Constrained Scaling of Trimmed NURBS Surfaces

Pifu Zhang, Caiming Zhang and Fuhua (Frank) Cheng
Graphics & Geometric Modeling Lab, Department of Computer Science
University of Kentucky, Lexington, Kentucky 40506-0046

Abstract. A method to scale and deform a trimmed NURBS surface while holding the shape and size of specific features (trimming curves) unchanged is presented. The new surface is formed by scaling the given surface according to the scaling requirement first, and then attaching the (original) features to the scaled NURBS surface at appropriate locations. The attaching process requires several geometric operations and constrained free-form surface deformation. The resulting surface has the same features as the original surface and same boundary curves as the scaled surface while reflecting the shape and curvature distribution of the scaled surface. This is achieved by minimizing a shape-preserving objective function which covers all the factors in the deformation process such as bending, stretching and spring effects. The resulting surface maintains a NURBS representation and, hence, is compatible with most of the current data-exchange standards. Test results on several car parts with trimming curves are included. The quality of the resulting surfaces is examined using the highlight line model.

Keywords: constrained scaling, constrained deformation, trimming curves, NURBS surfaces, strain energy

1 Introduction

A surface design problem of especial 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 expansive redesign process.

Using scaling as an altering technique is common in design. The problem of *constrained scaling* (i.e., scaling a model with some features fixed), however, has not been seriously addressed in the literature yet. Deformation as an altering technique, on the other hand, is not as commonly used in design. 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 trivariate parametric volume. Deformation is performed by manipulating the control points of the trivariate volumes [4][5][10][12][13][14][15][16][19].

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 the physical properties of mass distribution, tension, rigidity, damping and the action of applied forces. The resulting surface is determined by the equilibrium state of the dynamic model [6][18][22][23][24]. This method is mainly used in computer 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 [27]. 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.

In this paper, we will present a method that is capable of holding certain features (trimming curves) of a surface unchanged while scaling it. The new surface is formed by scaling the given surface according to the scaling requirement first; and then attaching the original features to the scaled NURBS surface at appropriate locations. The attaching process requires several geometric operations and constrained free-form surface deformation. The deformation process is similar to Celniker and Welch's approach [7] but in a different setting. The resulting surface has the same features as the original surface and same boundary curves as the scaled surface while reflecting the shape and curvature distribution of the scaled surface. The resulting surface also maintains a NURBS representation.

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 4-8. Implementation issues and test results of the proposed method are shown in Section 9. Concluding remarks are given in Section 10.

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 $\bar{S}(u,v)$ 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 u direction and degree q in v direction

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

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

where $Q_{i,j}$ are 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 weight functions. $N_{i,p}(u)$ and $N_{j,q}(v)$ are defined with respect to the knot vectors $\tau = \{\tau_0, \tau_1, \ldots, \tau_{m+p+1}\}$, and $\sigma = \{\sigma_0, \sigma_1, \ldots, \sigma_{n+q+1}\}$, respectively, with $\tau_0 = \ldots = \tau_p = \sigma_0 = \ldots = \sigma_q = 0$ and $\tau_{m+1} = \ldots = \tau_{m+p+1} = \sigma_{n+1} = \ldots = \sigma_{n+q+1} = 1$. The features to be held unchanged are closed trimming curves $S \circ C_i(t)$, $i = 1, 2, \ldots, r$, 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 x, y and z directions are S_x, S_y and S_z, respectively, then the new surface $\bar{S}(u,v)$ is supposed to be equal to $T_s \circ S$, the scaled NURBS surface, where T_s is a scaling matrix defined as follows:

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

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

If the new surface is not required to be precisely the same as $T_s \circ S(u,v)$, but only close enough to and reflecting the shape and curvature distribution of the scaled surface, then an approach based on the concept of *attach-and-deform* can be used to construct the new surface. The main idea of this approach is to attach the original trimming curves to the scaled NURBS surface at appropriate locations. The attaching process requires several geometric operations, such as translation and rotation, and a deformation of the scaled surface. The deformation is performed with constraints to ensure that the trimming curves are attached to the surface completely and the boundary continuity condition with adjacent surfaces of the scaled surface are unchanged. The deformed surface is again a NURBS surface. The deformation process will also ensure that the resulting surface reflects the shape and curvature distribution of the scaled surface.

An algorithm is presented 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. Scaling $S(u,v)$ with T_s
3. Relocating trimming curves $S \circ C_i(t)$
4. Setting up *shape-preserving objective function*
5. Setting up constraints
6. Performing *constrained surface deformation.*
7. Rendering

The third step is the most critical step because it also determines the outcome of the deformation process. Details of the above steps, except Step 2, are given in the subsequent sections.

4 Subdivision of $S(u, v)$

This step recursively subdivides the surface $S(u, v)$ until the following three conditions are satisfied:

1. Each subpatch intersects at most one trimming curve.
2. The trimming curves do not intersect any of the boundary subpatches (subpatches adjacent to the boundary of $S(u, v)$).
3. The number of trimming curve vertices contained in each subpatch is at most $(p + 1)(q + 1)$.

The first two conditions are to provide enough flexibility for setting up the *trimming curve constraint* and the *boundary constraint* (to be discussed in detail in Section 7). The third condition is to avoid over-determined systems in the deformation process.

The first two conditions are satisfied if the dimension of each subpatch is smaller than or equal to one half of the minimimum of Ω_1 and Ω_2: Ω_1 is the smallest distance between the vertices of the trimming curves $S \circ C_i(t)$ and the boundary of the NURBS surface $S(u, v)$, Ω_2 is the smallest distance between vertices of different trimming curves $S \circ C_i(t)$. The third condition has to be tested after each level of recursive subdivision once the first two conditions are satisfied.

The computation of Ω_2 is straightforward while Ω_1 can be computed using the Newton-Raphson method on the dirivative of a distance function.

It is possible to perform subdivision on boundary spans and spans that intersect the trimming curves only. This would reduce the subdivision time to certain extent. However, the highlight line model of the deformation 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 loss of generality, we shall use the same notations for the control points and parameter knots even though both of them might have been changed after the subdivision process.

5 Relocating Trimming Curves

This step is to move each original trimming curves $S \circ C_i(t)$ to an appropriate location that is not only as close to the scaled surface $T_s \circ S(u, v)$ as possible but also with an appropriate orientation. The first requirement is to ensure that only a small deformation is required to include the trimming curve as a feature. The second requirement is to ensure that deformation would not cause much distortion of the curvature of the scaled surface. The closeness will be measured in Euclidean distance.

We use three steps, two translations and one rotation, to determine the new location for each 3D trimming curve $S \circ C_i(t)$. First, the trimming curve $S \circ C_i(t)$ is translated from P_i, its centroid, to $T_s P_i$, the centroid of the scaled trimming curve $T_s \circ S \circ C_i(t)$. The mean normal vector N_i of the trimming curve $S \circ C_i(t)$,

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

where $N_{i,j}$ are normal vectors of the trimming curve $S \circ C_i(t)$ at its vertices $P_{i,j}$, is then rotated about the vector $U_i = N_i \times \bar{T}_s N_i$ until it is in the same direction as $\bar{T}_s N_i$, the mean normal vector of the scaled trimming curve where

$$\bar{T}_s = \begin{bmatrix} S_y S_z & 0 & 0 \\ 0 & S_x S_z & 0 \\ 0 & 0 & S_x S_y \end{bmatrix}. \tag{4}$$

These two steps align an original trimming curve with the corresponding trimming curve on the scaled surface in both centroid and direction. We will use the same notations for the translated and rotated original trimming curves, including their vertices.

After the translation and rotation, if one projects the vertices $P_{i,j}$ of the trimming curve $S \circ C_i(t)$ onto the scaled surface $T_s \circ S(u, v)$ in the direction of $\bar{T}_s N_i$, one gets a set of points $\bar{P}_{i,j}$ on $T_s \circ S(u, v)$. The *distance* between the translated and rotated trimming curve and the scaled surface $T_s \circ S(u, v)$ is defined as the sum of the distances between the vertices of the translated and rotated trimming curve to their projections $\bar{P}_{i,j}$ on the scaled surface.

The third step is to move the translated and rotated trimming curve $S \circ C_i(t)$ along the vector $\bar{T}_s N_i$ to be as close to the scaled surface as possible, i.e, the distance defined above is a minimum. This problem is equivalent to finding a plane whose distance to a set of finite points is a minimum. Such a problem can be solved using the least squares method.

The problem of finding $\bar{P}_{i,j}$ for the translated and rotated trimming curve is equivalent to finding the first intersection point of an arrow and a parametric surface. This problem can be solved using the adaptive subdivision method.

We use R_i to represent the relocating transformation for the trimming curve $S \circ C_i(t)$ and $\bar{C}_i$ to represent the trimming curve on the scaled surface $T_s \circ S(u, v)$ defined by $\bar{P}_{i,j}$.

6 Setting Up Shape-Preserving Objective Function

The deformation process requires the construction of a *shape-preserving objective function*. This function is used to determine the shape of the deformed surface in an optimization process. The deformed 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

$$V(u,v) = (\bar{S} - T_s \circ S)(u,v). \tag{5}$$

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

Several approximated energy functions have been used as the objective fuctions in geometric deformation [6][20][24][27]. 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 in our work.

The deformation of a surface is like the deformation of a thin plate. According to Courant [11], the *energy* of a deformed thin plate is composed of five parts: *bending strain energy*, *spring potential energy*, *gravity energy*, *moment energy* and *edge force energy*, as follows:

$$E(V) = E_{bending} - E_{gravity} + E_{spring} - E_{moment} - E_{edgeforce} \tag{6}$$

In a typical geometric modeling problem, the external forces such as *moment*, *edge force* and *gravity* are set to zero. This leads to a *free plate* [21]. So the potential energy of a free plate can be expressed as follows:

$$E(V) = E_{bending} + E_{spring} \tag{7}$$

Here we keep the spring potential energy because of the requirment that the new surface should have the smallest change in shape in order to keep characteristics of the original surface such as smoothness and curvature. On the other hand, deformation of a thin plate would also involve *stretching* if some features are required to be fixed, such as in sheet metal stamping. So the *stretching strain energy* should be included in eq. (7) as well, as follows:

$$E(V) = \alpha E_{bending} + \beta E_{stretching} + \gamma E_{spring}. \tag{8}$$

where α, β and γ are weights. According to the theory of mechanics [17][25], the *strain energy* for a thin plate bending process is defined as follows:

$$E_{benbing} = \int\int_D \kappa[\frac{1}{2}(V_{uu}+V_{vv})^2 - (1-\sigma)(V_{uu}V_{vv} - V_{uv}^2)]dudv \tag{9}$$

where σ is the Poisson constant (set to 0 here) and κ is a constant depending on the thickness and material property parameters of the plate.

The *strain energy* for the thin plate stretching process, by ignoring the influence of the shearing strain, is

$$E_{stretching} = \frac{1}{2}\int\int_D [(2G+\lambda)(V_u^2 + V_v^2) + 2\lambda(V_u V_v)]dudv \tag{10}$$

where G and λ are constants depending on the material property parameters of the plate.

By introducing springs at the knots $(\tau_i,\ \sigma_j)$ to pull the new surface toward the scaled surface, we can define the corresponding potential spring energy as follows:

$$E_{spring} = \frac{1}{2}\int\int_D K(\tau_i,\ \sigma_j)[V(\tau_i,\ \sigma_j)]^2 \tag{11}$$

where $K(\tau_i,\ \sigma_j)$ is the stiffness of the spring at $(\tau_i,\ \sigma_j)$.

For NURBS surfaces, eqs. (9), (10) and (11) lead to a quadratic equation with respect to the control points if homogeneous representation is used. The homogeneous representations of $T_s \circ S(u,v)$ and $\bar{S}(u,v)$ (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

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

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:

$$T_s \circ S(u,v) = \sum_{i=0}^{\Delta}\hat{Q}_k N_k(u,v); \tag{12}$$

$$\bar{S}(u,v) = \sum_{i=0}^{\Delta}\bar{Q}_k N_k(u,v), \tag{13}$$

where

$$\Delta \equiv (m+1)\times(n+1) - 1, \tag{14}$$

$$\hat{Q}_k = \hat{Q}_{i,j};\qquad \bar{Q}_k = \bar{Q}_{i,j}, \tag{15}$$

$$N_k(u,v) = N_{i,p}(u)N_{j,q}(v), \tag{16}$$

with

$$i = k - \lfloor k/(m+1) \rfloor \times (m+1),$$
$$j = \lfloor k/(m+1) \rfloor.$$

By substituting eqs. (12) and (13) into eq. (5) and eqs. (9), (10) and (11) and then subtituting eqs. (9), (10) and (11) into eq. (8), one gets the following expression through simple algebra:

$$E(\bar{Q} - \hat{Q}) = [\bar{Q} - \hat{Q}]^\top A[\bar{Q} - \hat{Q}] \tag{17}$$

where

$$\hat{Q} = [\hat{Q}_0, \hat{Q}_1, ..., \hat{Q}_\Delta] \tag{18}$$

$$\bar{Q} = [\bar{Q}_0, \bar{Q}_1, ..., \bar{Q}_\Delta] \tag{19}$$

and A is a $(\Delta+1) \times (\Delta+1)$ matrix whose entries are defined as follows:

$$\begin{aligned} a_{i,j} = \int_0^1 \int_0^1 [\alpha_1 N_i^{uu}(u,v) N_j^{uu}(u,v) + \alpha_2 N_i^{uv}(u,v) N_j^{uv}(u,v) \\ +\alpha_3 N_i^{uu}(u,v) N_j^{vv}(u,v) + \alpha_4 N_i^{vv}(u,v) N_j^{vv}(u,v) \\ +\beta_1 N_i^{u}(u,v) N_j^{u}(u,v) + \beta_2 N_i^{u}(u,v) N_j^{v}(u,v) \\ +\beta_3 N_i^{v}(u,v) N_j^{v}(u,v)] du dv + \gamma_1 N_i(u,v) N_j(u,v) \end{aligned} \tag{20}$$

where $N_k^{uu}(u,v)$ is the second derivative of $N_k(u,v)$ with respect to u, ... etc., and $i, j = 0, 1, ..., \Delta$. α_1,α_2, α_3 and α_4 are constants related to bending; β_1, β_2 and β_3 are constants related to stretching; and γ_1 is a constant for the spring effect. Since the bending energy affects surface curvature, while the stretching energy affects surface area and the spring energy affects the amount of surface displacement, we should adjust the weight constants according to practical requirement during deformation.

The quadratic equation $E(\bar{Q} - \hat{Q})$ has a minimum at the point where $A(\bar{Q} - \hat{Q}) = 0$. Hence, if the optimization process is performed with no constraints, the resulting surface will be exactly the same as the scaled surface.

7 Setting Up Constraints

Two types of constraints, *boundary constraint* and *trimming curve constraint*, are used to control the deformation process.

7.1 Boundary Constraint

The boundary constraint is to prevent the boundary curves of the scaled surface from being changed during the deformation process. Otherwise, the deformed surface might not fit adjacent surfaces (parts) well. For simplicity, we shall assume that the NURBS surface has multiple knots at the beginning and end of its knot vectors. In this case, the boundary curves of the surface are determined by the control points on the boundary of the control net only.

The C^0-continuity boundary constraint for the deformation process is

$$\bar{Q}_{0,j} = \hat{Q}_{0,j}, \quad \bar{Q}_{m,j} = \hat{Q}_{m,j}, \quad j = 0, \ldots, n; \tag{21}$$

$$\bar{Q}_{i,0} = \hat{Q}_{i,0}, \quad \bar{Q}_{i,n} = \hat{Q}_{i,n}, \quad i = 1, \ldots, m-1. \tag{22}$$

where $\hat{Q}_{i,j}$ are the control points of the scaled surface $T_s \circ S(u,v)$ and $\bar{Q}_{i,j}$ are the control points of the deformed surface

For a surface without multiple knots, one can fix the boundary of the surface by fixing a band of control points that defines the boundary curves of the surface. The subdivision process performed in the first step guarantees that enough unconstrained control points will still be available for the deformation process with the boundary constraint. If G^1 continuity is required, additional equations should be included as follows:

$$\bar{Q}_{1,j} = \hat{Q}_{1,j}, \quad \bar{Q}_{m-1,j} = \hat{Q}_{m-1,j}, \quad j = 1, \ldots, n-1; \tag{23}$$

$$\bar{Q}_{i,1} = \hat{Q}_{i,1}, \quad \bar{Q}_{i,n-1} = \hat{Q}_{i,n-1}, \quad i = 2, \ldots, m-2. \tag{24}$$

7.2 Trimming Curve Constraint

The *trimming curve constraint* is to ensure that, after deformation, the relocated trimming curves $R_i \circ S \circ C_i(t)$ become trimming curves of the deformed surface $\bar{S}(u,v)$, i.e., vertices $P_{i,j} = R_i \circ S \circ C_i(t_{i,j})$ of the relocated trimming curves $R_i \circ S \circ C_i(t)$ are points of the surface $\bar{S}(u,v)$. Hence, for each $1 \leq i \leq r$, there must exist $(u_{i,j}, v_{i,j})$ in the domain of $\bar{S}(u,v)$ so that

$$P_{i,j} = \sum_{k=0}^{\Delta} Q_k N_k(u_{i,j}, v_{i,j}), \tag{25}$$

$j = 1, 2, ..., n_i$, where Δ, Q_k and $N_k(u,v)$ are defined in (14), (15), and (16). The best choice for $(u_{i,j}, v_{i,j})$ is the parameters of the point $\bar{P}_{i,j}$ constructed in Section 5. The process of computing $\bar{P}_{i,j}$ in Section 5 would actually find its parameters first. Therefore, the values of $(u_{i,j}, v_{i,j})$ are already available at this stage.

The equations in (25) can be put into matrix form as follows

$$\bar{B}\, Q = \bar{\mathbf{b}} \tag{26}$$

where Q is defined in (18), and $\bar{B}$ and $\mathbf{b}$ are defined as follows:

$$\bar{B} = \begin{bmatrix} b_{1,1} & b_{1,2} & \ldots & b_{1,\Delta} \\ \vdots & \vdots & & \vdots \\ b_{M,1} & b_{M,2} & \ldots & b_{M,\Delta} \end{bmatrix} \tag{27}$$

$$\bar{\mathbf{b}} = [P_{1,1}, \ldots, P_{1,n_1}, \ldots, P_{1,n_r}, \ldots, P_{r,n_r}]^\top \tag{28}$$

with

$$M = n_1 + n_2 + \ldots + n_r$$

$$b_{i,j} = N_j(u_{k,l}, v_{k,l})$$

where k is the smallest integer such that $i \leq n_1+n_2+\ldots+n_k$ and $l = i - (n_1 + n_2 + \ldots + n_{k-1})$.

The trimming curve constraint and the boundary constraint can be merged into a single linear constraint system.

8 The Deformation Process

The optimization process of the quadratic equation with the linear constraints can be solved using the Lagrange Multiplier method, which transfers the constrained optimization problem into an unconstrained extremization problem. This follows from the observation that the solution to the quadratic optimization function $E(\bar{x}) = f(\bar{x})$ under constraint $g(\bar{x})$ is a critical point of $\bar{E}(\bar{x}) = f(\bar{x}) + \lambda g(\bar{x})$. λ is called the Lagrange Multiplier.

For our application, the optimization function is

$$E(\bar{Q} - Q) = \frac{1}{2}(\bar{Q} - Q)^\top A(\bar{Q} - Q) \tag{29}$$

and the linear constraint is

$$\bar{B}\, Q = \bar{\mathbf{b}} \tag{30}$$

By adding the Lagrange multiplier we have the following linear system:

$$\begin{bmatrix} A & \bar{B}^\top \\ \bar{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{a} \\ \bar{\mathbf{b}} \end{bmatrix} \tag{31}$$

where

$$\mathbf{a} = \mathbf{A}\, \mathbf{Q} \tag{32}$$

Solving the linear system (31) will obtain the control points of the desired deformed surface.

9 Implementation

An issue about degree of freedom. should be noted when implementing the above approach. The linear system constructed in Section 8 could be overly determined due to the fact that the number of control points of a surface patch is finite. This problem is resolve by performing subdivision in Section 4 to fulfill the third requirement. We first determine the band that has influence on the trimming curves and then insert some new knots and, consequently, some new control points into this band. Thus we can avoid the overly-determining problem.

The implementation of the above method is carried out using B-spline representation. For NURBS representation, one simply repeats this method first in the 4D space and then project the result back into the 3D space. Since the above method uses the control points as the variables, this can be easily achieved.

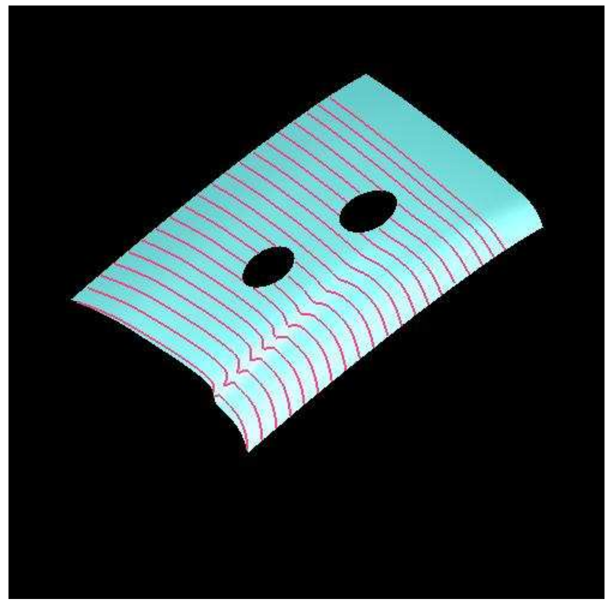

Figure 1: Trimmed door panel before scaling.

Test results on two data sets are presented here. These data sets include a trimmed door panel (Figures 1-2), and a trimmed front hood (Figures 3-4). The front hood is a degree 3×5 NURBS surface with 8 patches. 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.

Two images are shown for each case: the first one shows the trimmed surface before the scaling process and the second one shows the result of the constrained scaling process. The scaling factors for the two cases are: $S_x = 1.15$, $S_y = 1.2$, $S_z = 1.3$. The shaded trimmed surfaces before scaling and after scaling are displayed with a set of highlight lines [8][1][2]. 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 wireframe drawings or shaded pictures [9][28].

From the images, one can see in each case that the result of the constrained scaling has the same features as the

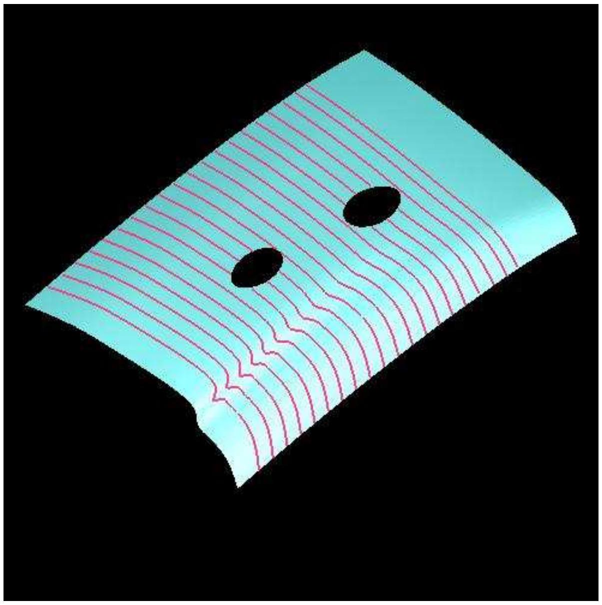

Figure 2: Trimmed door panel after scaling.

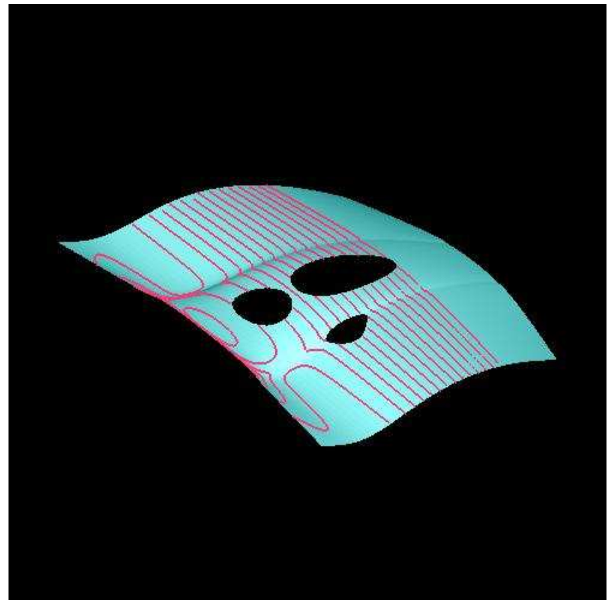

Figure 3: Trimmed front hood before scaling.

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 deformation process.

10 Conclusion

This paper presents a deformation based approach for constrained surface scaling of trimmed NURBS surfaces. The new surface is formed by scaling the given surface first, and then attaching the (original) features to the scaled NURBS surface at appropriate locations. The attaching process requires several geometric operations and constrained free-form surface deformation. The resulting surface has the same features as the original surface and same boundary curves as the scaled surface while reflecting the shape and curvature distribution of the scaled surface. The resulting surface also maintains a NURBS representation and, hence, is compatible with most of the current data-exchange standards.

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

Acknowledgements. Financial support of our research work by the Honda Initiative Grant is deeply appreciated.

References

[1] Beier, K-P, and Chen, Y., The highlight-line algorithm for real-time surface-quality assessment, *Computer-Aided Design* 26, 1994, 268-278.

[2] Beier, K-P, and Chen, Y., The highlight band, a simplified reflection model for interactive smoothness evaluation, in: Spapidis, N., ed., *Designing Fair Curves and Surfaces*, SIAM, 1994.

[3] Borrel, P. and Rappoport, A., Simple Constrained Deformations for Geometric Modeling and Interactive Design, *ACM Tansactions on Graphics*, 13(2), 1994, 137-155.

[4] Cavendish, J. C. and Marin, S. P., A procedural Feature-Based Approach for Designing Functional Surfaces, In *Topics in Surface Modeling*, Hans Hagen, editor, SIAM 1992, 145-168.

[5] Celniker, G. and Gossard, D., Deformable curve and surface finite elements for free form shape design, *Computer Graphics*, 25(4), 1991, 257-266.

[6] Celniker, G. and Welch, W., Linear constraints for deformable B-spline surfaces, In *Proceedings of the Sym-*

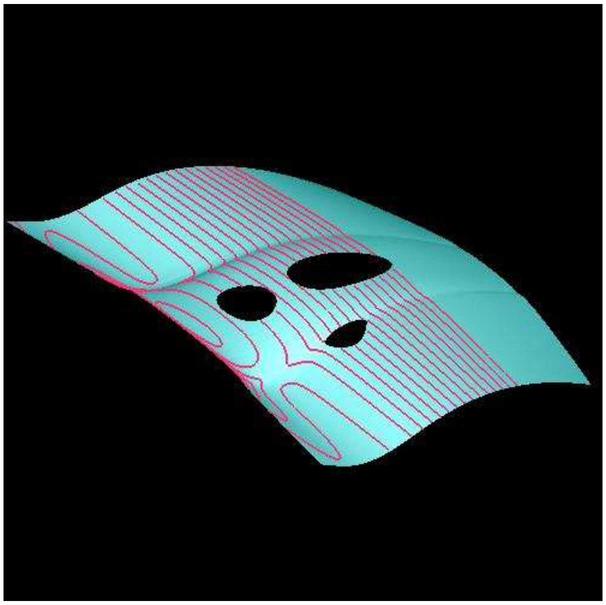

Figure 4: Trimmed front hood after scaling.

posium on Interactive 3D Graphics, ACM. New York, 1992, 165-170.

[7] Chen, Y., Beier, K-P, and Papageorgiou, D., Direct highlight line modification on nurbs surface, *Computer Aided Geometric Design* 14, 1997, 583-601.

[8] Coquillart, S., Extended free form deformation: A Sculpting tool for 3D geometric modeling, *Computer Graphics*, 24(4), 1990, 187-193.

[9] Courant, R. and Hilbert, D., Methods of Mathematical Physics, Vol. I, Interscience, London, 1953.

[10] Feng, J., Ma, L. and Peng, Q., A new free-form deformation through the control of frametic surfaces, *Computer & Graphics*, 20(4), 1996, 531-539.

[11] Griessmair, J. and Parray, R., Deformation of Solids with Trivariate B-spline, *Proceedings of Eurographics'89*, 1989, 137-148.

[12] Hsu, W. H., Hughes, J. F. and Kaufman, H., Direct manipulation of free-form deformations, *Computer Graphics*, 26(2), 1992, 177-184.

[13] Kalra, P., et al., Simulation of facial muscle actions based on rational free-form deformations, *Computer Graphics Forum*, 2(3), 1992, C59-C69.

[14] Lamousin, H. J. and Waggenspack, W. N., NURBS-based free-form deformations, *IEEE Transactions on Computer Graphics & Applications*, 14(6), 1994, 59-65.

[15] Mansfield, E. H., The Bending and Stretching of Plates, Machillan, New York, 1964.

[16] Platt, J. and Barr, A., Constraints method for flexible models, *Computer Graphics*, 22(4), 1988, 279-288.

[17] Sedbergerg, T. W. and Parry, S. R., Free-form deformation of solid geometric models, *Computer Graphics*, 20(4), 151-160.

[18] Sarraga, R. F., Recent Methods for Surface Shape Optimization, *Computer Aided Geometric Design*, 15(1998), 417-436

[19] Terzopoulos, D., Multilevel Computational Processes for Visual Surface Reconstruction, *Computer Vision, Graphics, and Image processing*, Vol. 24, 1983, 52-96.

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

[21] Terzopoulos, D. and Fleischer, K., Deformable models, *Visual Computing*, 4(6), 1988, 306-331.

[22] Terzopoulos, D. and Qin, H., Dynamic NURBS with Geometric Constraints for interactive Sculpting, *ACM Transactions on Graphics*, 13(2), 1994, 103-136.

[23] Ugural, A. C., Ferster, S. K., Advanced Strength and Applied Elasticity, America Elsevier Publishing Co., New York, 1975.

[24] Welch, W. and Witkin, A., Variational surface modeling, *Computer Graphics*, 26(2), 1992, 157-166.

[25] Zhang, C. and Cheng, F., Removing Local Irregularities of NURBS Surfaces by Modifying Highlight Lines, *Computer-Aided Design*, 30(12), 1998, 923-930.