Computer-Aided Design 33 (2001) 913–923

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Fairing spline curves and surfaces by minimizing energy ☆

Caiming Zhang, Pifu Zhang, Fuhua (Frank) Cheng*

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

Received 26 June 1999; revised 25 July 2000; accepted 16 August 2000

Abstract

New algorithms for the classical problem of fairing cubic spline curves and bicubic spline surfaces are presented. To fair a cubic spline curve or a bicubic spline surface with abnormal portions, the algorithms (automatically or interactively) identify the 'bad' data points and replace them with new points produced by minimizing the strain energy of the new curve or surface. The proposed algorithms are more general than the existing algorithms in that the new algorithms can adjust more than one 'bad' data point in each modification step and they include the existing algorithms [Computer-Aided Design 15(5) (1983) 288–293; 28 (1996) 59–66] as special cases. Test results of the new algorithms are included. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Splines curves/surfaces; Curve/surface fairing; Interpolation; Smoothness; Strain energy

1. Introduction

Fairing refers to the process of detecting and removing irregularities of a curve or surface. This is an important part of shape design. The current process relies heavily on the designers to visually identify regions with curvature irregularities and to fix them manually by, for instance, correcting the control points of the curve or surface. This is often an experience-based, trial-and-error, and time-consuming process. Thus, computer-assisted detecting and removal of local curve/surface curvature anomalies are in high demand from the design community.

Interactive fairing techniques for cubic spline curves have been proposed by Kjellander [9] and Farin et al. [5]. The designer identifies the data point to be faired, the curve is then faired by making a small adjustment to that point and fitting a new curve. This process can be iteratively repeated until a satisfactory curve is obtained. The idea from Ref. [9] has been extended to cover bicubic spline surfaces [10]. Limitations of these methods are discussed in Ref. [12]. One shortcoming of Kjellander's approach is that it is suitable for uniformly parametrized cubic spline curves and surfaces only.

An automatic fairing algorithm for B-spline curves is first proposed by Sapidis and Farin [18]. The data point with the biggest jump in curvature variation is identified and the curve is faired by changing the position of that point. An automatic fairing algorithm for cubic spline curves is recently presented by Poliakoff [15,16], which is an extension of Kjellander's method [9] for non-uniformly parametrized cubic spline curves. These methods fair a single data point in each modification step. In some cases, unfortunately, fairing a single point alone in each step cannot lead to a desired result. An example is given in Section 5.

In the design of free-form surfaces, curvatures are frequently used in analyzing the shape of a surface. However, isophotes [14], reflection lines [8,11], and *highlight lines* [1,2] have been proven to be more effective in assessing the quality of a surface. Several methods [8,11,20] for removing abnormal portions of a surface have been developed based on these models. These methods are effective in handling certain surface fairing problems.

This paper addresses the problem of fairing cubic spline curves and bicubic spline surfaces. Algorithms are presented for both the interactive and automatic fairing environments. To fair a cubic spline curve or a bicubic spline surface, the algorithms (automatically or interactively) identify the 'bad' points (using curvature plot for a curve and highlight line model for a surface) and replace them with points generated by minimizing the strain energy of the new spline curve/surface. This approach makes much sense since it is in line with the spline method. The proposed algorithms are more general than the existing algorithms in that the new algorithms can adjust more than one 'bad' point in one modification step and they include the existing algo-

☆ Research work of this project is supported by Ford and NSF (DMI-9912069).

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

0010-4485/01/$ - see front matter
PII: S0010-4485(00)00114-7

rithms [10,15] as special cases. The presented algorithms can also be used to fair B-spline curves/surfaces. The abnormal portions of a cubic B-spline curve or a bicubic B-spline surface is removed by fairing the interpolant to its control points.

The remaining part of the paper is arranged as follows. The basic idea in developing the algorithms is described in Section 2. The algorithms for fairing cubic spline curves are described in Section 3. The algorithms for fairing bicubic spline surfaces are described in Section 4. Two examples showing the iterative modification process of the new algorithms are discussed in Section 5. Concluding remarks are given in Section 6.

2. Basic idea

Let $C(t)$ be a cubic spline curve that interpolates a set of data points P_i at knot t_i, $i = 0, 1, \ldots, n$. Let M_i denote the derivative of $C(t)$ at t_i and $C_i(t)$ denote the segment of $C(t)$ on the interval $[t_i, t_{i+1}]$. $C_i(t)$ is defined by

$$C_i(t) = \varphi_0(s)P_i + \varphi_1(s)h_iM_i + \psi_1(s)h_iM_{i+1} + \psi_0(s)P_{i+1}, \quad (1)$$

where

$$\varphi_0(s) = (s-1)^2(2s+1), \qquad \varphi_1(s) = (s-1)^2 s,$$

$$\psi_0(s) = s^2(-2s+3), \qquad \psi_1(s) = s^2(s-1)$$

are cubic Hermite basis functions and $s = (t - t_i)/h_i$ with $h_i = t_{i+1} - t_i$.

M_i's can be determined from the condition that $C(t)$ is C^2-continuous. This gives the conditions that, for $i = 1, 2, \ldots, n-1$,

$$a_iM_{i-1} + 2M_i + b_iM_{i+1} = -\frac{3a_i}{h_{i-1}}P_{i-1} + c_iP_i + \frac{3b_i}{h_i}P_{i+1}, \quad (2)$$

where

$$a_i = h_i/(h_{i-1} + h_i),$$

$$b_i = h_{i-1}/(h_{i-1} + h_i),$$

$$c_i = 3(a_i/h_{i-1} - b_i/h_i).$$

Hence, with two end point conditions, $C(t)$ can be uniquely determined by Eq. (2). In this paper, the free-end conditions [4] will be used and the constructed $C(t)$ is called a *natural spline* curve.

However, following the *minimum curvature property* [7], if the following form is used as the strain energy of $C(t)$

$$E = \int_{t_0}^{t_n} C''(t)^2 \, dt, \quad (3)$$

then the conditions in Eq. (2) can also be derived from minimizing the strain energy at M_i's

$$\frac{\partial E}{\partial M_i} = 0, \qquad i = 1, 2, \ldots, n-1. \quad (4)$$

The basic idea of our algorithms is based on the *minimum curvature property* and may be described as follows. If a cubic spline curve $C(t)$ has abnormal portions near the interpolation points $P_{i_1}, P_{i_2}, \ldots, P_{i_r}$ (for simplicity, these points are assumed to be consecutive), we remove the abnormal portions of the curve by fairing these 'bad' points. The new locations of the 'bad' points after the fairing process are denoted $\bar{P}_{i_1}, \bar{P}_{i_2}, \ldots, \bar{P}_{i_r}$, and the corresponding new curve segments are denoted $\bar{C}_{i_1-1}(t), \bar{C}_{i_1}(t), \ldots, \bar{C}_{i_r}(t)$. $\bar{P}_{i_1}, \bar{P}_{i_2}, \ldots, \bar{P}_{i_r}$ can be determined in many different ways. However, with $C(t)$ being produced by minimizing the strain energy $\int_{t_0}^{t_n} C''(t)^2 \, dt$, a natural and logical choice in determining $\bar{P}_{i_1}, \bar{P}_{i_2}, \ldots, \bar{P}_{i_r}$ would be to minimize the strain energy of the new curve at these points, i.e.

$$\frac{\partial E}{\partial \bar{P}_{i_j}} = 0, \qquad j = 1, 2, \ldots, r. \quad (5)$$

Replacing $P_{i_1}, P_{i_2}, \ldots, P_{i_r}$ with $\bar{P}_{i_1}, \bar{P}_{i_2}, \ldots, \bar{P}_{i_r}$, respectively, in the related curve segments of $C(t)$ would result in a curve that has smaller energy than $C(t)$ but is not C^2-continuous at the knots $t_{i_1}, t_{i_2}, \ldots, t_{i_r}$. Such a curve, called $\tilde{C}(t)$, certainly can not be thought of as fairer than the original curve $C(t)$. However, if we construct a new cubic spline curve $\bar{C}(t)$ to interpolate the faired data points, since the energies of $\bar{C}(t)$ related segments of $\bar{C}(t)$ are smaller than the energies of corresponding segments of $\tilde{C}(t)$ due to the *minimum curvature property*, and energies of these segments of $\tilde{C}(t)$ are smaller than the energies of corresponding segments of $C(t)$, we have a C^2-continuous curve $\bar{C}(t)$ that has smaller strain energy than $C(t)$. Consequently, $\bar{C}(t)$ is fairer than $C(t)$ and we have the following theorem.

Theorem 1. *Let $C(t)$ denote the original cubic spline curve and $\bar{C}(t)$ denote the spline curve that interpolates the faired data points. We have*

$$\int_{t_0}^{t_n} \bar{C}''(t)^2 \, dt \le \int_{t_0}^{t_n} C''(t)^2 \, dt$$

if knots of $\bar{C}(t)$ are the same of those of $C(t)$.

Details of the fairing process for cubic spline curves and bicubic spline surfaces will be shown in Sections 3 and 4, respectively.

3. Fairing cubic spline curves

The setting of Eq. (5) allows several consecutive points to be faired in a single step. However, fairing more than two consecutive points in a single step could result in curve segments quite different from the original ones. This would violate the shape preserving requirement of the fairing process. Hence, we shall consider the problem of fairing one or two points in each step only.

3.1. Fairing one point

Let P_i be a 'bad' point that needs to be replaced with a new point $\bar{P}_i$. Since $\bar{P}_i$ is involved in $\bar{C}_{i-1}(t)$ and $\bar{C}_i(t)$ only, P_i can be determined by minimizing the strain energy of $\bar{C}_{i-1}(t)$ and $\bar{C}_i(t)$ at $\bar{P}_i$, i.e. setting the derivative of $E(\bar{P}_i)$ with respect to $\bar{P}_i$ to zero

$$\frac{\partial E(\bar{P}_i)}{\partial \bar{P}_i} = 0 \tag{6}$$

where

$$E(\bar{P}_i) = \int_{t_{i-1}}^{t_i} \bar{C}''_{i-1}(t)^2\,dt + \int_{t_i}^{t_{i+1}} \bar{C}''_i(t)^2\,dt. \tag{7}$$

By solving Eq. (6) one gets

$$\begin{aligned}\bar{P}_i = {} & \lambda_i P_{i-1} + \mu_i P_{i+1} + \tfrac{1}{2}h_{i-1}\lambda_i M_{i-1} \\ & + \tfrac{1}{2}(h_{i-1}\lambda_i - h_i\mu_i)M_i - \tfrac{1}{2}h_i\mu_i M_{i+1},\end{aligned} \tag{8}$$

where

$$\lambda_i = h_i^3/h_{i-1}^3 + h_i^3; \qquad \mu_i = h_{i-1}^3/h_{i-1}^3 + h_i^3$$

Poliakoff [15] reached the same formula by requiring that

$$\bar{C}'''_{i-1}(t_i) = \bar{C}'''_i(t_i)$$

in extending Kjellander's method [9]. Eq. (6) seems to be a more reasonable approach.

Following the idea presented in Section 2, a new cubic spline curve is then constructed to interpolate P_j, $j = 0, 1, \ldots, n$ (with P_i replaced with $\bar{P}_i$). The following theorem shows that this process can be iteratively repeated until P_i can not be improved any further, i.e. when $P_i = \bar{P}_i$.

Theorem 2. *For any given points P_j and derivatives M_j at the knot t_j, $j = i-1, i, i+1$, let $Q_i(t)$ be the cubic Hermite interpolant to the points and derivatives at t_{i-1} and t_{i+1}, $\bar{Q}_j(t)$ the cubic Hermite interpolant to the points and derivatives at t_j and t_{j+1}, $j = i-1, i$, then*

$$\int_{t_{i-1}}^{t_{i+1}} Q''_i(t)^2\,dt \le \int_{t_{i-1}}^{t_i} \bar{Q}''_{i-1}(t)^2\,dt + \int_{t_i}^{t_{i+1}} \bar{Q}''_i(t)^2\,dt. \tag{9}$$

Proof. It is sufficient to show that if $\bar{P}_i$ and M_i in $\bar{C}_{i-1}(t)$ and $\bar{C}_i(t)$ are defined by

$$\frac{\partial E(\bar{P}_i)}{\partial \bar{P}_i} = 0 \text{ and } \frac{\partial E(\bar{P}_i)}{\partial M_i} = 0 \tag{10}$$

where $E(\bar{P}_i)$ is defined in Eq. (7), then $\bar{C}_{i-1}(t) \equiv \bar{C}_i(t)$. Straightforward computation shows that Eq. (10) is equivalent to $\bar{C}'''_{i-1}(t_i) = \bar{C}'''_i(t_i)$ and $\bar{C}''_{i-1}(t_i) = \bar{C}''_i(t_i)$. Hence, $\bar{C}_{i-1}(t) \equiv \bar{C}_i(t)$. □

Eq. (8) can be used to fair P_i repeatedly until a desired result is obtained. Theorem 2 shows that if Eq. (8) is repeatedly used to fair P_i, $\bar{P}_i$ will eventually satisfy the following condition

$$\bar{P}_i = Q_i(t_i) \tag{11}$$

where Q_i(t) is a cubic Hermite interpolant to the positions and the derivatives of the cubic spline curve $C(t)$ at t_{i-1} and t_{i+1}. This final result can be obtained simply by constructing a cubic spline curve that interpolates the points $P_0, P_1, \ldots, P_{i-1}, P_{i+1}, P_{i+2}, \ldots, P_n$.

Note that if $h_{i-1} = h_i$, (8) and (11) both reduces to Kjellander's algorithm [9]. Kjellander's algorithm constructs the final curve by solving a system of linear equations in n unknowns. But the coefficient matrix of the system is not tri-diagonal. Based on the above discussion, the final curve can be produced by constructing a cubic spline curve that interpolates the points $P_0, P_1, \ldots, P_{i-1}, P_{i+1}, P_{i+2}, \ldots, P_n$. This is a process of solving a system of $n-2$ tri-diagonal equations in $n-2$ unknowns.

Poliakoff claims in Ref. [15] that if only one data point is faired in a modification step, then no matter how many times the curve is faired, the resulting curve can never be a straight line. However, from Theorems 1 and 2 and the above discussion, it is easy to see that this is not true.

3.2. Fairing two points

Let P_i and P_{i+1} be two consecutive 'bad' points that need to be replaced with new points $\bar{P}_i$ and $\bar{P}_{i+1}$, respectively. Since $\bar{P}_i$ and $\bar{P}_{i+1}$ are involved in $\bar{C}_{i-1}(t)$, $\bar{C}_i(t)$ and $\bar{C}_{i+1}(t)$ only, they can be determined by minimizing the strain energy of $\bar{C}_{i-1}(t)$, $\bar{C}_i(t)$ and $\bar{C}_{i+1}(t)$ at $\bar{P}_i$ and $\bar{P}_{i+1}$, i.e.

$$\frac{\partial E(\bar{P}_i, \bar{P}_{i+1})}{\partial \bar{P}_j} = 0, \qquad j = i, i+1,$$

where

$$E(\bar{P}_i, \bar{P}_{i+1}) = \sum_{j=i-1}^{i+1} \int_{t_j}^{t_{j+1}} \bar{C}''_j(t)^2\,dt. \tag{12}$$

By solving these equations one gets

$$\begin{aligned}\bar{P}_i = {} & \tfrac{1}{2}[(1-\lambda_i)h_{i-1}M_{i-1} + ((1-\lambda_i)h_{i-1} - \lambda_i h_i)M_i \\ & - \lambda_i(h_i + h_{i+1})M_{i+1} - \lambda_i h_{i+1}M_{i+2}] \\ & + (1-\lambda_i)P_{i-1} + \lambda_i P_{i+2},\end{aligned} \tag{13}$$

$$\begin{aligned}\bar{P}_{i+1} = {} & \tfrac{1}{2}[\mu_i h_{i-1}M_{i-1} + \mu_i(h_{i-1} + h_i)M_i \\ & + (\mu_i h_i - (1-\mu_i)h_{i+1})M_{i+1} - (1-\mu_i)h_{i+1}M_{i+2}] \\ & + \mu_i P_{i-1} + (1-\mu_i)P_{i+2},\end{aligned}$$

where

$$\lambda_i = h_{i-1}/(h_{i-1} + h_i + h_{i+1}),$$

$$\mu_i = h_{i+1}/(h_{i-1} + h_i + h_{i+1}).$$

A new cubic spline curve is then constructed to interpolate the faired data points. This process can be iteratively repeated until a desired result is obtained. If the modification process is repeated sufficiently many times, one shall reach a stage that none of P_i and P_{i+1} can be improved any further. The following theorem shows how to reach that stage in one modification step. The theorem follows directly from Theorems 1 and 2.

Theorem 3. If $C(t)$ is a cubic spline curve interpolating data points P_i at knot t_i, $i = 0, 1, \ldots, n$, and having end point conditions M_0 and M_n, then

$$\int_{t_0}^{t_n} C''(t)^2 \, dt \geq \int_{t_0}^{t_n} Q''(t)^2 \, dt, \tag{14}$$

for any cubic spline interpolant $Q(t)$ to P_0 and M_0 at t_0, and P_n and M_n at t_n.

Theorem 3 shows that if the conditions in Eq. (13) are repeatedly used to fair P_i and P_{i+1}, $\bar{P}_i$ and $\bar{P}_{i+1}$ will eventually satisfy the following conditions:

$$\bar{P}_i = Q_i(t_i) \text{ and } \bar{P}_{i+1} = Q_i(t_{i+1}) \tag{15}$$

where $Q_i(t)$ is the cubic Hermite interpolant to the positions and derivatives of the cubic spline curve $C(t)$ at t_{i-1} and t_{i+2}. Thus the final curve is a cubic spline curve interpolating the data points $P_0, P_1, \ldots, P_{i-1}, P_{i+2}, P_{i+3}, \ldots, P_n$. Consequently, one can obtain the final result in one modification step simply by constructing a cubic spline curve that interpolates the data points $P_0, P_1, \ldots, P_{i-1}, P_{i+2}, P_{i+3}, \ldots, P_n$.

Theorem 3 also shows that modifying more than two consecutive points in one fairing step will result in a curve quite different from the original one.

3.3. Algorithm for fairing cubic curves

To identify the 'bad' points on a curve, a fairness criterion is needed. The *local fairness indicator* of a curve is defined by Sapidis [17]

$$z_i = \left| \frac{d\kappa}{ds}(t_i+) - \frac{d\kappa}{ds}(t_i-) \right| \tag{16}$$

where $d\kappa/ds$ is the derivative of the curvature with respect to the arc length $s = s(t)$ of the curve. This fairness indicator has been recommended for interactive fairing in Refs. [13,15,18]. In our algorithm, the following fairness indicator [5]

$$z_i = |C'''(t_i+) - C'''(t_i-)|, \tag{17}$$

will be used in an interactive or automatic fairing process. The reason for choosing this fairness indicator is that if the point P_i is replaced with a new point $\bar{P}_i$ formed by Eq. (11) on a cubic spline curve interpolating $P_0, P_1, \ldots, P_{i-1}, P_{i+1}, P_{i+2}, \ldots, P_n$, then the new curve $\bar{C}(t)$ satisfies the condition

$$z_i = |\bar{C}'''(t_i+) - \bar{C}'''(t_i-)| = 0.$$

Hence, a point where the third derivative of $C(t)$ has a big jump should be considered a 'bad' one.

A curvature plot is a highly sensitive indicator of the shape of a curve. A curve with a 'pleasant' curvature plot is very likely to be considered acceptable. In our method, we use curvature plot to identify 'bad' points in interactive curve fairing.

In some cases, such as digitized data points, it might be necessary to fair all the data points. Based on the above discussion, this could lead to the situation that the final curve being a straight line. To overcome this problem, a region R_i is defined for each interior data point P_i to restrain it from moving too far from its original location. This region R_i, called the restraining region for P_i, is a circle centered at P_i with radius defined by the distance from P_i to $\bar{P}_i$. $\bar{P}_i^{\cdot}$ is computed using Eq. (11). The algorithms for interactive and automatic fairing are described below.

1. Interactive faring algorithm
   1.1. Identify 'bad' data points based on curvature plot of the curve.
   1.2. For the identified points, compute their z_i's using Eq. (17), and sort them into descending order: $z_{i_1} \geq z_{i_2} \geq \cdots$.
   1.3. If $|i_1 - i_2| > 1$, modify point P_{i_1} using expression Eq. (11). Otherwise, modify P_{i_1} and P_{i_2} using expressions Eq. (15).
   1.4. Construct a new spline curve to interpolate the data points P_i, $i = 0, 1, \ldots, n$.
   1.5. Plot the curvature of the new curve. If the curvature plot is satisfactory, stop. Otherwise, goto step 2.
2. Automatic faring algorithm
   2.1. For $i = 1, 2, \ldots, n-1$, compute the restraining region R_i for P_i.
   2.2. For each P_i, $i = 1, 2, \ldots, n-1$, compute z_i using Eq. (17).
   2.3. Let z_{i_1} be the maximum of z_i's. Construct a cubic spline curve to interpolate $P_0, P_1, \ldots, P_{i_1-1}, P_{i_1+1}, \ldots, P_n$, and compute $\bar{P}_{i_1}$ using Eq. (11). If $\bar{P}_{i_1}$ is within the restraining region R_{i_1}, replace P_{i_1} with $\bar{P}_{i_1}$. Otherwise, replace P_{i_1} with the intersection point of R_{i_1} and the line segment from the center of R_{i_1} to $\bar{P}_{i_1}$.
   2.4. Construct a new spline curve to interpolate the data points P_i, $i = 0, 1, \ldots, n$.
   2.5. Compute the strain energy of the new curve. If the new energy is the same as or within a specified range of the old energy, stop. Otherwise, goto step 2.

Note. The reason that in automatic fairing only one point is allowed to be faired in a modification step is for simplicity — fairing two points in one modification will make the normalization process (putting the new point in the restraining region) not so easy to handle.

4. Fairing cubic spline surface

Let $S(u,v)$ be a bicubic spline surface interpolating a set of data points $P_{i,j}$, $i = 0,1,\ldots,m$, $j = 0,1,\ldots,n$. The knots corresponding to $P_{i,j}$ are (u_i,v_j). The first derivatives of $S(u,v)$ with respect to u and v at (u_i,v_j) are denoted $M^u_{i,j}$ and $M^v_{i,j}$, respectively. The patch of $S(u,v)$ corresponding to the parametric region $[u_i,u_{i+1}]\times[v_j,v_{j+1}]$ is denoted $S_{i,j}(u,v)$. The surface is constructed by generating natural cubic spline curves in u and v directions first (see Section 2), and then generating a bicubic spline surface to interpolate the network of curves [6]. Each patch of $S(u,v)$ is determined by four data points, partial derivatives with respect to u and v, and twist vectors, at these points. The twist vectors are computed from natural cubic spline curves in u direction that interpolate the partial derivative of $S(u,v)$ with respect to v at the data points. For a survey and more techniques in this direction, see Ref. [19].

In the following, we will consider removing an abnormal portion of a spline surface by modifying one, two, or four 'bad' points in each modification step.

4.1. Fairing one point

Let $P_{i,j}$ be a 'bad' data point to be faired. $P_{i,j}$ is involved in two spline curves: one interpolates data points $P_{i,k}$, $k = 0,1,\ldots,n$ and one interpolates data points $P_{l,j}$, $l = 0,1,\ldots,m$. By extending the idea of Section 3.1, one can determine the new position $\bar{P}_{i,j}$ of $P_{i,j}$ by minimizing the following objective function with respect to $\bar{P}_{i,j}$

$$E(\bar{P}_{i,j}) = \sum_{l=i-1}^{i}\int_{u_l}^{u_{l+1}}\left[\frac{\partial^2\bar{S}_{l,j}}{\partial u^2}(u,v_j)\right]^2 \mathrm{d}u + \sum_{k=j-1}^{j}\int_{v_k}^{v_{k+1}}\left[\frac{\partial^2\bar{S}_{i,k}}{\partial v^2}(u_i,v)\right]^2 \mathrm{d}v \tag{18}$$

where $\bar{S}_{l,j}(u,v_j)$ is a cubic Hermite interpolant obtained by substituting $\bar{P}_{i,j}$ for $P_{i,j}$ in $S_{l,j}(u,v_j)$, and $\bar{S}_{i,k}(u_i,v)$ is a cubic Hermite interpolant obtained by substituting $\bar{P}_{i,j}$ for $P_{i,j}$ in $S_{i,k}(u_i,v)$, i.e. solving the following equation for $\bar{P}_{i,j}$:

$$\frac{\mathrm{d}E(\bar{P}_{i,j})}{\mathrm{d}\bar{P}_{i,j}} = 0. \tag{19}$$

The solution is

$$\bar{P}_{i,j} = (2(\bar{u}^3_{i-1}P_{i-1,j} + \bar{u}^3_i P_{i+1,j} + \bar{v}^3_{j-1}P_{i,j-1} + \bar{v}^3_j P_{i,j+1}) + \alpha_{i,j} + \beta_{i,j})/\tau_{i,j} \tag{20}$$

where

$$\alpha_{i,j} = \bar{u}^2_{i-1}M^u_{i-1,j} - (\bar{u}^2_i - \bar{u}^2_{i-1})M^u_{i,j} - \bar{u}^2_i M^u_{i+1,j},$$

$$\beta_{i,j} = \bar{v}^2_{j-1}M^v_{i,j-1} - (\bar{v}^2_j - \bar{v}^2_{j-1})M^v_{i,j} - \bar{u}^2_j M^u_{i,j+1},$$

$$\tau_{i,j} = 2(\bar{v}^3_{i-1} + \bar{u}^3_i + \bar{v}^3_{j-1} + \bar{v}^3_j)$$

with $\bar{u}_i = 1/u_i$ and $\bar{v}_j = 1/v_j$.

Replacing $P_{i,j}$ with $\bar{P}_{i,j}$ in related patches of $S(u,v)$ will result in a surface with smaller strain energy on its network curves but not C^2-continuous at (u_i, v_j). Following the same idea presented in Section 2, a new bicubic spline surface that interpolates the modified data points is then constructed. The above construction process of an interpolating bicubic spline surface shows that $S(u,v)$ and the new spline surface have different first partial derivatives only at the knots (u_l,v_j), $l = 1,2,\ldots,m-1$, and (u_i,v_k), $k = 1,2,\ldots,n-1$. Hence, to construct the new spline surface, one only needs to recalculate the first partial derivatives $M^u_{l,j}$, $l = 1,2,\ldots,m-1$, and $M^v_{i,k}$, $k = 1,2,\ldots,n-1$. However, all the twist vectors need to be calculated again.

Following this approach, Eq. (20) can be used to fair $P_{i,j}$ repeatedly until a desired result is obtained. If the modification process is repeated sufficiently many times, one shall reach a stage that $P_{i,j}$ can not be faired any further, i.e. any application of Eq. (20) will simply result in a $\bar{P}_{i,j}$ that is the same as $P_{i,j}$. This follows from the fact that the fairing process converges (due to the fact that the energies of the network curves of the new surfaces generated in the iterative process form a decreasing sequence). Such a point can be obtained simply by solving the following system of equations:

$$G(M^u_{1,j}, M^u_{2,j}, \ldots, M^u_{m-1,j}, \bar{P}_{i,j}), \tag{21}$$

$$G(M^v_{i,1}, M^v_{i,2}, \ldots, M^v_{i,n-1}, \bar{P}_{i,j}),$$

$$\frac{\mathrm{d}E(\bar{P}_{i,j})}{\mathrm{d}\bar{P}_{i,j}} = 0,$$

where $G(M^u_{1,j}, M^u_{2,j}, \ldots, M^u_{m-1,j}, \bar{P}_{i,j})$ is a system of $m-1$ equations, similar to Eq. (2), with m unknowns $\{M^u_{1,j}, M^u_{2,j}, \ldots, M^u_{m-1,j}, \bar{P}_{i,j}\}$, $G(M^v_{i,1}, M^v_{i,2}, \ldots, M^v_{i,n-1}, \bar{P}_{i,j})$ is a system of $n-1$ equations, similar to Eq. (2), with n unknowns $\{M^v_{i,1}, M^v_{i,2}, \ldots, M^v_{i,n-1}, \bar{P}_{i,j}\}$, and $E(\bar{P}_{i,j})$ is defined by Eq. (18). Totally, Eq. (21) is a system of $m+n-1$ equations with $m+n-1$ unknowns. The first system of equations $G(M^u_{1,j}, M^u_{2,j}, \ldots, M^u_{m-1,j}, \bar{P}_{i,j})$ is needed in constructing the cubic spline curve that interpolates the points $P_{l,j}$, $l = 0,1,\ldots,m$, and the second system of equations $G(M^v_{i,1}, M^v_{i,2}, \ldots, M^v_{i,n-1}, \bar{P}_{i,j})$ is needed in constructing the cubic spline curve that interpolates the points $P_{i,k}$, $k = 0,1,\ldots,n$. Eq. (21) is the general version of Eq. (13) in Ref. [10] for the non-uniform case.

4.2. Fairing two points

Let $P_{i,j}$ and $P_{i+1,j}$ be two 'bad' points to be faired. By extending the idea of Section 4.1 one can compute the new locations $\bar{P}_{i,j}$ and $\bar{P}_{i+1,j}$ of $P_{i,j}$ and $P_{i+1,j}$ by minimizing the following objective function at $\bar{P}_{i,j}$ and $\bar{P}_{i+1,j}$

$$E(\bar{P}_{i,j},\bar{P}_{i+1,j}) = \sum_{l=i-1}^{i+1} \int_{u_l}^{u_{l+1}} \left[\frac{\partial^2 \bar{S}_{l,j}}{\partial u^2}(u, v_j) \right]^2 \mathrm{d}u + \sum_{l=i}^{i+1} \sum_{k=j-1}^{j} \int_{v_k}^{v_{k+1}} \left[\frac{\partial^2 \bar{S}_{l,k}}{\partial v^2}(u_l, v) \right]^2 \mathrm{d}v, \quad (22)$$

i.e. solving the following equations for $\bar{P}_{i,j}$ and $\bar{P}_{i+1,j}$

$$\frac{\partial E(\bar{P}_{i,j},\bar{P}_{i+1,j})}{\partial \bar{P}_{l,j}} = 0; \qquad l = i, i+1.$$

The above equations induce the following equations:

$$c_{1,1}\bar{P}_{i,j} + c_{1,2}\bar{P}_{i+1,j} = c_{1,3}, \quad (23)$$

$$c_{2,1}\bar{P}_{i,j} + c_{2,2}\bar{P}_{i+1,j} = c_{2,3}$$

where

$$c_{1,1} = \tau_{i,j},$$

$$c_{1,2} = c_{2,1} = -2\bar{u}_i^3,$$

$$c_{1,3} = 2(\bar{u}_{i-1}^3 P_{i-1,j} + \bar{v}_{j-1}^3 P_{i,j-1} + \bar{v}_j^3 P_{i,j+1}) + \alpha_{i,j} + \beta_{i,j},$$

$$c_{2,2} = \tau_{i+1,j},$$

$$c_{2,3} = 2(\bar{u}_{i+1}^3 P_{i+2,j} + \bar{v}_{j-1}^3 P_{i+1,j-1} + \bar{v}_j^3 P_{i+1,j+1}) + \alpha_{i+1,j} + \beta_{i+1,j}.$$

with $\bar{u}_i$, $\bar{v}_j$, $\alpha_{i,j}$, $\beta_{i,j}$ and $\tau_{i,j}$ being defined by Eq. (20).

Similar to the one point case, once $P_{i,j}$ and $P_{i+1,j}$ are replaced with $\bar{P}_{i,j}$ and $\bar{P}_{i+1,j}$, respectively, a new spline surface is constructed to interpolate the modified data points. $S(u,v)$ and the new spline surface have different first partial derivatives only at the knots (u_l, v_j), $l = 1,2,\dots,m-1$, and (u_i, v_k) and (u_{i+1}, v_k), $k = 1,2,\dots,n-1$. Hence, to reconstruct the new spline surface, one only needs to calculate the new first partial derivatives $M_{l,j}^u$, $l = 1,2,\dots,m-1$, and $M_{i,k}^v$ and $M_{i+1,k}^v$, $k = 1,2,\dots,n-1$, and all the twist vectors. Eq. (23) can be used to fair $P_{i,j}$ and $P_{i+1,j}$ repeatedly until a desired result is obtained.

4.3. Fairing four points

There are occasions that an abnormal portion of a bicubic spline surface is caused by several 'bad' points. In such a case, the abnormal portion will be faired by modifying two 'bad' points a time, starting with the 'worst' two points. If the two 'bad' points are diagonal points of a 2×2 block, we fair the entire block, as follows. The reason for taking such a strategy is that (1) it simplifies the process of selecting 'bad' points from the bad point list; (2) it allows handling of several different cases with a single formula.

Let $P_{i,j}$, $P_{i+1,j}$, $P_{i,j+1}$ and $P_{i+1,j+1}$ be four 'bad' points of $S(u,v)$. The objective function in this case is

$$E(\bar{P}_{i,j},\bar{P}_{i+1,j},\bar{P}_{i,j+1},\bar{P}_{i+1,j+1}) = \sum_{k=j}^{j+1} \sum_{l=i-1}^{i+1} \int_{u_l}^{u_{l+1}} \left[\frac{\partial^2 \bar{S}_{l,k}}{\partial u^2}(u, v_k) \right]^2 \mathrm{d}u + \sum_{l=i}^{i+1} \sum_{k=j-1}^{j+1} \int_{v_k}^{v_{k+1}} \left[\frac{\partial^2 \bar{S}_{l,k}}{\partial v^2}(u_l, v) \right]^2 \mathrm{d}v. \quad (24)$$

The new locations $\bar{P}_{i,j}$, $\bar{P}_{i+1,j}$, $\bar{P}_{i,j+1}$ and $\bar{P}_{i+1,j+1}$ are determined by solving the following equations for $\bar{P}_{i,j}$, $\bar{P}_{i+1,j}$, $\bar{P}_{i,j+1}$ and $\bar{P}_{i+1,j+1}$:

$$\frac{\partial E(\bar{P}_{i,j},\bar{P}_{i+1,j},\bar{P}_{i,j+1},\bar{P}_{i+1,j+1})}{\partial \bar{P}_{l,k}} = 0,$$

$$l = i, i+1, \quad k = j, j+1.$$

These equations induce the following equations:

$$c_{1,1}\bar{P}_{i,j} + c_{1,2}\bar{P}_{i+1,j} + c_{1,3}\bar{P}_{i,j+1} = c_{1,5}, \quad (25)$$

$$c_{2,1}\bar{P}_{i,j} + c_{2,2}\bar{P}_{i+1,j} + c_{2,4}\bar{P}_{i+1,j+1} = c_{2,5},$$

$$c_{3,1}\bar{P}_{i,j} + c_{3,3}\bar{P}_{i,j+1} + c_{3,4}\bar{P}_{i+1,j+1} = c_{3,5},$$

$$c_{4,2}\bar{P}_{i+1,j} + c_{4,3}\bar{P}_{i,j+1} + c_{4,4}\bar{P}_{i+1,j+1} = c_{4,5},$$

where

$$c_{1,1} = \tau_{i,j},$$

$$c_{1,2} = c_{2,1} = c_{3,4} = c_{4,3} = -2\bar{u}_i^3,$$

$$c_{1,3} = c_{3,1} = c_{2,4} = c_{4,2} = -2\bar{v}_j^3,$$

$$c_{1,5} = 2\bar{u}_{i-1}^3 P_{i-1,j} + 2\bar{v}_{j-1}^3 P_{i,j-1} + \alpha_{i,j} + \beta_{i,j},$$

$$c_{2,2} = \tau_{i+1,j},$$

$$c_{2,5} = 2\bar{u}_{i+1}^3 P_{i+2,j} + 2\bar{v}_{j-1}^3 P_{i+1,j-1} + \alpha_{i+1,j} + \beta_{i+1,j},$$

$$c_{3,3} = \tau_{i,j+1},$$

$$c_{3,5} = 2\bar{u}_{i-1}^3 P_{i-1,j+1} + 2\bar{v}_{j+1}^3 P_{i,j+2} + \alpha_{i,j+1} + \beta_{i,j+1},$$

$$c_{4,4} = \tau_{i+1,j+1},$$

$$c_{4,5} = 2\bar{u}_{i+1}^3 P_{i+2,j+1} + 2\bar{v}_{j+1}^3 P_{i+1,j+2} + \alpha_{i+1,j+1} + \beta_{i+1,j+1}.$$

with $\bar{u}_i$, $\bar{v}_j$, $\alpha_{i,j}$, $\beta_{i,j}$ and $\tau_{i,j}$ being defined by Eq. (20).

Similar to the above cases, once $P_{i,j}$, $P_{i+1,j}$, $P_{i,j+1}$ and $P_{i+1,j+1}$ are replaced with $\bar{P}_{i,j}$, $\bar{P}_{i+1,j}$, $\bar{P}_{i,j+1}$ and $\bar{P}_{i+1,j+1}$, respectively, a new bicubic spline surface is constructed to interpolate the modified data points. $S(u,v)$ and the new spline surface have different first partial derivatives only at the knots (u_l,v_j) and (u_l,v_{j+1}), $l=1,2,\ldots,m-1$, and (u_i,v_k) and (u_{i+1},v_k), $k=1,2,\ldots,n-1$. Hence, to construct the new spline surface, one only needs to calculate the new first partial derivatives $M^u_{l,j}$ and $M^u_{l,j+1}$, $l=1,2,\ldots,m-1$, $M^v_{i,k}$ and $M^v_{i+1,k}$, $k=1,2,\ldots,n-1$, and all the twist vectors. Eq. (25) can be used to fair $P_{i,j}$, $P_{i+1,j}$, $P_{i,j+1}$ and $P_{i+1,j+1}$ repeatedly until a desired result is obtained.

4.4. Algorithm for fairing bicubic surfaces

For interactive fairing of bicubic surfaces, we use a *highlight line model* [1,2] to identify 'bad' data points. These are data points on and near unpleasant portions of the highlight lines. A highlight line model is a family of highlight lines on a surface created by an array of parallel linear light sources. A highlight line model is sensitive to the change of normal directions and, hence, can be used to detect surface normal (curvature) irregularities. In an interactive environment, a user can assess the quality of a surface by moving or rotating an array of parallel linear light sources and examining the quality of the corresponding highlight lines. Figs. 1–8 are all highlight line modes of some surfaces.

The following extension of Eq. (17) will be used as a fairness indicator for automatic and interactive fairing of bicubic surfaces:

$$z_{i,j} = |P'''(u_i+,v_j) - P'''(u_i-,v_j)| + |P'''(u_i,v_j+) - P'''(u_i,v_j-)|. \tag{26}$$

For automatic fairing of a bicubic surface, similar to the curve case, a *restraining region* $R_{i,j}$ is needed for each interior data point $P_{i,j}$ to prevent it from moving too far from its original location (so that the resulting surface would not be much different from the original one). The restraining region $R_{i,j}$ of $P_{i,j}$ is a sphere centered at $P_{i,j}$ with radius of the distance from $P_{i,j}$ to $\bar{P}_{i,j}$. $\bar{P}_{i,j}$ is determined by Eq. (21).

Unlike the curve case where the z_i value of a faired 'bad' point P_i is always zero, the $z_{i,j}$ value of a faired 'bad' point $P_{i,j}$ in the surface case may still be a maximum. Hence, in each fairing step, the 'bad' points will all be faired, in the order from the one with the biggest $z_{i,j}$ value to the one with the smallest $z_{i,j}$ value. The algorithms are described as follows:

- Interactive fairing algorithm
  1.1. Identify 'bad' data points based on examining highlight lines of the surface.
  1.2. For the identified points, compute their $z_{i,j}$'s and sort the $z_{i,j}$'s into descending order $z_{i_1 j_1} \geq z_{i_2 j_2} \geq \cdots \geq z_{i_K j_K}$.
  1.3. Set $k=1$. Repeat the following work until $k>K$:
  If $|i_k - i_{k+1}| + |j_k - j_{k+1}| = 1$, fair points $P_{i_k j_k}$ and $P_{i_{k+1} j_{k+1}}$; $k=k+2$; else if $|i_k - i_{k+1}| = 1$ and $|j_k - j_{k+1}| = 1$, fair the four points that have $P_{i_k j_k}$ and $P_{i_{k+1} j_{k+1}}$ as their diagonal points, $k=k+2$;
  else fair P_{i_k}; $k=k+1$.
  1.4. Construct a new bicubic spline surface to interpolate the faired data points $P_{i,j}$, $i=0,1,\ldots,m$, $j=0,1,\ldots,n$.
  1.5. Create a highlight line model of the new spline surface. If the highlight line model is satisfactory or if the difference between the new strain energy and the old strain energy is less than a given tolerance, stop. Otherwise, goto Step 2.

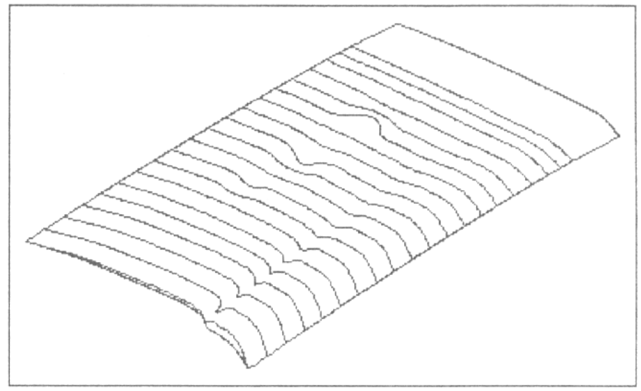


Fig. 1. Highlight line model of a bicubic spline surface with irregular portions.

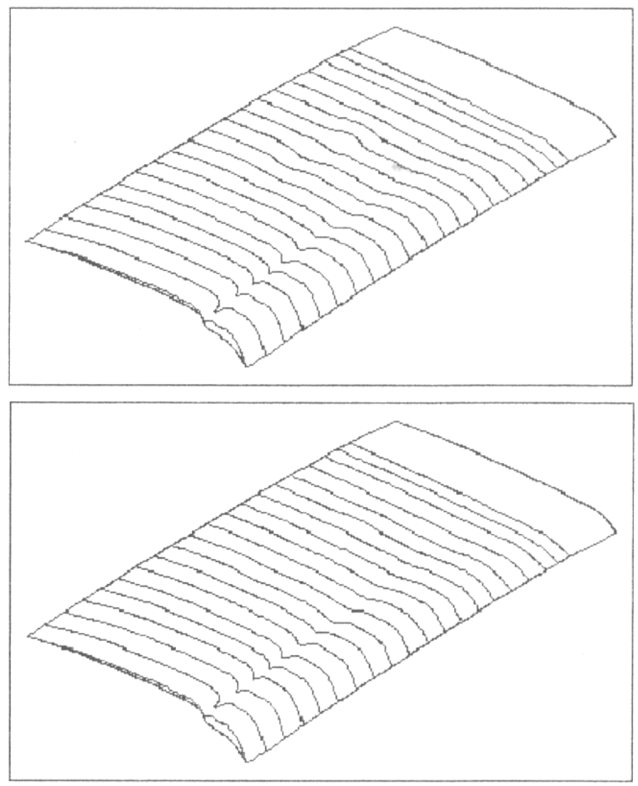


Fig. 2. Highlight line models of the surface after four iterations of (a) Kjellander's method and (b) the new method.

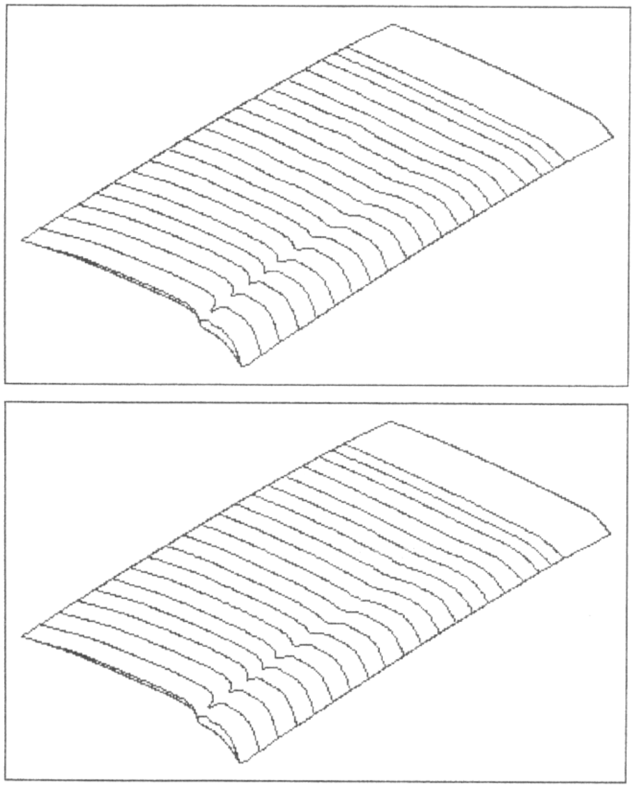


Fig. 3. Highlight line models of the surface after eight iterations of (a) Kjellander's method and (b) the new method.

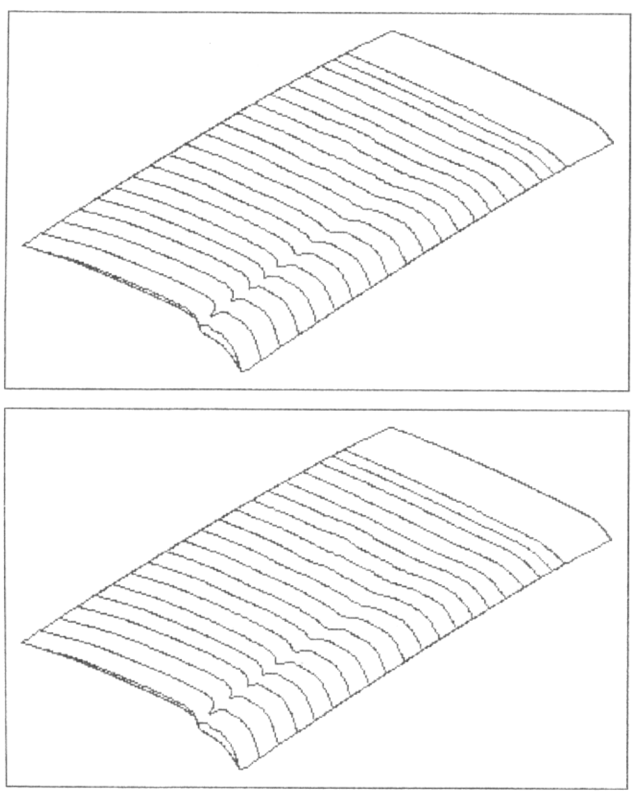


Fig. 4. Highlight line models of the surface after (a) 50 iterations and (b) 100 iterations of Kjellander's method.

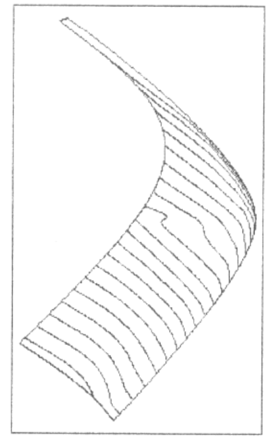


Fig. 5. Highlight line model of a bicubic NURBS surface with irregular portions.

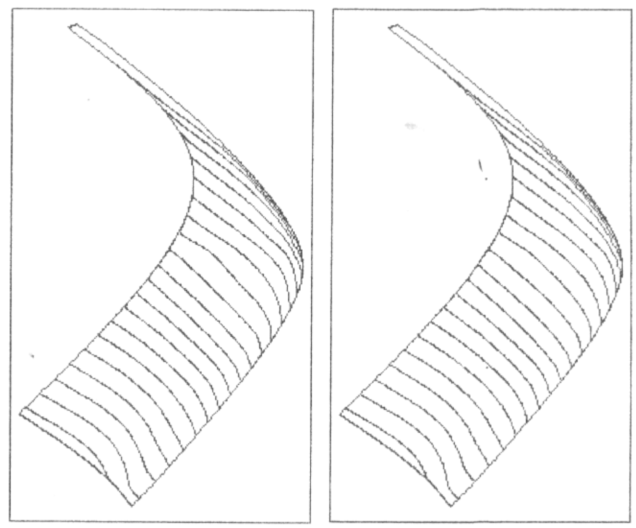


Fig. 6. Highlight line models of the surface after 15 iterations of (a) Kjellander's method and (b) the new method.

- Automatic fairing algorithm
  - 2.1. Fair data points on the boundary curves of the surface using curve fairing algorithm.
  - 2.2. Compute a restraining region $R_{i,j}$ for each interior $P_{i,j}$, $i = 1, 2, \ldots, m-1$, $j = 1, 2, \ldots, n-1$,
  - 2.3. Compute $z_{i,j}$ for each interior $P_{i,j}$, $i = 1, 2, \ldots, m-1$, $j = 1, 2, \ldots, n-1$, and sort $z_{i,j}$'s into descending order $z_{i_1,j_1} \geq z_{i_2,j_2} \geq \cdots \geq z_{i_K j_K}$.
  - 2.4. For $k = 1, 2, \ldots, K$, compute $\bar{P}_{i_k j_k}$. If $\bar{P}_{i_k j_k}$ is located in the restraining region $R_{i_k j_k}$, $P_{i_k j_k}$ is replaced with $\bar{P}_{i_k j_k}$. Otherwise, $\bar{P}_{i_k j_k}$ is replaced with the intersection point of the sphere $R_{i_k j_k}$ with the line segment between the center of $R_{i_k j_k}$ and $\bar{P}_{i_k j_k}$.
  - 2.5. Construct a new bicubic spline surface to interpolate the faired data points $P_{i,j}$, $i = 0, 1, \ldots, m$, $j = 0, 1, \ldots, n$.

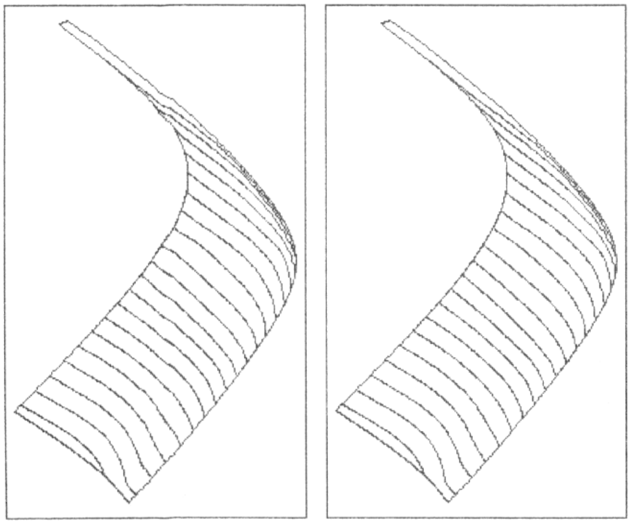


Fig. 7. Highlight line models of the surface after 30 iterations of (a) Kjellander's method and (b) the new method.

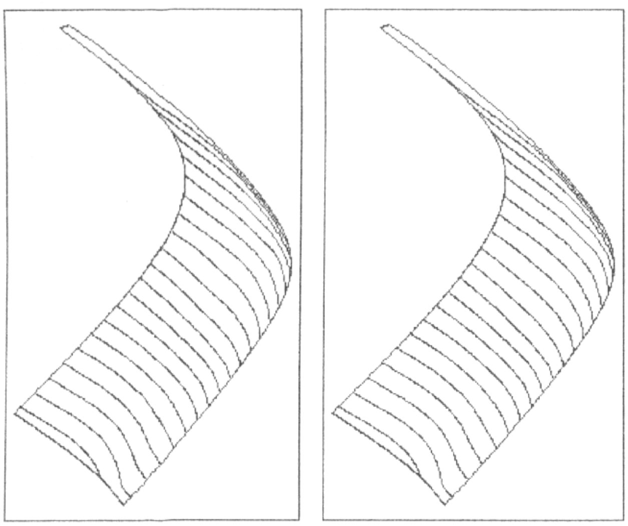


Fig. 8. Highlight line models of the surface after 45 iterations of (a) Kjellander's method and (b) the new method.

2.6. Compute strain energy of the new spline surface. If the difference between the new energy and the old energy is smaller than a specified tolerance, stop. Otherwise, goto Step 3.

4.5. Remarks

The objective functions used in Sections 4.1–4.3 are strain energies of iso-parametric cubic spline curves that pass through the data points to be faired. It is possible to define objective functions using strain energies of surface patches that interpolate the points to be faired. For instance, based on the thin plate model, one can fair one point $P_{i,j}$, two points $P_{i,j}$ and $P_{i+1,j}$, or four points $P_{i,j}$, $P_{i+1,j}$, $P_{i,j+1}$ and $P_{i+1,j+1}$ by minimizing the following energy functions, respectively,

$$E(\bar{P}_{i,j}) = \sum_{l=i-1}^{i} \sum_{k=j-1}^{j} \int_{u_l}^{u_{l+1}} \int_{v_k}^{v_{k+1}} \left[\frac{\partial^2 P_{l,k}(u,v)}{\partial u^2} \right]^2 + 2\left[\frac{\partial^2 P_{l,k}(u,v)}{\partial u\,\partial v} \right]^2 + \left[\frac{\partial^2 P_{l,k}(u,v)}{\partial v^2} \right]^2 \mathrm{d}u\,\mathrm{d}v, \tag{27}$$

$$E(\bar{P}_{i,j}, \bar{P}_{i+1,j}) = \sum_{l=i-1}^{i+1} \sum_{k=j-1}^{j} \int_{u_l}^{u_{l+1}} \int_{v_k}^{v_{k+1}} \left[\frac{\partial^2 P_{l,k}(u,v)}{\partial u^2} \right]^2 + 2\left[\frac{\partial^2 P_{l,k}(u,v)}{\partial u\,\partial v} \right]^2 + \left[\frac{\partial^2 P_{l,k}(u,v)}{\partial v^2} \right]^2 \mathrm{d}u\,\mathrm{d}v, \tag{28}$$

$$E(\bar{P}_{i,j}, \bar{P}_{i+1,j}, \bar{P}_{i,j+1}, \bar{P}_{i+1,j+1}) = \sum_{l=i-1}^{i+1} \sum_{k=j-1}^{j+1} \int_{u_l}^{u_{l+1}} \int_{v_k}^{v_{k+1}} \left[\frac{\partial^2 P_{l,k}(u,v)}{\partial u^2} \right]^2 + 2\left[\frac{\partial^2 P_{l,k}(u,v)}{\partial u\,\partial v} \right]^2 + \left[\frac{\partial^2 P_{l,k}(u,v)}{\partial v^2} \right]^2 \mathrm{d}u\,\mathrm{d}v. \tag{29}$$

Our test results, however, show that these objective functions (27)–(29) do not provide better results than objective functions (18), (22) and (24), except making the algorithms more complicated and the computation process more expensive.

It should be pointed out that the above algorithms can be used to fair B-spline curves and surfaces as well. This is done by fairing the spline curve/surface that interpolates the control points of the B-spline curve/surface to be faired. This is based on the following observation. If the control points of a B-spline surface are properly taken from a surface, then the B-spline surface would be a good approximation of the surface. Thus if the surface is fair, its approximation B-spline surface is fair too.

5. Implementation

In this section, we compare the new method with Kjellander's method on two surfaces provided by the automobile industry. Highlight line models [1,2] of the surfaces corresponding to 20 linear light sources will be shown both before and after the fairing process. A highlight line model can detect very small surface normal (curvature) irregularities and, hence, is a good indicator of the smoothness of a surface. This sometimes is not possible with wire-frame drawings or shaded pictures [2,20].

The first case, a door panel with three irregular portions, is a bicubic spline surface defined by 13 × 13 data points. The original surface is shown in Fig. 1. The 'bad' data points of the surface, identified interactively, are faired by both the Kjellander method and the interactive fairing algorithm described in Section 4.4 repeatedly until the highlight line model of the surface remain unchanged visually. The Kjellander method adjusts only one point in each fairing step. The results after four fairing steps and eight fairing steps are shown in Figs. 2 and 3, respectively.

The top surface is the result produced by Kjellander's method and the bottom surface is the result produced by the new method. Fig. 4 is the results of the Kjellander method after 50 modification steps (top surface) and 100 modification steps (bottom surface). The new method generates a satisfactory result after only eight fairing steps (see Fig. 3), while the Kjellander method can not get a satisfactory result even after 100 fairing steps (see highlight line no. 7 from the right in Fig. 4).

The second case, a body part underneath the right head light of a car model, is a bicubic NURBS surface (actually, a B-spline surface since all the values of the weights are one), defined by 26×9 control points. This surface is frequently used to test if a fairing method is effective because the twisted nature of the surface makes it implicitly difficult to be faired. Highlight line model of the NURBS surface is shown in Fig. 5. The NURBS surface is faired by fairing the bicubic spline surface (called the *control point surface*) that interpolates the control points of the NURBS surface. The fairing process is performed as follows. First the 'bad' control points are identified by examining the abnormal portions of the bicubic NURBS surface. Then the 'bad' control points are faired by fairing the control point surface using both the Kjellander method and the interactive fairing algorithm described in Section 4.4 repeatedly until the highlight line model of the NURBS surface does not change any more visually. Results of the fairing process of both methods after fifteen, thirty and forty five iterations are shown in Figs. 6–8, respectively, with the left surface being the result of Kjellander's method and the right one being the result of the new method. The results show that for this case both methods can produce satisfactory result, but Kjellander's method requires more fairing steps.

6. Conclusions

Algorithms for interactive and automatic fairing of cubic spline curves and bicubic spline surfaces are presented. The 'bad' points of a cubic spline curve or bicubic spline surface are replaced with new ones generated by minimizing the strain energy of the corresponding segments or patches. The presented algorithms can also be used to fair cubic B-spline curves and bicubic B-spline surfaces. The abnormal portions of a B-spline curve/surface are removed by fairing the interpolant to its control points. In general, interactive methods are suitable for curve/surface construction, while the automatic methods are suitable for smoothing noised data points. Our test results show that the presented algorithms work quite effectively in removing the abnormal portions of cubic spline curves and bicubic spline surfaces.

Both the thin plate energy model and the thin strip energy model have been used to construct the objective functions for the fairing of bicubic spline and B-spline surfaces. Our test results show that these approaches have no advantages over each other in producing fair surfaces, except that the thin plate energy based method is more expensive in developing code and in computation.

The spline method is built on the assumption that the curvature of a spline curve can be approximated by its second derivative (subject to a constant factor). Our method is based on the same assumption and, hence, has the same limitation as the spline method. Our next work is to study the possibility of developing exact strain energy based spline curve/surface fairing techniques.

Acknowledgements

The consulting support of Drs Paul Stewart and Yifan Chen of the Ford Research Lab is deeply appreciated. We also thank the reviewers for several comments and suggestions which improved 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] Chen Y, Beier K-P, Papageorgiou D. Direct highlight line modification on nurbs surface. Computer Aided Geometric Design 1997;14:583–601.
[4] de Boor C. A Practical Guide to Splines. New York: Springer, 1978.
[5] Farin G, Rein G, Sapidis N, Worsey A. Fairing cubic B-spline curves. Computer Aided Geometric Design 1987;4(1/2):91–103.
[6] Faux ID, Pratt MJ. Computational Geometry for Design and Manufacture. New York: Wiley, 1979.
[7] Holladay JD. Smoothest curve interpolation. Math. Tables Aids Computation 1957;11:233–43.
[8] Kaufmann E, Klass R. Smoothing surface using reflection lines for families of splines. Computer-Aided Design 1988;20:312–6.
[9] Kjellander JAP. Smoothing of cubic parametric splines. Computer-Aided Design 1983;15(3):175–9.
[10] Kjellander JAP. Smoothing of bicubic parametric splines. Computer-Aided Design 1983;15(5):288–93.
[11] Klass R. Correction of local surface irregularities using reflection lines. Computer-Aided Design 1980;12:73–76.
[12] Lee ETY. Energy, fairness, and a counterexample. Computer-Aided Design 1990;22(1):37–40.
[13] Nowacki H. Curve and surface generation and fairing. In: Zobrist GW, Sabharwal C, editors. Progress in Computer Graphics, vol. 1. Norwood, NJ: Ablex, 1992. p. 137–76.
[14] Poeschl T. Detecting surface irregularities using isophotes. Computer Aided Geometric Design 1984;1:163–8.
[15] Poliakoff JF. An improved algorithm for automatic fairing of non-uniform parametric cubic splines. Computer-Aided Design 1996;28:59–66.
[16] Poliakoff JF, Wong YK, Thomas PD. An automatic curve fairing

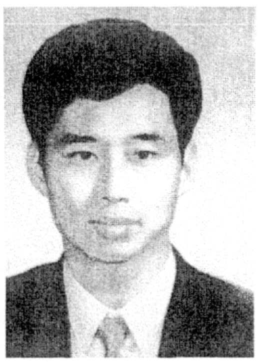
algorithm for cubic B-spline curves. Journal of Computational and Applied Mathematics 1999;102:73–85.

[17] Sapidis N. Towards automatic shape improvement of curves and surfaces for computer graphics and CAD/CAM applications. In: Zobrist GW, Sabharwal C, editors. Progress in Computer Graphics, vol. 1. Norwood, NJ: Ablex, 1992. p. 216–53.

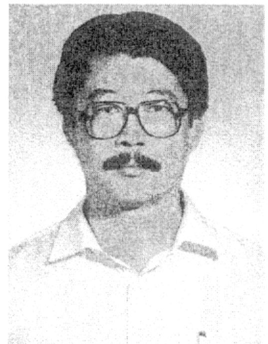
[18] Sapidis N, Farin G. Automatic fairing algorithm for B-spline curves. Computer-Aided Design 1990;22(2):121–9.

[19] Wang X, Cheng F. Surface design based on hermite interpolation with tension control and optimal twist vectors. Neural, Parallel and Scientific Computations 1997;5(1/2):37–57 (Special Issue in Computer Aided Geometric Design).

[20] Zhang C, Cheng F. Removing local irregularities of NURBS surface by modifying highlight lines. Computer-Aided Design 1998; 30:923–30.



Caiming Zhang is a postdoctoral 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 doctorate degree in Engineering from 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.



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 MS 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.

Pifu Zhang is a postdoctoral fellow in the Graphics and Geometric Modelling Lab at the University of Kentucky. He received his BEngng and ME in mechanical engineering from Hunan Agricultural University and Beijing Agricultural Engineering University in 1983 and 1987, respectively, and a doctorate degree in Engineering from Hunan University, China, in 1998. His research interests include computer-aided geometric design and modeling, computer graphics, visualization, and mesh generation.