

Constrained scaling of trimmed NURBS surfaces based on fix-and-stretch approach

C. Zhang, P. Zhang, F. Cheng*

Graphics and Geometric Modeling Lab, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA

Received 5 September 1999; revised 5 February 2000; accepted 16 March 2000

Abstract

A new method to scale a trimmed NURBS surface while holding the shape and size of specific features (trimming curves) unchanged is presented. The new method is fix-and-stretch based: the new surface is formed by fixing selected regions of the given trimmed NURBS surface that contain the trimming curves and stretching the remaining part of the surface to reach certain boundary conditions. The stretching process is performed using an optimization process to ensure that the resulting surface reflects the shape and curvature distribution of the scaled version of the given surface. The resulting surface maintains a NURBS representation and, hence, is compatible with most of the current data-exchange standards. The new approach is more robust than a previous, attach-and-deform based approach (Zhang P, Zhang C, Cheng F. Constrained shape scaling of trimmed NURBS surfaces. In: Proceedings of the 1999 ASME Design Theory and Methodology Conference. Las Vegas, Nevada, 1999) in that it can tolerate scaling factors of bigger values (up to 2 in some cases). The new approach also guarantees that the features remain exactly the same after scaling. Test results on several car parts with trimming curves and comparison with the previous approach are included. The quality of the resulting surfaces is examined using the highlight line model. The presented technique is important for integrating standard parts into a sculptured product. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Constrained scaling; Subdivision; Constrained deformation; Optimization; Highlight lines; Constrained stretching; Trimming curves; NURBS surfaces; Strain energy

1. Introduction

A surface design problem of special urgency to the design community is the *lack of constrained shape modification capabilities*, i.e. lack of tools/techniques that are capable of holding significant features of a model unchanged while globally or locally altering it. The altering process may involve scaling and/or deformation. Addressing and solving this problem would provide the design industry with the capability of globally or locally modifying an existing model in length, height, or width without affecting certain significant features and, consequently, avoiding an expensive redesign process.

Using scaling as an altering technique is common in design. The problem of *constrained shape scaling* (i.e. scaling a model with some features fixed), however, has not been seriously addressed in the literature yet. The only known result is an attach-and-deform based approach presented recently [3]. The new surface is formed by scaling the given surface according to the scaling require-

ment first, and then attaching the original features to the scaled NURBS surface at appropriate locations. The attaching process requires a minor deformation of the scaled surface through an optimization process to ensure complete attachment.

Another commonly used altering technique is deformation. The *free-form deformation* (FFD) method for surface design has been studied in several approaches. The *spatial deformation approach* operates on the space inside which the deformed objects are embedded. This approach is independent of the representation of the surface. Most works in this approach use a trivariate parametric volume. Deformation is performed by manipulating the control points of the trivariate volumes [4,5,10,12–16,20].

The *physics-based deformation approach* uses physical simulation to obtain realistic shapes and motions. This approach introduces a time variable into the surface representation to form a dynamic model. The behavior of the model is controlled by the physical laws, such as physical properties of mass distribution, tension, rigidity, damping and action of applied forces. The resulting surface is determined by the equilibrium state of the dynamic model [6,18,22–24]. This method is mainly used in computer

* Corresponding author. Tel.: +1-606-257-6760; fax: +1-606-323-1971.
E-mail address: cheng@cs.engr.uky.edu (F. Cheng).

animation and focuses on the process of transforming physical forces into changes of the dynamic model.

Constrained deformation (i.e. deforming a model while holding certain features of the model unchanged) was first studied by Celniker and Welch [7]. The purpose was to provide a modeling technique that separates the surface representation from the surface modeling operators. The user controls the surface by requiring the surface to preserve a set of geometric constraints while sculpturing it. The shape of the surface is faired by minimizing a global energy function. This technique has also been used in direct surface shape manipulation [26]. However, most of the time, it is used in surface interpolation and lofting, where a surface is designed to interpolate a curve net or scattered discrete points. The optimization process used in [3] is similar to Celniker and Welch's approach.

In this paper, we will present a new approach to the constrained shape scaling problem. The new approach is fix-and-stretch based: the new surface is formed by fixing selected regions of the given trimmed NURBS surface that contain the trimming curves and stretching the remaining part of the surface to reach certain boundary conditions. The stretching process is performed using an optimization process that leads to a system of linear equations. The new approach is more robust than the attach-and-deform based approach [3] in that: (1) it tolerates scaling factors of bigger values (up to 2 in some cases); and (2) it guarantees that the features remain exactly the same after scaling.

The remaining part of the paper is arranged as follows. A formal description of the problem is given in Section 2. The basic idea of the proposed method is presented in Section 3. Techniques needed in constructing the new surface are described in Sections 4–7. Test results of the proposed method and comparison of the new approach with the previous approach are shown in Section 8. Concluding remarks are given in Section 9.

2. Problem formulation

The problem of *constrained surface scaling* can be described as follows: given a NURBS surface $S(u, v)$ and a set of features C_i in the domain of the surface, construct a new surface $\tilde{S}(u, v)$ whose representation is a scaled version of the given surface $S(u, v)$ but carries all the original features $S \circ C_i$.

More specifically, let $S(u, v)$ be a NURBS surface of degree p in the u direction and degree q in the v direction

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n w_{i,j} Q_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{i,j} N_{i,p}(u) N_{j,q}(v)} \quad (1)$$

$$(u, v) \in [0, 1] \times [0, 1]$$

where $Q_{i,j}$ are the 3D control points, $N_{i,p}(u)$ and $N_{j,q}(v)$ are B-spline basis functions of degree p and q , respectively, and $w_{i,j}$ are weights. $N_{i,p}(u)$ and $N_{j,q}(v)$ are defined with respect to the knot vectors $\tau = \{\tau_0, \tau_1, \dots, \tau_{m+p+1}\}$, and $\sigma = \{\sigma_0, \sigma_1, \dots, \sigma_{n+q+1}\}$, respectively, with $\tau_0 = \dots = \tau_p = \sigma_0 = \dots = \sigma_q = 0$ and $\tau_{m+1} = \dots = \tau_{m+p+1} = \sigma_{n+1} = \dots = \sigma_{n+q+1} = 1$. The features to be held unchanged are closed trimming curves $S \circ C_i(t)$, $i = 1, 2, \dots, r$, where $C_i(t) = (u_i(t), v_i(t))$ are closed parametric curves defined in the domain of S with $S \circ C_i \cap S \circ C_j = \emptyset$ if $i \neq j$. All the trimming curves are inside the NURBS surface, they do not intersect the boundary of the surface. If the scaling factors in the x , y and z directions are S_x , S_y and S_z , respectively, then the new surface $\tilde{S}(u, v)$ is required to be as close to $T_s \circ S$ as possible, where $T_s \circ S$ is the scaled NURBS surface with T_s being a scaling matrix defined as follows:

$$T_s = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \quad (2)$$

The requirement that the new surface carries all the original features $S \circ C_i$ means that $S \circ C_i$ are also trimming curves of the new surface subject to some translation and rotation.

In industrial applications, a trimming curve of a free-form surface is usually represented as a linear polygon in the domain of the surface with vertices of the polygon being points of the curve. We follow the same approach in this work.

3. Basic idea

The main idea of this approach is to fix some selected regions of the given NURBS surface that contain the trimming curves while stretching the remaining part of the surface until a certain boundary condition is reached. The regions that need to be fixed during the stretching process have to be transformed to appropriate locations first.

The stretching process ensures that the shape and curvature distribution of the the resulting surface $\tilde{S}(u, v)$ are as close as possible to those of the scaled version of the given trimmed surface, $T_s \circ S(u, v)$, while carrying all the original features $S \circ C_i$. This is achieved by minimizing a shape-preserving objective function constructed based on the difference of these two surfaces. The resulting surface $\tilde{S}(u, v)$ is again a NURBS surface and it maintains the same boundary continuity condition with adjacent surfaces.

The main steps of our approach are shown below. The last step is for the user to visually examine the quality of the resulting surface using a highlight line model.

1. Subdividing surface $S(u, v)$;
2. Relocating control points;
3. Setting up *shape-preserving objective function*;

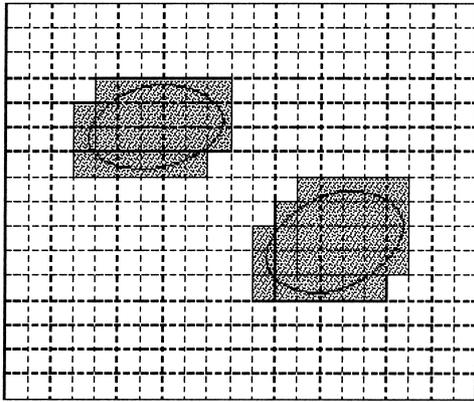


Fig. 1. Two trimming curves and their envelopes.

- 4. Performing *constrained surface stretching*;
- 5. Rendering.

Details of the above steps are given in the subsequent sections.

4. Subdivision of $S(u,v)$

This step recursively subdivides the surface $S(u,v)$ until two conditions are satisfied. We need to define three terms first.

Given a trimmed NURBS surface $S(u,v)$, the *u-extent* of a trimming curve $C_i(t)$ is the smallest interval in the *u* parameter space that contains the *u* components of the trimming curve $C_i(t)$. The *v-extent* of a trimming curve $C_i(t)$ can be defined similarly. The *envelop* of a trimming curve $C_i(t)$ is the smallest union of domain patches in the parameter space of the trimmed NURBS surface that contains the trimming curve. The set of control points needed to define the image of a trimming curve’s envelop is called a *control envelop*. Fig. 1 shows the envelopes of two trimming curves.

The two conditions to be satisfied by the recursive subdivision process are:

- 1. Each trimming curve has at least $p + 1 + i$ knots on each

side of its *u-extent* and $q + 1 + i$ knots on each side of its *v-extent* if the trimmed NURBS surface satisfies C^i -continuity with adjacent surfaces.

- 2. Control envelopes of different trimming curves do not overlap and they should be at least three control points apart in each direction.

These conditions are to ensure that the stretching process can be performed with enough flexibility. The second condition is satisfied if the envelopes of two trimming curves intersect the same *v-span* (*u-span*) then the envelopes in this *v-span* (*u-span*) are $2p$ *u-spans* ($2q$ *v-spans*) apart. The envelopes shown in Fig. 1 do not satisfy this condition.

It is possible to perform subdivision on boundary spans and spans that intersect the trimming curves only. This would reduce the subdivision time to a certain extent. However, the highlight line model of the stretching results show that the curvature distribution in this case is not as good as the results of uniform subdivision on all the spans. This is reasonable because a movement of a control point in a large patch causes shape change in a larger area.

Without the loss of generality, we shall use the same notations for the control points and parameter knots after the subdivision step even though both of them might have been changed during the subdivision process.

5. Relocating control points

The control points of a trimmed NURBS surface are divided into three categories. *Type one control points* are those that belong to one of the control envelopes. *Type three control points* are those that determine a boundary region of the trimmed surface. *Type two control points* are the ones between the type one and type three control points. Fig. 2 shows the partition of the control points of a trimmed NURBS surface. The width of type three control points depends on the degree of continuity of the trimmed surface with adjacent surfaces. For simplicity, we assume that the NURBS surface has multiple knots at the beginning and end of its knot vectors. In this case, the width of type three control points is one for C^0 -continuity and two for C^1 -continuity. Since the desired surface $\tilde{S}(u,v)$ is required to have the same trimming curves as the given trimmed NURBS surface $S(u,v)$ (subject to some translation and rotation), type one control points of the new surface $\tilde{S}(u,v)$ should be the same as type one control points of the given trimmed NURBS surface $S(u,v)$ (subject to some translation and rotation). On the other hand, the new surface should maintain the same boundary curve and degree of continuity with adjacent surfaces after scaling. Therefore, type three control points of the new surface $\tilde{S}(u,v)$ should be the same as type three control points of the scaled version of the given trimmed NURBS surface $T_s \circ S(u,v)$. In the following, we will show how type one and type three control points for the new surface are arranged. The construction of type two

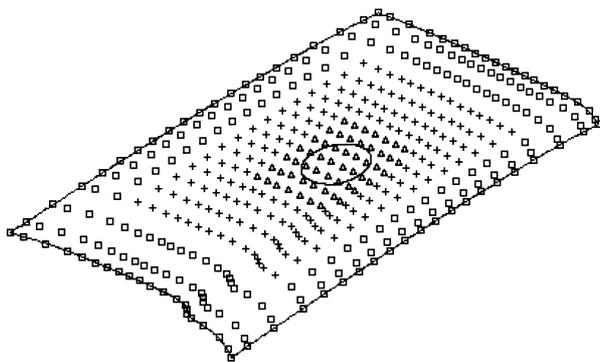


Fig. 2. (Δ), (\times) and (\square) denote control points of the first, second and third type, respectively.

control points for the new surface is shown in Section 7. We will use $\hat{Q}_{i,j}$ and $\bar{Q}_{i,j}$ to represent the control points of $T_s \circ S(u, v)$ and $\bar{S}(u, v)$, respectively.

5.1. Type one control points

Each control envelop of the given trimmed NURBS surface $S(u, v)$ has to be transformed to an appropriate location so that a corresponding trimming curve can be defined for the new surface $\bar{S}(u, v)$. This is the most critical step since it has a decisive influence on the curvature distribution of the new surface $\bar{S}(u, v)$ in the vicinity of the trimming curves. We need to define two terms first.

Let $P_{i,j}, j = 1, \dots, n_i$, be the vertices of the trimming curve $S \circ C_i(t)$. The centroid of the trimming curve $S \circ C_i(t)$ is defined as follows.

$$P_i = \frac{1}{n_i} \sum_{j=1}^{n_i} P_{i,j}$$

The direction vector N_i of the trimming curve $S \circ C_i(t)$ is defined by

$$N_i = \frac{1}{n_i} \sum_{j=1}^{n_i} N_{i,j} \quad (3)$$

where $N_{i,j}$ are normal vectors of the surface $S(u, v)$ at vertices $P_{i,j}$.

Three steps, one rotation and two translations, will be used to transform the control envelop of the trimming curve $S \circ C_i(t)$ to its new location. First, the control envelop of $S \circ C_i(t)$ is rotated, with P_i as pivot, so that its direction vector would be in the same direction as the direction vector of the scaled trimming curve $T_s \circ S \circ C_i(t)$ where T_s is defined in Eq. (2). It is easy to see that the direction vector of the scaled trimming curve is $\bar{T}_s N_i$ where

$$\bar{T}_s = \begin{bmatrix} S_y S_z & 0 & 0 \\ 0 & S_x S_z & 0 \\ 0 & 0 & S_x S_y \end{bmatrix} \quad (4)$$

with S_x, S_y and S_z being the scaling factors for the x, y and z directions. Second, the rotated control envelop of $S \circ C_i(t)$ is translated from its centroid P_i to $T_s P_i$, the centroid of the scaled trimming curve $T_s \circ S \circ C_i(t)$. These two steps align an original trimming curve with the corresponding trimming curve on the scaled surface in both centroid and orientation.

The third step is to move the rotated and translated control envelop along the direction vector of the scaled trimming curve $T_s \circ S \circ C_i(t)$ to a new location that is most appropriate for the new surface $\bar{S}(u, v)$. If we use $\bar{Q}_{ik,il}$ to represent the control points in the rotated control envelop of the trimming curve $S \circ C_i(t)$, and D_i to represent the displacement vector $T_s P_i - P_i$ of the second step, then the corresponding control points $\bar{Q}_{ik,il}$ for the new surface are defined by

$$\bar{Q}_{ik,il} = \bar{Q}_{ik,il} + D_i + d_i(\bar{T}_s N_i) \quad (5)$$

where d_i is a constant to be determined. In Ref. [3], a similar quantity in relocating a trimming curve is determined individually for each trimming curve using a least squares method. While this is possible, it may not provide a solution as good as the one produced through a global optimization process. In this paper $d_i, i = 1, 2, \dots, r$, will be determined with the type two control points in an optimization process to be performed in Section 7.

It is possible to write Eq. (5) as

$$\bar{Q}_{ik,il} = \bar{Q}_{ik,il} + \bar{D}_i \quad (6)$$

with \bar{D}_i being a displacement vector to be determined in an optimization process. This approach provides bigger flexibility for the relocation process of the control envelops. However, it works well only if the trimming curves are located on convex portions of the surface. Further discussion on this issue will be given in Section 7.

5.2. Type three control points

Type three control points of $\bar{S}(u, v)$ for a C^0 -continuity boundary constraint are arranged as follows

$$\bar{Q}_{0,j} = \hat{Q}_{0,j} \quad \bar{Q}_{m,j} = \hat{Q}_{m,j} \quad j = 0, \dots, n \quad (7)$$

$$\bar{Q}_{i,0} = \hat{Q}_{i,0} \quad \bar{Q}_{i,n} = \hat{Q}_{i,n} \quad i = 1, \dots, m - 1 \quad (8)$$

where $\hat{Q}_{i,j}$ are control points of the scaled surface $T_s \circ S(u, v)$. If C^1 continuity is required, additional equations should be included as follows:

$$\bar{Q}_{1,j} = \hat{Q}_{1,j} \quad \bar{Q}_{m-1,j} = \hat{Q}_{m-1,j} \quad j = 1, \dots, n - 1 \quad (9)$$

$$\bar{Q}_{i,1} = \hat{Q}_{i,1} \quad \bar{Q}_{i,n-1} = \hat{Q}_{i,n-1} \quad i = 2, \dots, m - 2 \quad (10)$$

The subdivision process performed in Section 4 guarantees that, in each direction, there will be at least three type two control points between type one and type three control points, or between type one and type one control points of different control envelops. Hence, there will be enough type two control points for us to perform the stretching process.

6. Setting up shape-preserving objective function

The stretching process requires the construction of a *shape-preserving objective function*. This function is used to determine the type two control points of the new surface $\bar{S}(u, v)$ in an optimization process. The new surface must reflect the shape and curvature distribution of the scaled surface. Hence, the objective function should be constructed based on the difference of these two surfaces. In our problem, the displacement function is

$$V(u, v) = (\bar{S} - T_s \circ S)(u, v) \quad (11)$$

where $\bar{S}(u, v)$ represents the new surface.

Several approximated energy functions have been used as the objective functions in geometric deformation

[6,19,23,26]. The goal is to minimize the energy of the displacement function so as to minimize the shape change of the deformed surface. We will use a physics-based approach similar to the one followed by Celniker and Welch [7] and Welch and Witkin [26].

The manipulation of a surface is like the manipulation of a thin plate. In our case, we only need to consider a *free plate* [21] since external forces such as *moment*, *edge force* and *gravity* either do not exist or are not required. On the other hand, since *stretching* is also involved in the manipulation process, the potential energy of the free plate considered in our case can be expressed as follows:

$$E(V) = \alpha E_{\text{bending}} + \beta E_{\text{stretching}} + \gamma E_{\text{spring}} \quad (12)$$

where E_{bending} , $E_{\text{stretching}}$ and E_{spring} are the *bending strain energy*, *stretching strain energy* and *spring potential energy* of V , respectively, and α , β and γ are weights to be determined. The *strain energy* of a thin plate bending process is defined as follows [17,25]:

$$E_{\text{bending}} = \frac{1}{2} \iint_D \kappa [(V_{uu} + V_{vv})^2 - 2(1 - \sigma)(V_{uu}V_{vv} - V_{uv}^2)] du dv \quad (13)$$

where σ is the Poisson constant and κ is a constant depending on the thickness and material property parameters of the plate. Here σ is set to zero and κ is set to one.

The *strain energy* of the thin plate stretching process, by ignoring the influence of the shearing strain, is [17,25]

$$E_{\text{stretching}} = \frac{1}{2} \iint_D [(2G + \lambda)(V_u^2 + V_v^2) + 2\lambda(V_uV_v)] du dv \quad (14)$$

where G and λ are constants depending on the material property parameters of the plate. In our case, λ is set to one and G is set to zero.

As far as the spring effect is concerned, we introduce springs between the scaled surface and the stretched surface to pull the stretched surface toward the scaled surface. Based on the spring energy definition, the potential spring energy is

$$E_{\text{spring}} = \frac{1}{2} \iint_D K \cdot V(u, v)^2 du dv \quad (15)$$

where K is the stiffness of the spring. Its value is set to one here.

The values of the weights α , β and γ in Eq. (12) are set as follows:

$$\alpha = \frac{E_\alpha}{E_{\alpha+\beta+\gamma}} \quad \beta = \frac{E_\beta}{E_{\alpha+\beta+\gamma}} \quad \gamma = \frac{E_\gamma}{E_{\alpha+\beta+\gamma}} \quad (16)$$

where

$$E_\alpha = \iint_D [W_{uu}(u, v)^2 + 2W_{uv}(u, v)^2 + W_{vv}(u, v)^2] du dv \quad (17)$$

$$E_\beta = \iint_D [W_u(u, v)^2 + 2W_u(u, v)W_v(u, v) + W_v(u, v)^2] du dv \quad (18)$$

$$E_\gamma = \iint_D W(u, v)^2 du dv \quad (19)$$

$$E_{\alpha+\beta+\gamma} = E_\alpha + E_\beta + E_\gamma \quad (20)$$

and $W(u, v)$ denotes $T_s \circ S(u, v)$. This follows from the observation that a term in expression (12) with a bigger energy should carry a bigger weight in the minimization process. Note that in an approximation process with fixed degree of freedom, the error is proportional to the complexity of the function. $V(u, v)$ is the approximation error of $\tilde{S}(u, v)$ to $T_s \circ S(u, v)$. Hence, if the bending energy of $W(u, v)$ is bigger than the stretching energy and the spring energy (i.e. $E_\alpha > E_\beta$ and $E_\alpha > E_\gamma$), we would have $E_{\text{bending}}(G_\alpha) > E_{\text{stretching}}(G_\beta)$ and $E_{\text{bending}}(G_\alpha) > E_{\text{spring}}(G_\gamma)$, where G_α , G_β and G_γ are the solution vectors of Eqs. (13)–(15), respectively.

On the other hand, let G be the solution vector of Eq. (12), i.e.

$$E(G) = \alpha E_{\text{bending}}(G) + \beta E_{\text{stretching}}(G) + \gamma E_{\text{spring}}(G) \quad (21)$$

The bending energy $E_{\text{bending}}(G)$ in the above equation is inversely proportional to α . Thus, to find a minimum point G that is close to that of $E_{\text{bending}}(G_\alpha)$, one needs to use an α that is relatively large compared with β and γ (note that $G \rightarrow G_\beta$ if $\beta \rightarrow 1$ and $G \rightarrow G_\gamma$ if $\gamma \rightarrow 1$). Here, without the loss of generality, we assume that the weights are normalized). Hence, a bigger bending energy of $W(u, v)$ should imply a bigger α value. Similarly, a bigger stretching energy or spring energy of $W(u, v)$ should imply a bigger β or γ value. Therefore, α , β and γ should be defined in a way that their values are proportional to E_α , E_β and E_γ , respectively.

A simpler approach is to set all the weights in expression (12) to one. The result of this approach sometimes is not as good as that of the above approach. This is because setting α , β and γ to one in Eq. (12) means minimizing the average energy of $E_{\text{bending}}(G_\alpha)$, $E_{\text{stretching}}(G_\beta)$ and $E_{\text{spring}}(G_\gamma)$ while setting the values of α , β and γ based on Eq. (16) means minimizing the maximum energy of expressions (13)–(15).

7. Optimization: the constrained stretching process

For NURBS surfaces, minimization of (13), (14) and (15) leads to a quadratic equation with respect to the control points if a homogeneous representation is used. The homogeneous representations of $T_s \circ S(u, v)$ and $\tilde{S}(u, v)$ (see Section 2 for the definition of $S(u, v)$) are

$$\sum_{i=0}^m \sum_{j=0}^n (w_{i,j} \hat{Q}_{ij}, w_{i,j}) N_{i,p}(u) N_{j,q}(v)$$

and

$$\sum_{i=0}^m \sum_{j=0}^n (w_{i,j} \hat{Q}_{ij}, w_{i,j}) N_{i,p}(u) N_{j,q}(v)$$

respectively. For simplicity of notations, we shall use $T_s \circ S(u, v)$ and $\bar{S}(u, v)$ to represent their own homogeneous forms, i.e. $\hat{Q}_{i,j}$ and $\bar{Q}_{i,j}$ are homogeneous control points of the following forms:

$$\hat{Q}_{i,j} = (w_{i,j} \hat{Q}_{ij}, w_{i,j}) \quad \bar{Q}_{i,j} = (w_{i,j} \bar{Q}_{ij}, w_{i,j})$$

$T_s \circ S(u, v)$ and $\bar{S}(u, v)$ can be written as linear equations with respect to their control points as follows:

$$T_s \circ S(u, v) = \sum_{i=0}^{\Delta} \hat{Q}_k N_k(u, v) \quad (22)$$

$$\bar{S}(u, v) = \sum_{i=0}^{\Delta} \bar{Q}_k N_k(u, v) \quad (23)$$

where

$$\Delta \equiv (m+1) \times (n+1) - 1 \quad (24)$$

$$\hat{Q}_k = \hat{Q}_{i,j} \quad \bar{Q}_k = \bar{Q}_{i,j} \quad (25)$$

$$N_k(u, v) = N_{i,p}(u) N_{j,q}(v) \quad (26)$$

with

$$i = k - [k/(m+1)] \times (m+1) \quad j = [k/(m+1)]$$

By representing type one, type two and type three control points of $\bar{S}(u, v)$ by $\bar{Q}_{i_g}^1$, $\bar{Q}_{j_g}^2$ and $\bar{Q}_{i_g}^3$, respectively, $\bar{S}(u, v)$ can be expressed as

$$\bar{S}(u, v) = \sum_{g=0}^{\Delta_1} \bar{Q}_{i_g}^1 N_{i_g}(u, v) + \sum_{g=0}^{\Delta_2} \bar{Q}_{j_g}^2 N_{j_g}(u, v) + \sum_{g=0}^{\Delta_3} \bar{Q}_{i_g}^3 N_{i_g}(u, v) \quad (27)$$

where $\Delta_1 + \Delta_2 + \Delta_3 = \Delta - 2$. According to Eq. (5), and Eqs. (7) and (8) (or, Eqs. (9) and (10), depending on the boundary smoothness condition), the unknowns of the above equation are r scalars d_i , $i = 1, 2, \dots, r$, for the r control envelopes and $\Delta_2 + 1$ type two control points, $\bar{Q}_{j_g}^2$, $g = 0, 1, \dots, \Delta_2$.

By substituting Eqs. (22) and (27) into Eq. (11) and expressions (13)–(15), and then substituting expressions (13)–(15) into Eq. (12), one gets a quadratic expression with unknowns d_i , $i = 1, 2, \dots, r$, and $\bar{Q}_{j_g}^2$, $g = 0, 1, \dots, \Delta_2$:

$$G(\bar{Q}_{j_0}^2, \bar{Q}_{j_1}^2, \dots, \bar{Q}_{j_{\Delta_2}}^2, d_1, d_2, \dots, d_r) \quad (28)$$

The unknowns are determined by differentiating expression (28) with respect to d_i and $\bar{Q}_{j_g}^2$ to get the following

system of linear equations:

$$\frac{\partial G(\bar{Q}_{j_0}^2, \bar{Q}_{j_1}^2, \dots, \bar{Q}_{j_{\Delta_2}}^2, d_1, d_2, \dots, d_r)}{\partial d_i} = 0$$

$$\frac{\partial G(\bar{Q}_{j_0}^2, \bar{Q}_{j_1}^2, \dots, \bar{Q}_{j_{\Delta_2}}^2, d_1, d_2, \dots, d_r)}{\partial \bar{Q}_{j_g}^2} = 0 \quad (29)$$

where $i = 1, 2, \dots, r$ and $g = 0, 1, \dots, \Delta_2$, and then solving the system (29) for d_i and $\bar{Q}_{j_g}^2$.

Instead of using Eq. (5) for the type one control points in Eq. (27), one can use Eq. (6) for the same thing. This would lead to a quadratic expression with unknowns \bar{D}_i , $i = 1, 2, \dots, r$, and $\bar{Q}_{j_{\Delta_2}}^2$, $g = 0, 1, \dots, \Delta_2$ as follows.

$$G(\bar{Q}_{j_0}^2, \bar{Q}_{j_1}^2, \dots, \bar{Q}_{j_{\Delta_2}}^2, \bar{D}_1, \bar{D}_2, \dots, \bar{D}_r) \quad (30)$$

The unknowns are determined similarly, i.e. differentiating Eq. (30) with respect to \bar{D}_i and $\bar{Q}_{j_g}^2$ to get the following system of linear equations:

$$\frac{\partial G(\bar{Q}_{j_0}^2, \bar{Q}_{j_1}^2, \dots, \bar{Q}_{j_{\Delta_2}}^2, \bar{D}_1, \bar{D}_2, \dots, \bar{D}_r)}{\partial \bar{D}_i} = 0$$

$$\frac{\partial G(\bar{Q}_{j_0}^2, \bar{Q}_{j_1}^2, \dots, \bar{Q}_{j_{\Delta_2}}^2, \bar{D}_1, \bar{D}_2, \dots, \bar{D}_r)}{\partial \bar{Q}_{j_g}^2} = 0 \quad (31)$$

where $i = 1, 2, \dots, r$ and $g = 0, 1, \dots, \Delta_2$, and then solving the system (31) for \bar{D}_i and $\bar{Q}_{j_g}^2$.

These two approaches produce similar results if the trimming curves are located on convex portions of a surface. Otherwise, Eq. (29) generally produces better results than Eq. (31). The reason that Eq. (29) produces better results in scaling concave NURBS surfaces is that, with Eq. (29) the rotated control envelop of the trimming curve $S \circ C_i(t)$ can only move along the line $P_i + D_i + d_i \bar{T}_s N_i$ (here d_i is viewed as a parameter) to minimize the objective function (28). The optimization process will make the computed d_i small and, hence, the final position of the control points in the control envelop, $\bar{Q}_{i_g}^1$, will be close to that of $\bar{Q}_{i_g}^1 + D_i$. On the other hand, with Eq. (31) the rotated control envelop of the trimming curve $S \circ C_i(t)$ can move in any direction to minimize the objective function (30). In case that $S(u, v)$ is not a convex surface, the determined $\bar{Q}_{i_g}^1$ may not be close to $\bar{Q}_{i_g}^1 + D_i$ and, thus, the new surface $\bar{S}(u, v)$ may be visually very different from the original surface $S(u, v)$.

The advantage of Eq. (30) is that it will result in three smaller independent sets of linear equations with *X-components*, *Y-components* and *Z-components* as unknowns, respectively. The *X*-, *Y*- and *Z-components* include those of both $\bar{Q}_{i_g}^1$ and D_i . As the three sets of equations are independent of each other, the computation cost is lower and the systems are easier to solve. expression (28) will result in a set of linear equations with *X-components*, *Y-components*, *Z-components* and d_1, \dots, d_r as its unknowns. Such a system is more costly to solve since it involves all the unknowns in a large system.

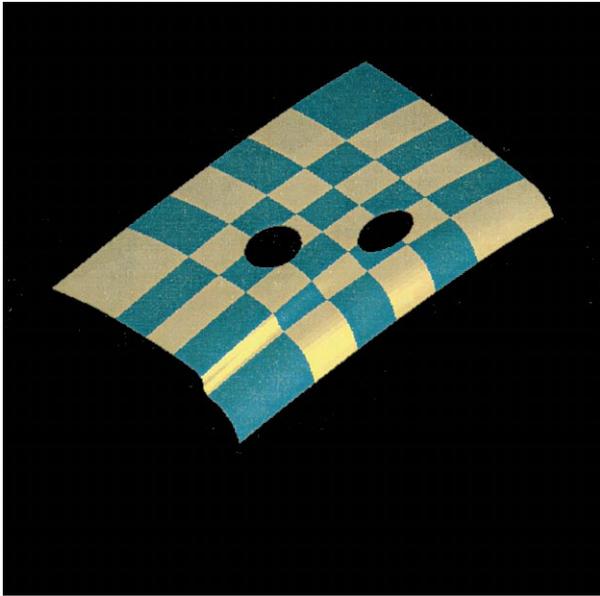


Fig. 3. Patches of the trimmed door panel.

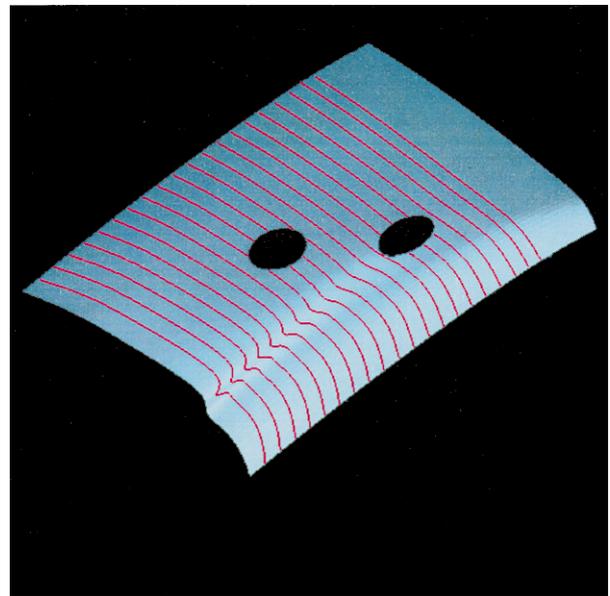


Fig. 5. Trimmed door panel after scaling.

8. Implementation

Test results on three data sets are presented here. These data sets include a trimmed door panel (Figs. 3–5), a trimmed trunk hood (Figs. 6–8), and a trimmed front hood (Figs. 9–11). The front hood is a degree 3×5 NURBS surface with 8 patches. The trunk hood and the door panel are bicubic NURBS surfaces with 36 patches each. These three surfaces are also B-spline surfaces because the weights in the NURBS representations of these surfaces are all equal to one.

Three images are shown for each case: the first one shows

the patches of the given trimmed surface; the second one shows the trimmed surface before the scaling process and the third one shows the result of the constrained scaling process. The surfaces in Figs. 5, 8, and 11 are produced using expression (28). The scaling factors for the three cases are: $S_x = 1.1$, $S_y = 1.2$, $S_z = 1.25$. The shaded trimmed surfaces before scaling and after scaling are displayed with a set of highlight lines [1,2,8]. Highlight lines are sensitive to the change of normal directions, hence, can be used to detect surface normal (curvature) irregularities. This sometimes is not possible with wire-frame drawings or shaded pictures [9,11].

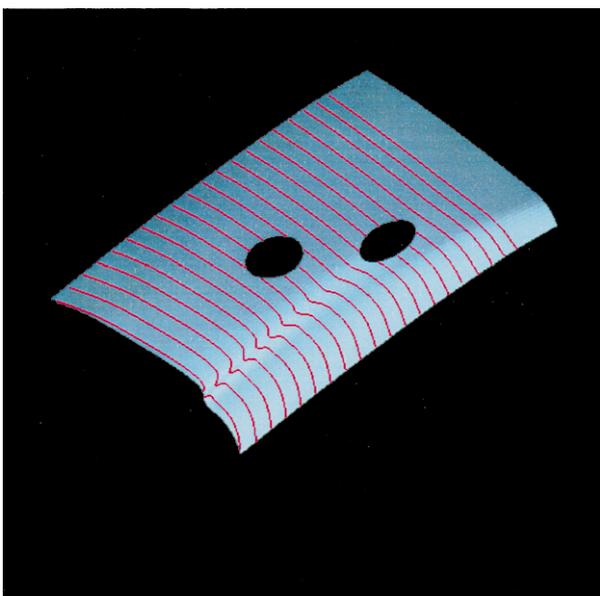


Fig. 4. Trimmed door panel before scaling.

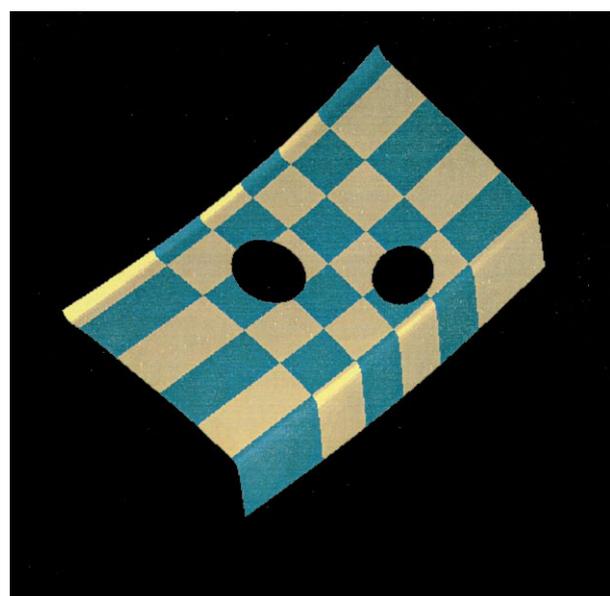


Fig. 6. Patches of the trimmed trunk hood.

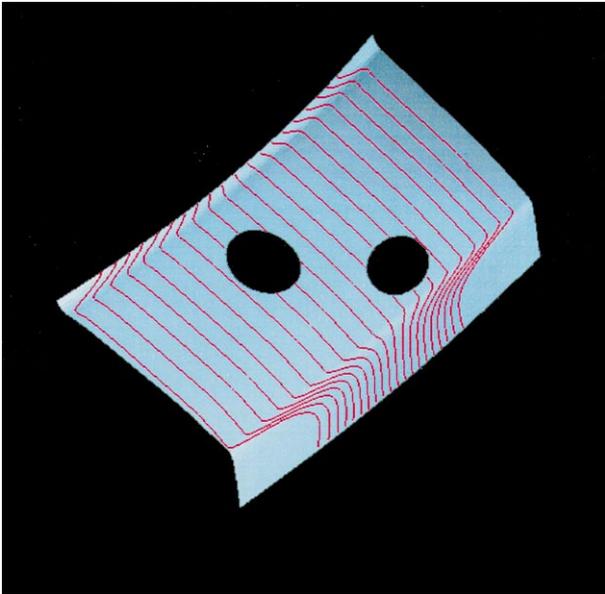


Fig. 7. Trimmed trunk hood before scaling.

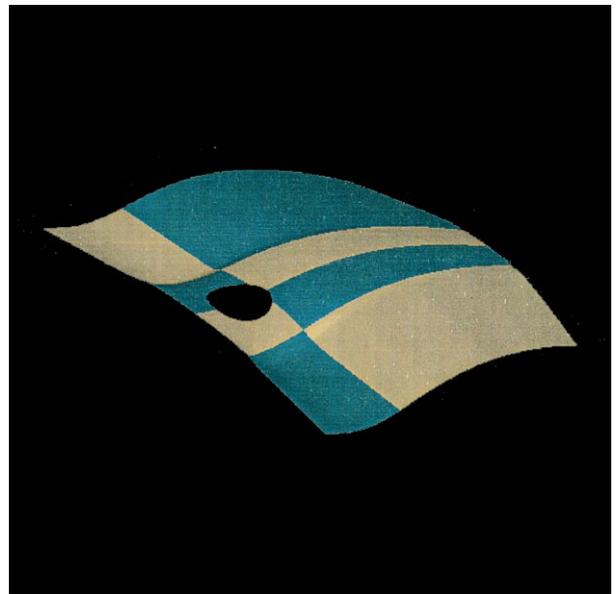


Fig. 9. Patches of the trimmed front hood.

The examples show that, in each case, the result of the constrained scaling has the same features as the original surface, while having the same shape as the scaled surface. The curvature distribution of the result is also the same as the scaled surface. This can be verified by comparing the highlight lines on the surface before and after the constrained scaling process.

Comparison of the new approach and the previous approach [3] has also been performed. In addition to the fact that the new approach maintains both the shape and size of the features after the constrained scaling process, it also tolerates larger scaling factors. While the attach-and-deform

based approach generates distorted results for scaling factors as low as 1.3, the new approach can tolerate scaling factors as large as 2 in some cases. Fig. 12 shows the result of constrained scaling using the attach-and-deform based approach with scaling factors $S_x = 1.3$, $S_y = 1.3$ and $S_z = 1.3$. Fig. 13 shows the result of constrained scaling using the fix-and-stretch based approach with scaling factors $S_x = 2.0$, $S_y = 2.0$ and $S_z = 2.0$ (result is scaled down to fit into the fixed sized viewport). The distorted highlight lines of Fig. 12 shows that the curvature distribution of the resulting surface is not the same as the original surface (Fig. 4). Figs. 12 and 13 also show that the displacement vector \bar{D}_i

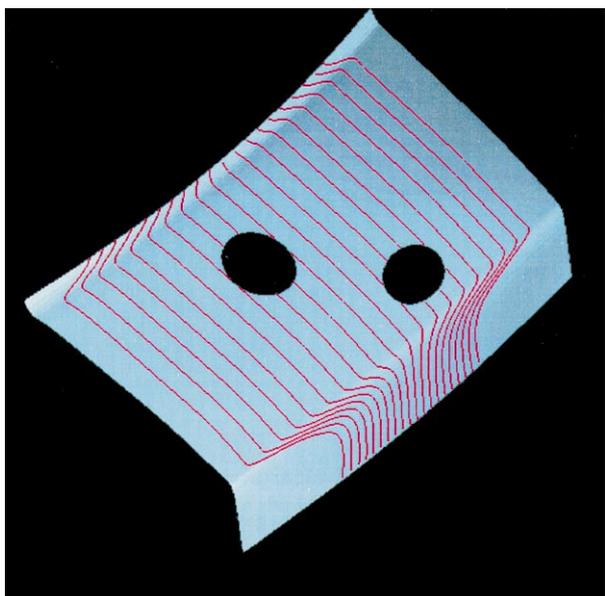


Fig. 8. Trimmed trunk hood after scaling.

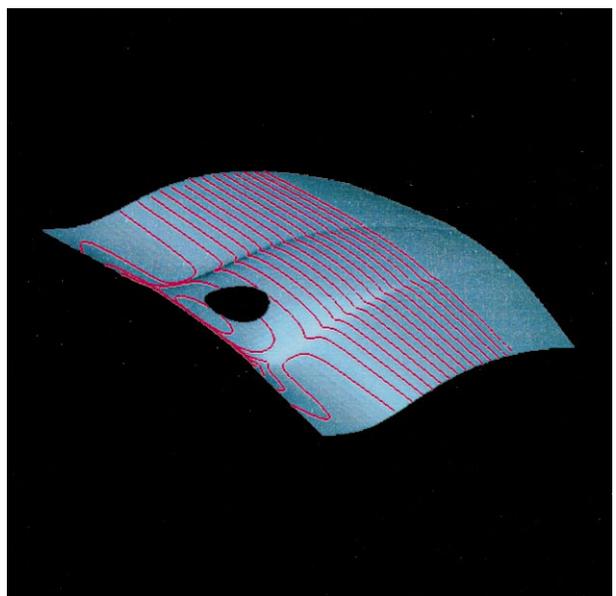


Fig. 10. Trimmed front hood before scaling.

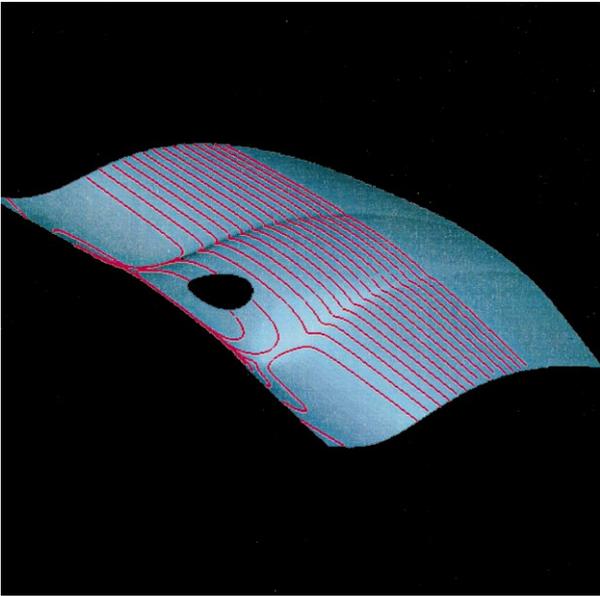


Fig. 11. Trimmed front hood after scaling.

has much to do with the curvature distribution of the new surface near the region of the trimming curve $S \circ C_i$.

9. Conclusion

A new, fix-and-stretch based approach for constrained surface scaling of trimmed NURBS surfaces is presented. The new surface is formed by fixing selected regions of the given trimmed NURBS surface that contain the trimming curves and stretching the remaining part of the surface to

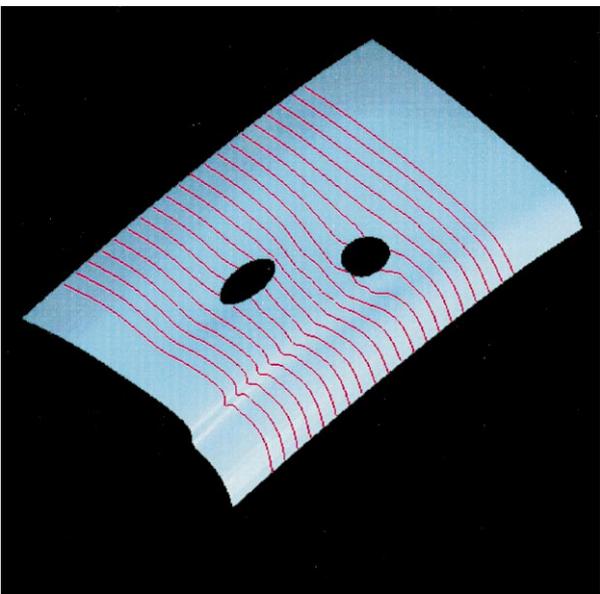


Fig. 12. Constrained scaling using the attach-and-deform based approach with scaling factors $S_x = 1.3$, $S_y = 1.3$ and $S_z = 1.3$.

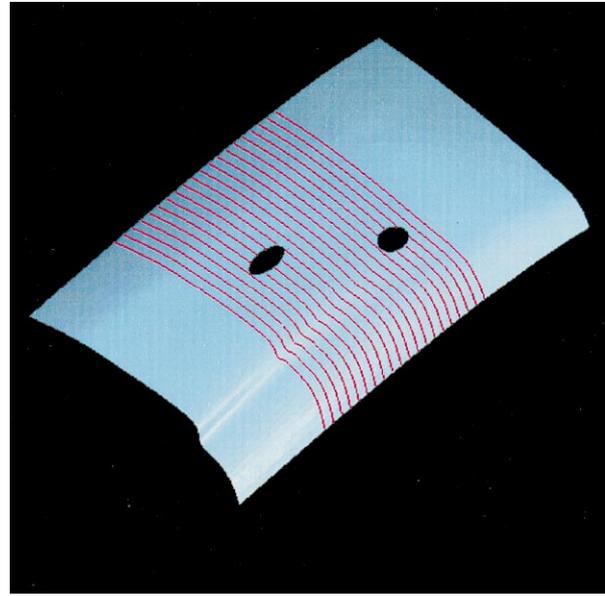


Fig. 13. Constrained scaling using the fix-and-stretch based approach with scaling factors $S_x = 2.0$, $S_y = 2.0$ and $S_z = 2.0$.

reach certain boundary conditions. The stretching process is performed using an optimization process to ensure that the resulting surface reflects the shape and curvature distribution of the scaled version of the given surface. The resulting surface maintains a NURBS representation and, hence, is compatible with most of the current data-exchange standards.

Advantages of the new approach over the previous, attach-and-deform based approach include: (1) it tolerates bigger scaling factors (up to 2 in some cases) than the previous, deformation based approach; and (2) it guarantees that the features remain exactly the same after scaling. However, the new approach requires more memory than the previous approach in some cases.

The features considered in the new and previous approaches are within a single NURBS surface. A future work is to consider the case when a feature intersects the boundary of the given surface. Another future research direction is to scale different components of an object with different scaling factors while maintaining the overall smoothness of the object and keeping certain features fixed.

Acknowledgements

The financial support of Ford Motor Company through a URP grant and Honda Motor Company through two HIG grants, and the consulting support of Drs Paul Stewart and Yifan Chen of the Ford Research Lab and Dr Shane Chang of the Honda R and D North America are deeply appreciated. We also thank the reviewers for several comments and suggestions which helped improve the quality of the paper.

References

- [1] Beier K-P, Chen Y. The highlight-line algorithm for real-time surface-quality assessment. *Computer-Aided Design* 1994;26:268–78.
- [2] Beier K-P, Chen Y. The highlight band, a simplified reflection model for interactive smoothness evaluation. In: Spapidis N, editor. *Designing fair curves and surfaces*, Philadelphia, PA: SIAM, 1994. p. 213–30.
- [3] Zhang P, Zhang C, Cheng F. Constrained shape scaling of trimmed NURBS surfaces. In: *Proceedings of 1999 ASME Design Theory and Methodology Conference*. Las Vegas, Nevada, 1999.
- [4] Borrel P, Rappoport A. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics* 1994;13(2):137–55.
- [5] Cavendish JC. Integrating feature-based surface design with free-form deformation. *Computer-Aided Design* 1995;27(9):703–11.
- [6] Celniker G, Gossard D. Deformable curve and surface finite elements for free form shape design. *Computer Graphics* 1991;25(4):257–66.
- [7] Celniker G, Welch W. Linear constraints for deformable B-spline surfaces. In: *Proceedings of the Symposium on Interactive 3D*, 1992. p. 165–70.
- [8] Chen Y. Highlight lines for surface quality control and manipulation. PhD dissertation, Department of Naval Architecture and Marine Engineering, The University of Michigan, 1993.
- [9] Chen Y, Beier K-P, Papageorgiou D. Direct highlight line modification on nurbs surface. *Computer Aided Geometric Design* 1997;14:583–601.
- [10] Coquillart S. Extended free-form deformation: a sculpting tool for 3D geometric modeling. *Computer Graphics* 1990;24(4):187–93.
- [11] Zhang C, Cheng F. Removing local irregularities of NURBS surfaces by modifying highlight lines. *Computer-Aided Design* 1998;30(12):923–30.
- [12] Feng J, Ma L, Peng Q. A new free-form deformation through the control of fratic surfaces. *Computer and Graphics* 1996;20(4):531–9.
- [13] Griessmair J, Parray R. Deformation of solids with trivariate B-spline. In: *Proceedings of Euro-graphics'89*, 1989. p. 137–48.
- [14] Hsu WH, Hughes JF, Kaufman H. Direct manipulation of free-form deformations. *Computer Graphics* 1992;26(2):177–84.
- [15] Kalra P, et al. Simulation of facial muscle actions based on rational free-form deformations. *Computer Graphics Forum* 1992;2(3):C59–69.
- [16] Lamousin HJ, Waggenspack WN. NURBS-based free-form deformations. *IEEE Transactions on Computer Graphics and Applications* 1994;14(6):59–65.
- [17] Mansfield EH. *The bending and stretching of plates*. New York: Macmillan, 1964.
- [18] Platt J, Barr A. Constraints method for flexible models. *Computer Graphics* 1988;22(4):279–88.
- [19] Sarraga RF. Recent methods for surface shape optimization. *Computer Aided Geometric Design* 1998;15:417–36.
- [20] Sederberg TW, Parry SR. Free-form deformation of solid geometric models. *Computer Graphics* 1986;20(4):151–60.
- [21] Terzopoulos D. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image processing* 1983;24:52–96.
- [22] Terzopoulos D, Fleischer K. Deformable models. *Visual Computing* 1988;4(6):306–31.
- [23] Terzopoulos D, Qin H. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics* 1994;13(2):103–36.
- [24] Terzopoulos D, Platt J, Barr A, Fleischer K. Elastically deformable models. *Computer Graphics* 1987;21(4):205–14.
- [25] Ugural AC, Ferster SK. *Advanced strength and applied elasticity*. New York: America Elsevier Publishing, 1975.
- [26] Welch W, Witkin A. Variational surface modeling. *Computer Graphics* 1992;26(2):157–66.



Caiming Zhang is a post doctoral fellow in the Graphics and Geometric Modeling Lab at the University of Kentucky. He received a BS and an ME in computer science from the Shandong University in 1982 and 1984, respectively, and a Dr. Engng degree from the Tokyo Institute of Technology, Japan, in 1994. Dr Zhang has also held visiting position at the Tokyo Institute of Technology, Japan. His research interests included computer aided geometric design and modeling, computer graphics and image processing. His permanent address is: Department of Computer Science, Shandong University, Jinan, China.



Pifu Zhang is a post doctoral fellow in the Graphics and Geometric Modeling Lab at the university of Kentucky. He received his B. Eng. and ME in mechanical engineering from Hunan Agricultural University and Beijing Agricultural Engineering University in 1983 and 1987, respectively, and a Dr. Engng degree from Hunan University, China, in 1998. His research interests include computer geometric design and modeling, computer graphics, and visualization, and mesh generation.



Fuhua (Frank) Cheng is Professor of Computer Science and Supervisor of the Graphics and Geometric Modeling Lab at the University of Kentucky where he joined the faculty in 1986. He is also an Adjunct Professor of Applied Mathematics at the Shandong University of Technology, Jinan, China. He received a BS and an MS in mathematics from the National Tsing Hua University in Taiwan in 1973 and 1975, respectively, an MS in mathematics, an MSD in computer science, and a PhD in applied mathematics and computer science from the Ohio State University, in 1978, 1980, and 1982, respectively. Dr Cheng has held visiting positions at the University of Tokyo and the University of Aizu, Japan. His research interests include computer aided geometric modeling, computer graphics, and parallel computing in geometric modeling and computer graphics.