的外形

郑津津，张建军

(英国国家计算机动画中心, Bournemouth University, Poole BH12 5BB,英国)

摘要: 提出了一种高效基于物理性质的算法来动态的塑造 *n* 边形曲面的外形.这种算法是基于曲面的物理性质.通过极小化一个能量泛函,用户能够直接使用不同形式的外力作为虚拟的塑造工具来改变 *n* 边形曲面的外形.用户也能够定义必要的几何约束来进一步控制曲面外形.与通常移动控制点的方法比较,这种基于物理性质的方法更直观和有效.

关键词: 基于物理性质的变形;*n* 边形曲面;控制点;虚拟塑造;几何约束

中图法分类号: TP391 文献标识码: A