Shape Improvement of Surfaces

Stefanie Hahmann[*]

Preprint: to be published in Journal of Computing 1998

Abstract. *An automatic and local fairing algorithm for bicubic B-spline surfaces is proposed. A local fairness criterion selects the knot, where the spline surface has to be faired. A fairing step is than applied, which locally modifies the control net by a constrained least-squares approximation. It consists of increasing locally the smoothness of the surface from C^2 to C^3. Some extensions of this method are also presented, which show how to build further methods by the same basic fairing principle.*

---

1. INTRODUCTION

The problem of *fairness* is of central importance during the design process of free form surfaces. Interpolating or approximating some data sets, the surface scheme has to reflect the geometric shape implied by the data set and should produce *visual pleasing* surfaces. This means that fairing is a post-processing step which applies to a given surface.

Another technique for producing smooth surfaces is known as *variational design*. The fairness criterion is here already incorporated in the design process. An optimization process uses the degrees of freedom of the surface geometry in order to minimize some energy integral.

There is no unique mathematical criterion measuring the fairness of a given surface because it is the designers subjective decision to accept a surface as fair enough or not. Nevertheless, most existing fairing methods for curves and surfaces deal with physically based fairness criteria, like minimal energy. They proved to be excellent criteria for producing smooth surfaces. For some early papers using these criteria see [22, 24, 25, 11]. Other criteria which are based on surface areas were proposed by [27]. Aesthetic criteria based

[*] Laboratoire LMC-IMAG, University of Grenoble, B.P. 53, F-38041 Grenoble cedex 9, France, Tel: +33 4 76 63 57 88, Fax +33 4 76 63 12 63, Stefanie.Hahmann@imag.fr

on the light reflection behaviour of a surface are the best in translating the intuitive notion of fairness that the designer has. Special light lines, like reflection lines [19], isophotes [26] or highlight lines [1] are drawn on the surface and visualize shape imperfections [14].

The following fairing method applies to bicubic tensor product B-spline surfaces. Those are often used in geometric modeling due to their well known advantages which result basicly from the local support of the basis splines and their dependence on the knots. To start the description of the present method, Section 2 first recalls two related works, which influenced the fairing method presented then in Section 3 and 4. In addition to this, Section 5 gives some alternatives to the algorithm, i.e. it shows how to construct further fairing methods which are always based on the same fairing idea. This paper finally presents some results and ends with practical considerations.

2. RELATED WORK

Plenty of papers in the CAGD literature deal with fairing curves and surfaces. A fair surface can be obtained by two different ways. The first one consists of **modeling surfaces with fairness constraints**: A linearized physical based fairness criterion (i.e. minimal bending energy of torsion and flexure, minimal jerk, etc) is incorporated in the interpolation or approximation method. For more details see e.g. [12, 13, 3]. Non-linear methods for constrained surface modeling can be found in [20, 8, 23, 9]. The other way to obtain fair surfaces is to apply a **post-processing fairing** method to a given surface [18, 4, 10, 15]. Due to digitization errors, surface imperfections can occur and need to be faired. We will use isophotes [26] as tool for visualizing those unwanted surface wiggles.

The fairing method presented in Section 3, will be at the time an *automatic, local* and *efficient* method. The execution time of the algorithm should be reasonably short. Even for objects with hundreds or more control points this is the crucial point for most existing methods. Locality is also very important in the sense that a surface which needs to be faired only in a small region doesn't have to change its shape everywhere. The basic fairing criterion we use, aims to decrease the sum of the C^3-discontinuities at the junction points of the surface patches. Two curve fairing methods mainly influenced the present work.

Kjellander's method for cubic Ferguson splines

Each segment of a cubic Ferguson spline x is uniquely determined by Hermite interpolating the positions $p_i = x(t_i)$, $p_{i+1} = x(t_{i+1})$, and the tangent input $x'(t_i)$, $x'(t_{i+1})$ at its end points x_i, x_{i+1}. Kjellander [17] states that the energy of a physical spline is decreased by decreasing shear forces applied to it. This is achieved by decreasing the difference in the third derivatives on both sides of a data point. A specified data point p_k is therefore moved into a "better" position in order to make the difference equal zero. Recalculation of the tangent input is necessary to stay in the class of C^2 curves, but unfortunately, this leads to a global algorithm. Nevertheless, each iteration usually gives a fairer curve.

A generalization to piecewise bicubic tensor product (Hermite) surfaces is also due to Kjellander [18]. Decreasing the sum of the difference in the third derivatives of the two space curves at the junction point of four patches by moving it into a better position decreases the physical spline energy. Here again, it is a *global fairing method.*

Farin et al.'s method for cubic B-spline curves

Based on Kjellander's fairness criterion Farin et al. [7] propose a *local algorithm* where in each fairing step only a small number of control points is involved. They use a *knot-removal-reinsertion* step at the most offending inner knot t^*. The B-spline curve becomes momentarily C^3-continuous at t^*. The same procedure is than applied to another knot. The iteration continues while the global fairness measure (sum of third discontinuities at all inner knots) decreases. Knot removal [21, 28, 6] is in general not the inverse operation of knot insertion [2]. The resulting curve is therefore an approximation which differs from the original one in at least one control point.

The idea of using knot removal for decreasing the sum of C^3-discontinuities over the curve is difficult to generalize to surfaces. Knot removal of surface knots doesn't make the surface C^3-continuous at those points. Furthermore, the fairing step is not local any longer because whole rows (or columns) of control points are involved when removing one knot of the surface.

The fairing method we propose, although based on the ideas behind Kjellander and Farin's methods, consists of a fairing step which is better appropriated for surfaces and has the desired quality to be local. The basic fairing principle will be described in the next sections. Later, some extensions will be presented in Section 5.

3. THE BASIC FAIRING METHOD

Some notations and properties of bicubic B-spline surfaces that will be useful for the next chapters are briefly reviewed first. Given the positive integers $n, m \geq 3$ and $\boldsymbol{u} = (u_i)_{i=0}^{n+4}$, $\boldsymbol{v} = (v_j)_{j=0}^{m+4}$ two sequences of real numbers with $u_i < u_{i+4}$ and $v_j < v_{j+4}$ $(i, j = 0, \ldots, n, m)$, the B-spline basis functions associated with the *knot vector* $\boldsymbol{u}$ are denoted by $(N_{i,4,u})_{i=0}^{n}$ (or simply by $N_{i,4}$) and are assumed to be normalized to sum to one. The same holds for $(N_{j,4,v})_{j=0}^{m}$. A parametric *tensor product B-spline surface* $\boldsymbol{X}$ in $\mathbb{R}^3$ of order $(4,4)$ is then defined by

$$\boldsymbol{X}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \boldsymbol{d}_{ij} N_{i,4}(u) N_{j,4}(v)\,, \quad (u,v) \in \Omega := [u_3, u_{n+1}] \times [v_3, v_{m+1}] \qquad (1)$$

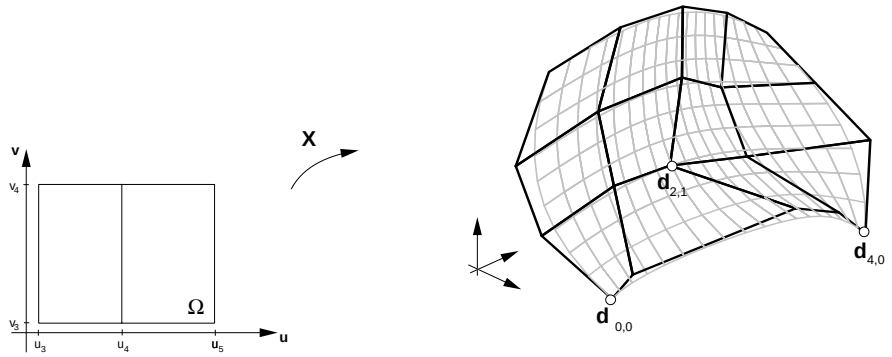

Figure 1: *illustration of the bicubic B-spline notations for $n = 4$, $m = 3$*

where the coefficients $\boldsymbol{d}_{ij} \in \mathbb{R}^3$ form the control net, Figure 1. The knots $u_4, \ldots, u_n$, $v_4, \ldots, v_m$ are called *interior knots* and are supposed to be of multiplicity one for the following considerations. Let $I \subset \mathbb{Z}^2$ be the index set for the knot pairs (u_k, v_l) of the interior knots, $(k,l) \in I = \{(4,4),(4,5),\ldots,\ldots,(n,m)\}$. Let us call those knots *free inner knots*, because they will be possible candidates for the fairing process. A bicubic B-spline surface is a composite of $(n-2)(m-2)$ polynomial surface patches of bi-degree 3 which are joining each other at least C^2-continuously at the knots. It is important to notice that the *local support* property of the B-spline basis functions implies that each control point has only a local influence: moving one control point $\boldsymbol{d}_{ij}$ changes the surface only locally for all $(u,v) \in [u_i, u_{i+4}] \times [v_j, v_{j+4}]$. For more details about B-splines, see [5, 29].

3.1 The fairing principle

For the development of the fairing method we need first some quantitative measures of fairness. The fairing principle refers to Kjellander's beam model (Section 2) for surfaces [18]. In terms of bicubic B-spline surfaces it can be resumed as follows:

Fairing principle 1: *A B-spline surface $\boldsymbol{X}(u,v)$ of class C^2 is fairer at the knot (u_k, v_l), $(k,l) \in I$, if $\boldsymbol{X}$ is C^3 at (u_k, v_l).*

This is a local criterion which applies only at one knot pair (u_k, v_l). A local fairing step consists therefore in decreasing the sum of differences in third partial derivatives at the knot (u_k, v_l). It implies also that fairing the whole surface means to decrease the sum of its third order discontinuities at all interior knots of the surface. The formulas for those local and global fairness measures will now be developed.

A surface $\boldsymbol{X}$ is C^3 at (u,v) if and only if all third order partial derivatives of $\boldsymbol{X}$ at (u,v) are continuous. Let $\boldsymbol{X}_{u^\nu v^\mu} := \frac{\partial^{\nu+\mu} X}{\partial u^\nu \partial v^\mu}$ with $(\nu + \mu = 3,\ \nu, \mu \in \mathbb{N}_0)$ denote the third order partial derivatives of $\boldsymbol{X}$. Bicubic B-spline surfaces have the pleasant property, that all mixed third order partial derivatives (i.e. $\boldsymbol{X}_{u^\nu v^\mu}$ with $\nu, \mu \geq 1$) are continuous on Ω.

$$\boldsymbol{X}_{u^\nu v^\mu}(u,v) = \nu!\mu! \sum_{i=\nu}^{n} \sum_{j=\mu}^{m} \boldsymbol{d}_{ij}^{(\nu,\mu)} N_{i,4-\nu}(u) N_{j,4-\mu}(v)\ , \tag{2}$$

where $\boldsymbol{d}_{ij}^{(\nu,\mu)}$ is a linear combination of the control points $\boldsymbol{d}_{rs}$ $(i-\nu \le r \le i,\ j-\mu \le s \le j)$. For details see [16]. This is due to the definition of the basis functions $N_{i,4-\nu}$ and $N_{j,4-\mu}$ resp., which are at least C^0 on $[u_3, u_{n+1}]$ and $[v_3, v_{m+1}]$ resp., if $\nu, \mu \ge 1$ and $\nu + \mu = 3$ $(\Rightarrow\ \nu \le 2$ and $\mu \le 2)$.

The sum of the difference of $\boldsymbol{X}_{uuu}$ in u-direction and $\boldsymbol{X}_{vvv}$ in v-direction at the interior knot (u_k, v_l), $(k,l) \in I$, is therefore the appropriate local fairness measure according to the fairing principle 1. For simplification let us introduce the following *discontinuity vectors*:

$$\begin{aligned}
\Delta_{uuu}(u_k, v_l) :&= \boldsymbol{X}_{uuu}(u_k^-, v_l) - \boldsymbol{X}_{uuu}(u_k^+, v_l) \\
&= \sum_{j=0}^{m} \boldsymbol{d}_{k-1,j}^{(3,0)} N_{j,4}(v_l) - \sum_{j=0}^{m} \boldsymbol{d}_{k,j}^{(3,0)} N_{j,4}(v_l) \\
&=: \sum_{i=k-4}^{k} \sum_{j=l-3}^{l-1} \alpha_{ij} \boldsymbol{d}_{ij}
\end{aligned} \tag{3}$$

$$\begin{aligned}
\Delta_{vvv}(u_k, v_l) :&= \boldsymbol{X}_{vvv}(u_k, v_l^-) - \boldsymbol{X}_{vvv}(u_k, v_l^+) \\
&= \sum_{i=0}^{n} \boldsymbol{d}_{i,l-1}^{(0,3)} N_{i,4}(u_k) - \sum_{i=0}^{n} \boldsymbol{d}_{i,l}^{(0,3)} N_{i,4}(u_k) \\
&=: \sum_{i=k-3}^{k-1} \sum_{j=l-4}^{l} \beta_{ij} \boldsymbol{d}_{ij}
\end{aligned}$$

with

$$\alpha_{ij} = \alpha_{ij}(u_k, v_l) \quad \text{and} \quad \beta_{ij} = \beta_{ij}(u_k, v_l)\,.$$

In the case of equidistant interior knots $(u_{i+1} - u_i = v_{j+1} - v_j = 1$ for $i, j = 3, \ldots, n, m)$ the value of the coefficients α_{ij}, β_{ij} of Δ_{uuu} and Δ_{vvv} at (u_k, v_l) are shown schematically in Figure 2 (for example: $\alpha_{k-4,l-1} = -1/6$, $\alpha_{k-3,l-1} = 2/3$, $\alpha_{k-2,l-1} = -1$, and so on). The bullets $\bullet$ in the regular mesh in Figure 2 indicate the control points $\boldsymbol{d}_{ij}$, which are involved in the calculation of $\Delta_{uuu}(u_k, v_l)$ and $\Delta_{vvv}(u_k, v_l)$.

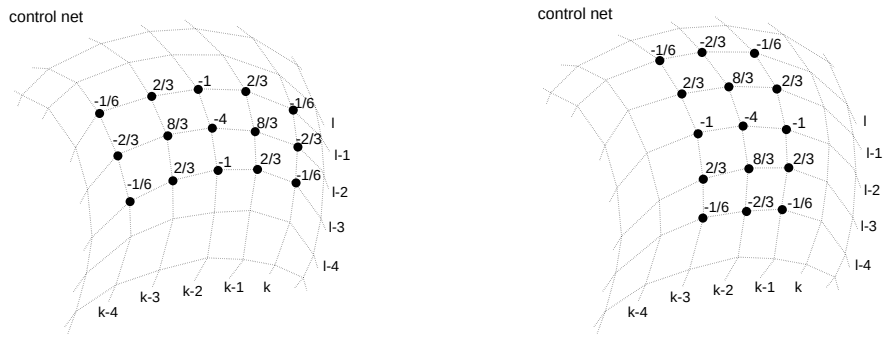

Figure 2: *coefficients and control points involved in the calculation of $\Delta_{uuu}(u_k, v_l)$ (left) and $\Delta_{vvv}(u_k, v_l)$ (right) for uniform knots*

The *local fairness measure* L_{kl} at the point (u_k, v_l) is defined as

$$L_{kl} := \|\Delta_{uuu}(u_k, v_l)\|^2 + \|\Delta_{vvv}(u_k, v_l)\|^2 \ , \quad (k,l) \in I \ . \tag{4}$$

The 21 control points, which determine L_{kl} are shown schematically in Figure 3.

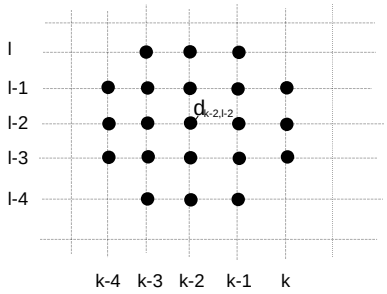

Figure 3: *Control points involved in the calculation of* L_{kl}

The whole surface $\boldsymbol{X}$ can now be associated with the quantity

$$G_X := \sum_{(k,l)\in I} L_{kl} \ , \tag{5}$$

which implies that G_X characterizes the *global fairness measure.*

Fairing principle 2: A surface $\boldsymbol{X}$ is fairer than $\boldsymbol{Y}$ if $G_X < G_Y$.

3.2 The local fairing step

The strategy we adopt now in order to find a surface which decreases the fairness measure G_X includes two steps:

(a) perform a fairing step at the knot (u_k, v_l), where $(k,l) \in I \ : \ L_{kl} = \max_{ij} L_{ij}$,
(b) update G_X and goto (a).

The smoothness of the surface increases at (u_k, v_l) from C^2 to C^3 ($L_{kl} = 0$), i.e. the third order discontinuity vanishes at (u_k, v_l). The surface has now the highest possible continuity at (u_k, v_l). Step (b) is straightforward to calculate with formula (5). Step (a) needs some more explications. It contains the main step of the fairing process. We will refer to it as *local fairing step.* After a fairing step at the knot (u_k, v_l) the local fairness measure L_{kl} has to be equal zero, which is equivalent to the two conditions:

$$\Delta_{uuu}(u_k, v_l) = 0 \quad \text{and} \quad \Delta_{vvv}(u_k, v_l) = 0 \ . \tag{6}$$

21 (unknown) control points are involved in these two equations (see (3) and Figure 3). There exists an infinite number of solutions to this problem. It is easy to understand that the resulting surface $\hat{\boldsymbol{X}}$ should deform the given surface $\boldsymbol{X}$ as less as possible, i.e. the maximal distance between the two surfaces should be minimal. In terms of the L^∞-norm

$\|\boldsymbol{X}-\hat{\boldsymbol{X}}\|_\infty \to \min$, this leads to a non-linear problem, which is not appropriate for a fairing procedure. In order to get a fairing step which is as local as possible without restricting the solution too much, we proceed as illustrated in Figure 4. The 12 'furthest' control points (denoted by $\circ$ in Figure 4) are kept fixed. We then use a classical least-squares approximation with 2 linear constraints on the 9 remaining control points (denoted by $\bullet$ in Figure 4).

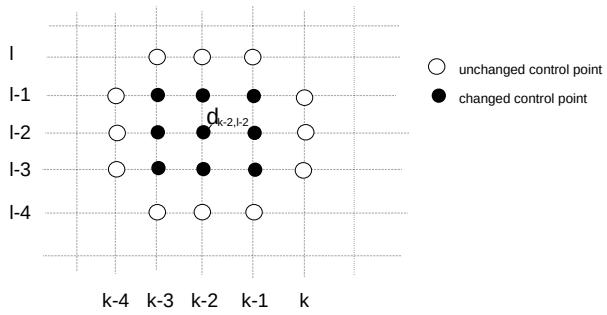

Figure 4: *The control points which are involved in the constrained least-squares approximation (7)*

The solution therefore consists of the 9 new control points $\hat{\boldsymbol{d}}_{ij}$ resulting from

$$\text{Minimize} \quad F(\hat{\boldsymbol{d}}_{ij}) = \sum_{i=k-3}^{k-1} \sum_{j=l-3}^{l-1} \|\boldsymbol{d}_{ij} - \hat{\boldsymbol{d}}_{ij}\|^2 \quad \text{subject to} \quad (6)\ . \tag{7}$$

The technique of Lagrange multipliers [30]

$$\Phi(\hat{\boldsymbol{d}}_{ij}, \lambda, \mu) = F(\hat{\boldsymbol{d}}_{ij}) + \lambda(\Delta_{uuu}(u_k, v_l)) + \mu(\Delta_{vvv}(u_k, v_l)) \to \min \tag{8}$$

leads to a (11,11)-linear system of equations

$$\begin{aligned} \frac{\partial \Phi}{\partial \hat{d}_{ij}} &= 0 \qquad 9 \text{ equations} \\ \frac{\partial \Phi}{\partial \lambda} &= 0 \qquad 1 \text{ equation} \\ \frac{\partial \Phi}{\partial \mu} &= 0 \qquad 1 \text{ equation.} \end{aligned} \tag{9}$$

To summarize, one (local) fairing step at the knot (u_k, v_l) consists of solving the system of linear equations (9) in order to get a new surface $\hat{\boldsymbol{X}}$ with $L_{kl} = 0$, i.e. a surface which is C^3 at (u_k, v_l) after one fairing step.

Remark: In the case of equidistant knots, the matrix of system (9) is independent of the current knot under consideration. If $u_{i+1} - u_i = v_{j+1} - v_j = 1$, then the system is given

by

$$\begin{bmatrix}
2 & & & & & & & & & 4 & 4 \\
 & 2 & & & & & & & & -6 & 16 \\
 & & 2 & & & 0 & & & & 4 & 4 \\
 & & & 2 & & & & & & 16 & -6 \\
 & & & & 2 & & & & & -24 & -24 \\
 & & 0 & & & 2 & & & & 16 & -6 \\
 & & & & & & 2 & & & 4 & 4 \\
 & & & & & & & 2 & & -6 & 16 \\
 & & & & & & & & 2 & 4 & 4 \\
4 & -6 & 4 & 16 & -24 & 16 & 4 & -6 & 4 & 0 & 0 \\
4 & 16 & 4 & -6 & -24 & -6 & 4 & 16 & 4 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\hat{\boldsymbol{d}}_{k-3,l-1} \\ \hat{\boldsymbol{d}}_{k-2,l-1} \\ \hat{\boldsymbol{d}}_{k-1,l-1} \\ \hat{\boldsymbol{d}}_{k-3,l-2} \\ \hat{\boldsymbol{d}}_{k-2,l-2} \\ \hat{\boldsymbol{d}}_{k-1,l-2} \\ \hat{\boldsymbol{d}}_{k-3,l-3} \\ \hat{\boldsymbol{d}}_{k-2,l-3} \\ \hat{\boldsymbol{d}}_{k-1,l-3} \\ \lambda \\ \mu
\end{bmatrix}
=
\begin{bmatrix}
2\boldsymbol{d}_{k-3,l-1} \\ 2\boldsymbol{d}_{k-2,l-1} \\ 2\boldsymbol{d}_{k-1,l-1} \\ 2\boldsymbol{d}_{k-3,l-2} \\ 2\boldsymbol{d}_{k-2,l-2} \\ 2\boldsymbol{d}_{k-1,l-2} \\ 2\boldsymbol{d}_{k-3,l-3} \\ 2\boldsymbol{d}_{k-2,l-3} \\ 2\boldsymbol{d}_{k-1,l-3} \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2
\end{bmatrix},$$

where

$$\begin{aligned}
\boldsymbol{c}_1 &= (\boldsymbol{d}_{k-4,l-3} + 4\boldsymbol{d}_{k-4,l-2} + \boldsymbol{d}_{k-4,l-1} + \boldsymbol{d}_{k,l-3} + 4\boldsymbol{d}_{k,l-2} + \boldsymbol{d}_{k,l-1})\,, \\
\boldsymbol{c}_2 &= (\boldsymbol{d}_{k-3,l-4} + 4\boldsymbol{d}_{k-2,l-4} + \boldsymbol{d}_{k-1,l-4} + \boldsymbol{d}_{k-3,l} + 4\boldsymbol{d}_{k-2,l} + \boldsymbol{d}_{k-1,l})\,.
\end{aligned}$$

Once the inverse of this matrix is calculated, each local fairing step remains to perform one matrix-vector-multiplication in order to get the new 9 control points $\hat{\boldsymbol{d}}_{ij}$. Notice, that in the case of uniform B-splines, the constrained least-squares fitting has to be solved only once for the first fairing step. All the following steps reduce to a simple matrix-vector-multiplication, which is equivalent to an *explicit solution* for the local fairing step.

Remains to answer the *choice of the knot pair* (u_k, v_l) for the local fairing step. It is very natural to fair the surface at the knot corresponding to the largest third order discontinuity, because we can expect there the biggest fairing effect.

4. FAIRING B-SPLINE SURFACES

The fairness indicators, the local fairing step and the automatic choice of the offending knot result in a completely automatic and local fairing algorithm, called

basic fairing algorithm

1. Calculate initial global measure G^0: (5)
2. `WHILE` (end condition = `false`) `DO`
   (a) choose (u_k, v_l), where $(k,l) \in I \;:\; L_{kl} = \max_{ij} L_{ij}$: (4)
   (b) apply local fairing step at (u_k, v_l): (7)-(9)
   (c) calculate new global measure G^j

   `END`.

A good condition to stop the algorithm is, when there is no more considerable improvement of the global measure: $|G^{j+1} - G^j| < \varepsilon$. Another possibility is to stop the algorithm, when

G^j increases for the first time. We think it is not a good choice, because it can stop the iterations to early.

The following examples demonstrate the capacity and the limits of the proposed fairing method. The isophote technique [26] is used as visual fairness indicator. Isophotes are very sensitive to small surface imperfections. A fair surface is characterized by well behaved isophotes. An Isophote is defined as a line of constant angle between the unit normal vector $\boldsymbol{N}$ and a given light direction vector $\boldsymbol{L}$:

$$\boldsymbol{N}(u,v) \cdot \boldsymbol{L} = c = const\ , \quad c \in I\!R\ . \tag{10}$$

It is a first order interrogation tool, which is not reflecting the well behaved curvature. Attention should be payed to the light direction which has to be chosen. Each light direction gives different isophotes on the surface. Before qualifying a surface as fair enough different isophotes should be tested.

Example 1:

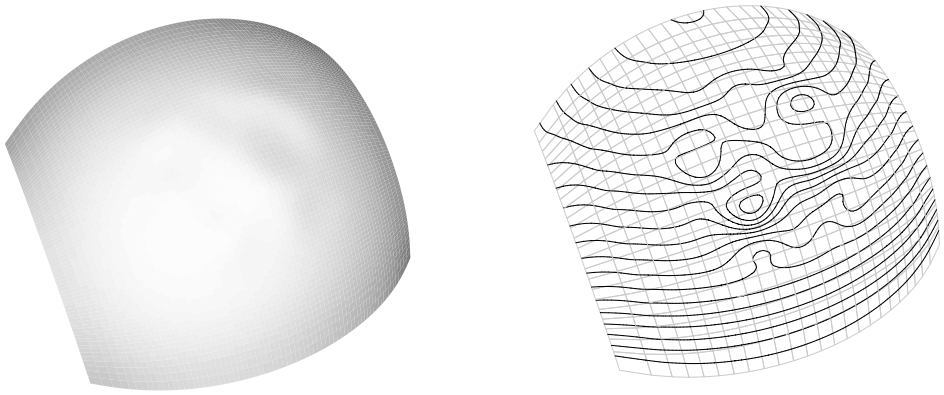

Figure 5: *Unfair surface with 225 control points lying nearby a part of the unit sphere*

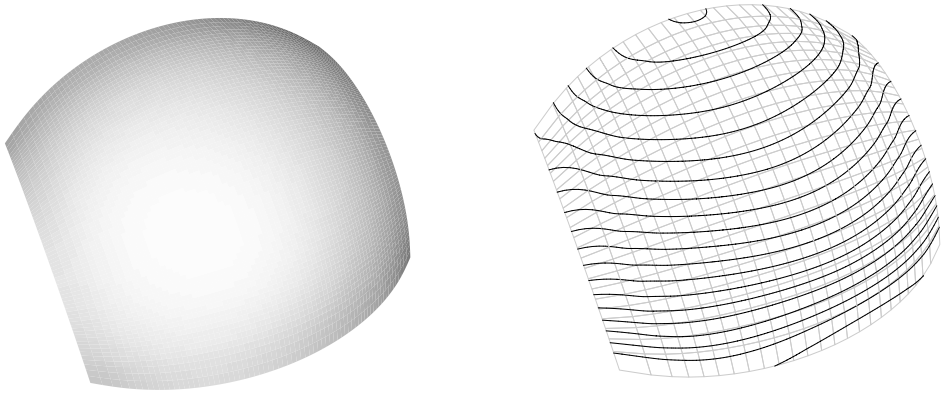

Figure 6: *Faired surface after 500 iterations*

The first example is a surface where the control points were originally taken from a part of the *unit sphere* and than some inner control points were slightly perturbed. The isophote analysis of this surface (Figure 5) reflects the unwanted surface wiggles. The surface has

15×15 control points, i.e. 11×11 interior knots and is uniformly parametrized. The fairing algorithm with only 500 iterations decreases the global fairness measure from $G_{X_{init}} = 0.78$ to $G_{X_{fair}} 0.0001$. The *maximum relative distance* between the initial (before fairing) and new control points (after fairing), $\tau = \max \|\boldsymbol{d}_{ij} - \hat{\boldsymbol{d}_{ij}}\| / \max \|\boldsymbol{d}_{ij} - \boldsymbol{d}_{kl}\|$, amounts to $\tau = 0.01$ (mean relative distance: 0.001). The execution time of the algorithm for 500 iterations believes only to 0.3 s (SGI Octane). This is due to the uniform parametrization. The matrix of the constrained least-squares approximation (9) doesn't change during the algorithm. Therefore this matrix has to be inversed only once in the beginning. Each fairing step consists then only in a matrix-vector multiplication. The smoothness of the resulting surface is very satisfying for this example, see Figure 6.

Example 2: Also the second example shows the efficiency of the fairing method. The unfair surface has 20×20 control points, i.e. 16×16 interior knots and a uniform parametrization, see Figure 7.

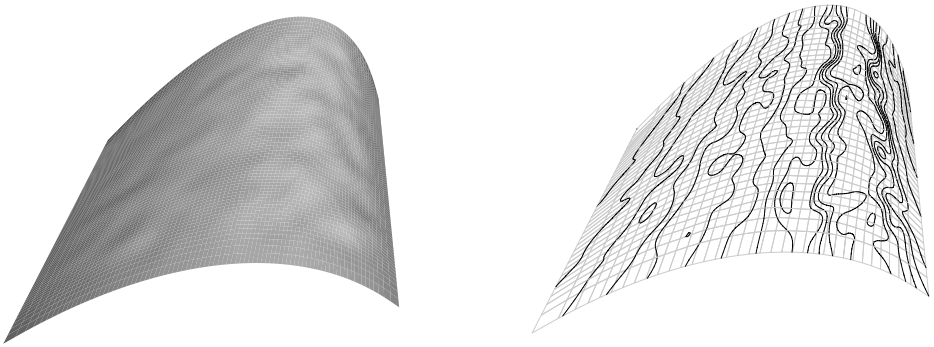

Figure 7: *Unfair surface with 400 control points*

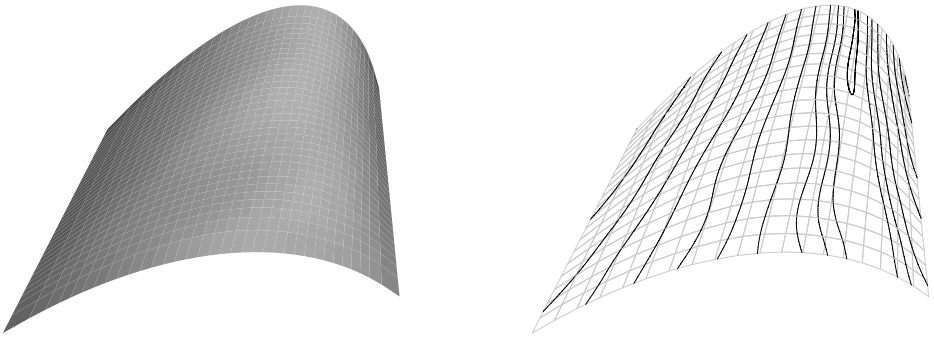

Figure 8: *Faired surface after 6000 iterations*

6000 iteration (running time: 7 s) are needed to decrease the global fairness measure from $G_{X_{init}} = 108$ to $G_{X_{fair}} = 0.03$. The result is shown in Figure 8. The maximum *relative distance* between the initial and new control points is equal to $\tau = 0.006$, i.e. 0.6% (mean relative distance: 0.001).

Example 3:

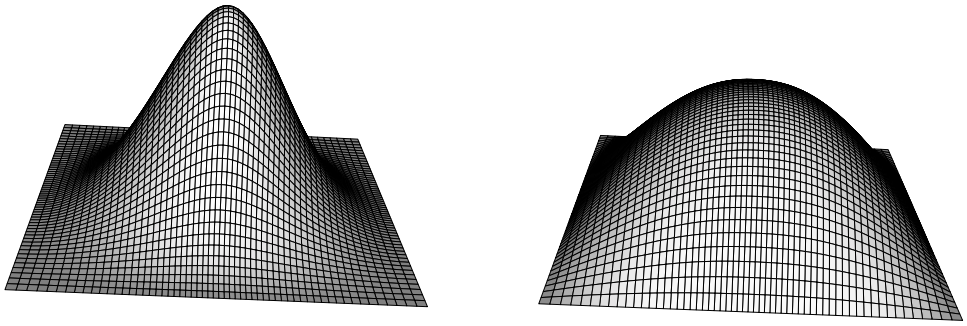

Figure 9: *The local fairing effect for one iteration*

The local fairing effect of one fairing step is illustrated in Figure 9. The given surface (left) has only one free interior knot (i.e. 5×5 control points). All control points are coplanar except one in the middle of the surface. After one iteration the border control points keep unchanged and the peak is flattened out.

5. EXTENSIONS TO THE FAIRING METHOD

5.1 1-point fairing step

The fairing step (7) of the basic algorithm consists of increasing the smoothness of a given surface at one point $\boldsymbol{X}(u_k, v_l)$ from C^2 to C^3. So let us call it *one-point fairing step*. 9 control points are involved in a constrained least-squares fitting. This can be seen as a **mask**, which is applied to the 9 control points (see Figures 4 and 10):

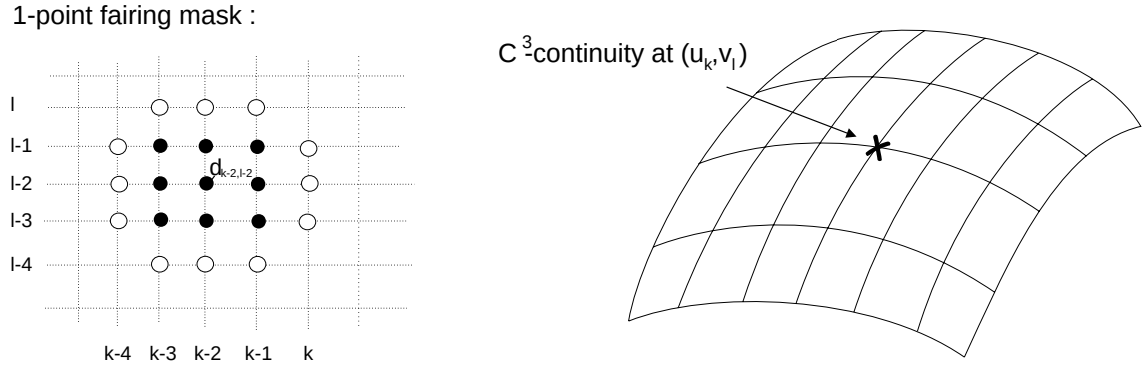

Figure 10: *1-point fairing mask*

Let us now introduce other fairing steps which enforce C^3-continuity either

- at several points (patch corners) at the same time

and / or

- between adjacent surface patches.

5.2 4-point fairing step

Applying the 1-point fairing step to four neighboring points *simultaneously* (see Figure 11) means to enforce C^3-continuity at 4 points (patch corners) simultaneously, i.e.

$$\Delta_{uuu}(u_r, v_s) = 0 \quad \text{and} \quad \Delta_{vvv}(u_r, v_s) = 0 \quad \text{for} \quad \begin{cases} r = k, k+1 \\ s = l, l+1 \end{cases} . \tag{11}$$

Using a constrained least-squares approximation, the corresponding fairing step states as follows:

$$\text{Minimize} \quad F(\hat{\boldsymbol{d}}_{ij}) = \sum_{i=k-3}^{k} \sum_{j=l-3}^{l} \|\boldsymbol{d}_{ij} - \hat{\boldsymbol{d}}_{ij}\|^2 \quad \text{subject to} \quad (11) , \tag{12}$$

where $\hat{\boldsymbol{d}}_{ij}$ $(i = k-3, \ldots, k;\ j = l-3, \ldots, l-3)$ are the 16 control points and (11) are the 8 linear constraints.

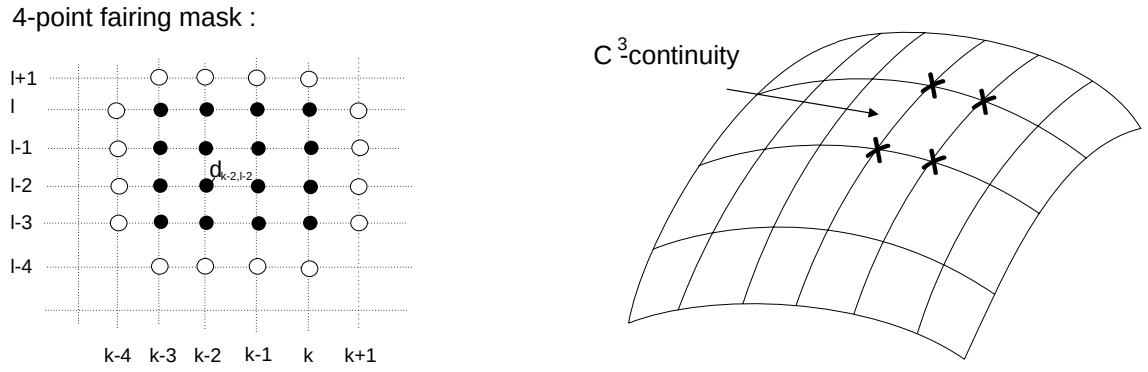

Figure 11: *4-point fairing mask*

5.3 1-segment fairing step

Another type of fairing mask is the extension of the fairing step to a curve segment. The idea is to increase the smoothness of $\boldsymbol{X}$ between two adjacent patches, i.e. these two patches will joint together $C^3(C^\infty)$-continuously across their common boundary curve. This common boundary is determined by the parameter interval $[u_{k-1}, u_k]$ for a fixed parameter v_l. Here again the C^3 condition is incorporated in a constrained least-squares approximation. It can be illustrated as follows:

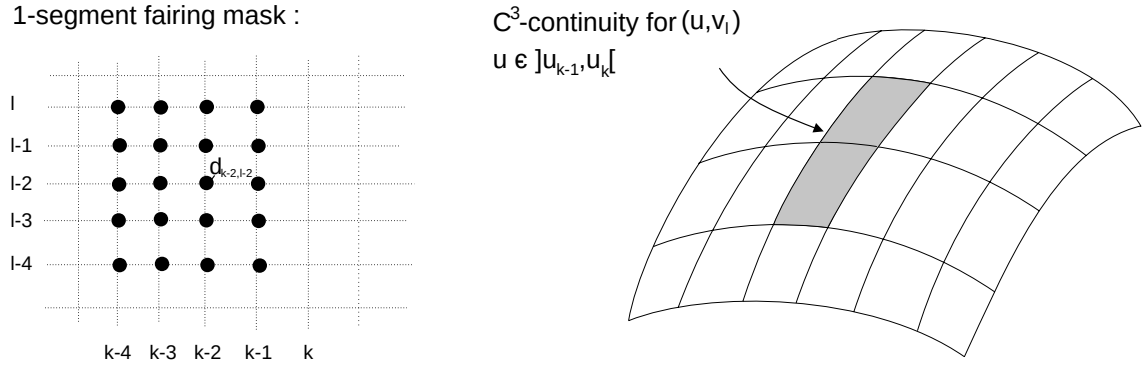

Figure 12: *1-segment fairing mask*

The third order partial derivatives (v-direction) left and right of the common curve segment define two cubic curve segments:

$$\begin{aligned} \boldsymbol{X}_{vvv}(u, v_l^-) &= \sum_{i=0}^{n} \boldsymbol{d}_{i,l-1}^{(0,3)} N_{i,4}(u) \qquad u \in [u_{k-1}, u_k[\,, \\ \boldsymbol{X}_{vvv}(u, v_l^+) &= \sum_{i=0}^{n} \boldsymbol{d}_{i,l}^{(0,3)} N_{i,4}(u) \qquad u \in [u_{k-1}, u_k[\,. \end{aligned} \tag{13}$$

Each of them is determined by 4 control points: $\boldsymbol{d}_{i,l-1}^{(0,3)}$ for $\boldsymbol{X}_{vvv}(u, v_l^-)$ and $\boldsymbol{d}_{i,l}^{(0,3)}$ for $\boldsymbol{X}_{vvv}(u, v_l^+)$, where $(i = k-4, \ldots, k-1)$. These points need to be identical, i.e.

$$\boldsymbol{d}_{i,l-1}^{(0,3)} = \boldsymbol{d}_{i,l}^{(0,3)} \qquad i = k-4, \ldots, k-1\,, \tag{14}$$

if one wants $\boldsymbol{X}_{vvv}$ to be continuous for (u, v_l), $u \in [u_{k-1}, u_k[$. As C^3-condition we have therefore the 4 equations (14). These 4 constraints together with a least-squares approximation on the 20 control points $\hat{\boldsymbol{d}}_{ij}$ $(i = k-4, \ldots, k-1, j = l-4, \ldots, l)$ which are involved in equations (14), build the so-called *1-segment fairing step*.

$$\text{Minimize} \quad F(\hat{\boldsymbol{d}}_{ij}) = \sum_{i=k-4}^{k-1} \sum_{j=l-4}^{l} \|\boldsymbol{d}_{ij} - \hat{\boldsymbol{d}}_{ij}\|^2 \quad \text{subject to} \quad (14)\,. \tag{15}$$

Its schematical description is shown in Figure 12.

5.4 4-segment fairing step

Now it is straightforward to extend this basic mask to more than one segment, e.g. four segments. In this case four patches will join together C^3 (C^∞)-continuously in order to form one polynomial surface patch. What basically happens can be seen in Figure 13.

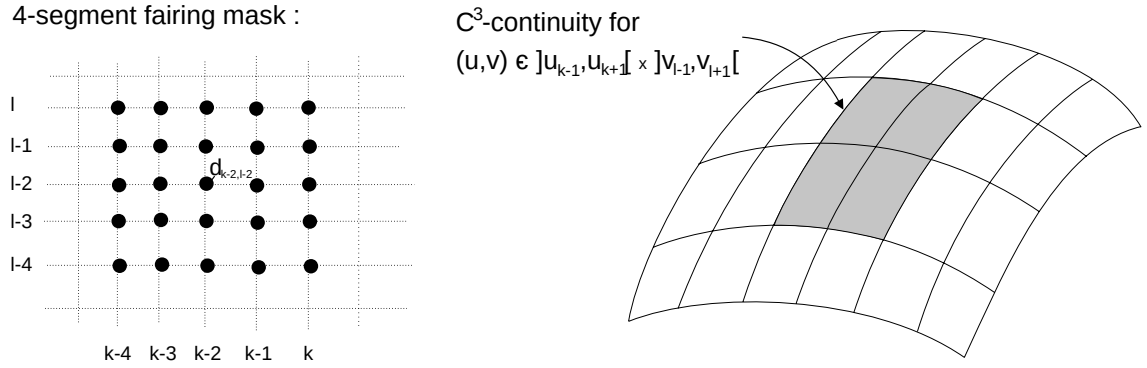

Figure 13: *4-segment fairing mask*

It is a right and up shift of the basic 1-segment mask. The two crossing curve segments correspond to the parameter intervals $[v_{l-1}, v_{l+1}[$ for a fixed parameter u_k and $[u_{k-1}, u_{k+1}[$

for a fixed parameter v_l. The left and right third order partial derivatives are given by:

$$\begin{aligned}
\boldsymbol{X}^-_{uuu} &:= \boldsymbol{X}_{uuu}(u_k^-, v) = \sum_{j=0}^{m} \boldsymbol{d}^{(3,0)}_{k-1,j} N_{j,4}(v) \qquad v \in [v_{l-1}, v_{l+1}[\\
\boldsymbol{X}^+_{uuu} &:= \boldsymbol{X}_{uuu}(u_k^+, v) = \sum_{j=0}^{m} \boldsymbol{d}^{(3,0)}_{k,j} N_{j,4}(v) \qquad v \in [v_{l-1}, v_{l+1}[\\
\boldsymbol{X}^-_{vvv} &:= \boldsymbol{X}_{vvv}(u, v_l^-) = \sum_{i=0}^{n} \boldsymbol{d}^{(0,3)}_{i,l-1} N_{i,4}(u) \qquad u \in [u_{k-1}, u_{k+1}[\\
\boldsymbol{X}^+_{vvv} &:= \boldsymbol{X}_{vvv}(u, v_l^+) = \sum_{i=0}^{n} \boldsymbol{d}^{(0,3)}_{i,l} N_{i,4}(u) \qquad u \in [u_{k-1}, u_{k+1}[\,.
\end{aligned} \tag{16}$$

The 10 linear constraints are therefore given by:

$$\begin{aligned}
\boldsymbol{X}^-_{uuu} = \boldsymbol{X}^+_{uuu} \quad &\Leftrightarrow \quad \boldsymbol{d}^{(3,0)}_{k-1,j} = \boldsymbol{d}^{(3,0)}_{k,j} \quad \text{for} \quad j = l-4, \ldots, l \,, \\
\boldsymbol{X}^-_{vvv} = \boldsymbol{X}^+_{vvv} \quad &\Leftrightarrow \quad \boldsymbol{d}^{(0,3)}_{i,l-1} = \boldsymbol{d}^{(0,3)}_{i,l} \quad \text{for} \quad i = k-4, \ldots, k \,.
\end{aligned} \tag{17}$$

25 control points $\boldsymbol{d}_{ij}$ $(i = k-4, \ldots, k,\ j = l-4, \ldots, l)$ are involved in these 10 equations. The least-squares approximation for this four-segment fairing step is therefore performed on 25 control points with 10 linear constraints (Figure 13):

$$\text{Minimize} \quad F(\hat{\boldsymbol{d}_{ij}}) = \sum_{i=k-4}^{k} \sum_{j=l-4}^{l} \|\boldsymbol{d}_{ij} - \hat{\boldsymbol{d}_{ij}}\|^2 \quad \text{subject to} \quad (17) \,. \tag{18}$$

This are only some possible examples of extensions, explaining the principle how to build a C^3-fairing step for bicubic B-spline surfaces. From now it is easy to construct further fairing methods based on two basic fairing masks, the 1-point and the 1-segment fairing masks. It is clear that more the C^3-condition is strong, less the fairing step is local. Let us define the *"localness"* $\mathcal{L}$ of a fairing step as the number of control points changed by it:

- 1-point fairing step: $\mathcal{L}$=9
- 4-point fairing step: $\mathcal{L}$=16
- 1-segment fairing step: $\mathcal{L}$=20
- 4-segment fairing step: $\mathcal{L}$=25

Nevertheless, if the given surface has enough control points, the 4-segment method is also quiet efficient.

Example 4: Figure 14 shows a test surface similar to that from example 1, it has much more control points: 25×25, which are perturbed a little more than in example 1. This

is not a realistic fairing candidate, but illustrates in some sense the worst case a fairing algorithm can meet.

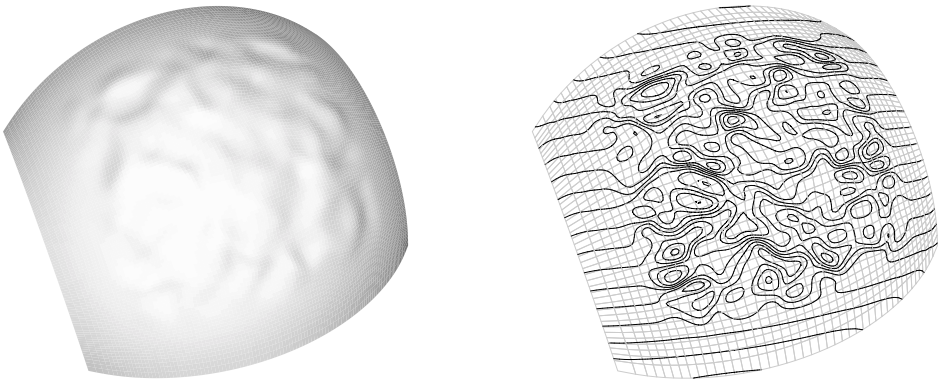

Figure 14: *Unfair surface with 525 control points*

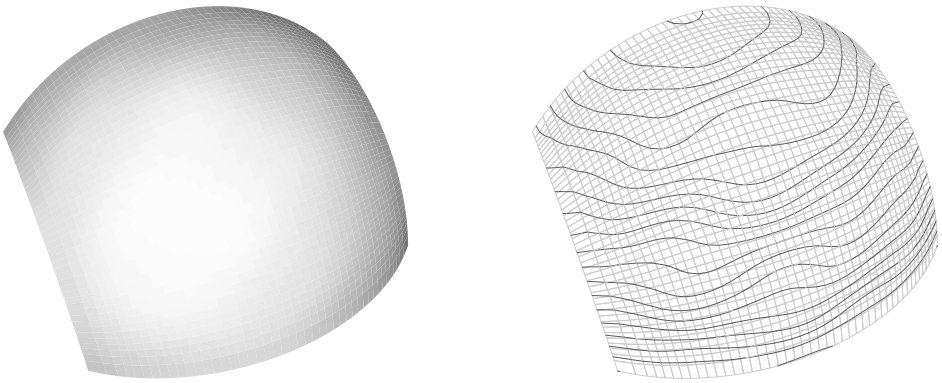

Figure 15: *Faired surface with 1-point fairing: 3000 iterations*

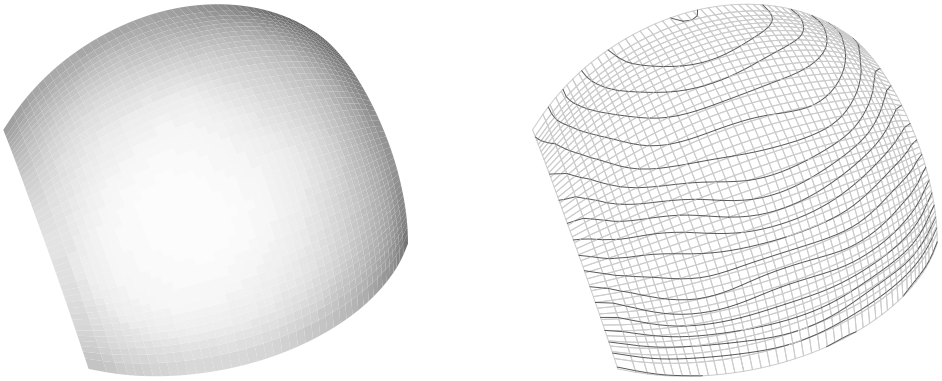

Figure 16: *Faired surface with 4-segment fairing: 5000 iterations*

We now compare the efficiency of the 1-point algorithm (Figure 15) and the 4-segment algorithm (Figure 16) for this test surface. The 4-segment algorithm provides a better result with the same number of iterations. This is due to the fact that the locality here is

bigger ($\mathcal{L} = 25$ vs. $\mathcal{L} = 9$). More control points are involved in a fairing step which enforces a regularity to the control points because of the C^3-condition. Both algorithms decrease the global fairness measure from $G_{X_{init}} = 6.5$ to $G^9_{X_{fair}} = 1 \times 10^{-3}$ and $G^{25}_{X_{fair}} = 5 \times 10^{-4}$ resp. And the maximum relative error between the new and initial control points is equal to 0.01. Running time for the 1-point algorithm on this example believes to 6.3 s, for the 4-point algorithm on this example it believes to 11.2 s.

If one applies up to 20.000 iterations with the 4-segment algorithm, one can decrease the global measure down to $G^{25}_{X_{new}} = 9 \times 10^{-5}$. But the isophotes don't look better. This is quite natural for this fairing method because only small surface imperfections can be faired without changing the surface too much.

6. SUMMARY AND FINAL REMARKS

An automatic and local fairing method for bicubic B-spline surfaces has been presented. The method is efficient for small surface imperfections as illustrated with various examples. Further fairing methods were proposed, which can be seen as extensions of the main method. The basic idea is to apply special masks on a subset of control points at each fairing step. The fairing process then iteratively results in a fairer surface.

To close this paper the following remarks can be made:

Flexibility: The algorithm is flexible in the sense that different fairing steps can be applied. The local fairing steps are based on a constrained least-squares fitting on a small number of control points. They consist of enforcing the surface to be C^3 at some points or between some patches. More the C^3-condition is strong, more control points are involved in the fairing step.

Tolerance control: Another open point we didn't mention until yet is the control of tolerances. A given tolerance can be preserved *a posteriori* by allowing only steps which keep the new surface $\hat{\boldsymbol{X}}$ inside a prescribed tolerance. Probably the fairing result would be poorer if the tolerance is too small bacause not all necessary fairing steps will be performed.

Smoothing some local surface region: If it is desired to apply fairing only to a limited region of the surface, this can easily be done by restricting the set of inner knots. The local fairing steps are then applied only to the so-called *free inner knots* of $J \subset I$ (see Figure 17). Sometimes it is necessary to maintain border positions and derivatives of the surface. The algorithm should therefore not affect the two rows and columns of control points near the border, i.e. two rows and columns of interior knots should be kept out of the fairing process.

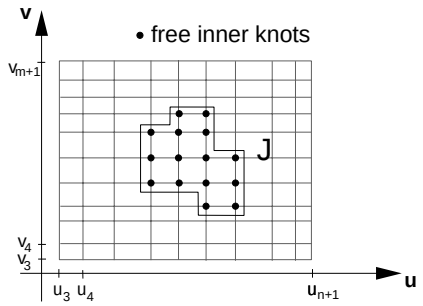

Figure 17: *Subset J of free inner knots*

References

1. Beier K.-P., Highlight-line algorithm for real-time surface quality assessment, Computer Aided Design **26** (1994), 268–277.
2. Böhm, W., Inserting new knots into B-spline curves, Computer Aided Design **12** (1980), 199–201.
3. Bonneau G.-P., and H. Hagen, Variational design of rational Bézier curves and surfaces, in *Curves and Surfaces in Geometric Design*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters, Wellesley MA, (1994), 51–58.
4. Brunet, P., Increasing the smoothness of bicubic spline surfaces, Computer Aided Geometric Design **2** (1985), 157–164.
5. De Boor, C., *A Practical Guide to Splines*, Springer, New York, 1978.
6. Eck, M., and J. Hadenfeld, Knot removal for B-spline curves, Computer Aided Geometric Design **12** (1995), 259–282.
7. Farin, G., G. Rein, N. Sapidis, and A. J. Worsey, Fairing cubic B-spline curves, Computer Aided Geometric Design **4** (1987), 91–103.
8. Ferguson, D. R., P. D. Frank, and K. Jones, Surface shape control using constrained optimization on the B-spline representation, Computer Aided Geometric Design **5** (1988), 87–103.
9. Greiner, G., Variational design and fairing of spline surfaces, Proc. Eurographics 1994, 143–154.
10. Hadenfeld, J., Local energy fairing of B-spline surfaces, in *Mathematical Methods for Curves and Surfaces*, Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), Vanderbilt University Press, Nashville & London, 1995
11. Hagen H., Geometric spline curves, Computer Aided Geometric Design **2** (1985), 223–227.
12. Hagen H., and G. Schulze, Automatic smoothing with geometric surface patches, Computer Aided Geometric Design **4** (1987), 231–235.
13. Hagen H., and P. Santarelli, Variational design of smooth B-spline surfaces, in *Topics in surface modeling*, H. Hagen (ed.), SIAM (1992), 85–92.

14. Hahmann, S., Visualization techniques for surface analysis, to be published in *Data Visualization Techniques*, C. Bajaj (ed.), 1998.
15. Hahmann, S., and S. Konz, Knot-removal surface fairing using search strategies, to be published in CAD, 1998.
16. Hoschek J., and D. Lasser, *Fundamentals of CAGD*, A K Peters, 1993.
17. Kjellander, J. A., Smoothing of cubic parametric splines, Computer Aided Design **15** (1983), 175–179.
18. Kjellander, J. A., Smoothing of bicubic parametric surfaces, Computer Aided Design **15** (1983), 289–293.
19. Klass R., Correction of local irregularities using reflection lines, Computer Aided Design **12** (1980), 73–77.
20. Lott, N. J., and D. I. Pullin, Method for fairing B-spline surfaces, Computer Aided Design **20** (1988), 597–604.
21. Lyche, T., and K. Morken, Knot removal for parametric B-spline curves and surfaces, Computer Aided Geometric Design **4** (1987), 217–230.
22. Mehlum E., Nonlinear spline, in *Computer Aided Geometric Design*, R. Barnhill, R. Riesenfeld (eds.), Academic Press, (1974), 173–208.
23. Moreton, H. P., and C. H. Séquin, Functional optimization for fair surface design, Comp. Graph. **26** (1992), 167–176.
24. Nielson G.M., Some piecewise polynomial alternatives to splunes under tension, in *Computer Aided Geometric Design*, R. Barnhill, R. Riesenfeld (eds.), Academic Press, (1974), 209–235.
25. Nowacki H., D. Reese, Design and fairing ship surfaces, in *Surfaces in CAGD*, R. Barnhill, W. Böhm (eds.), North-Holland Publ. (1983), 121–134.
26. Poeschl T., Detecting surface irregularities using isophotes, Computer Aided Geometric Design **1** (1984), 163–168.
27. Rando T., and J. A. Roulier, Designing faired parametric surfaces, Computer Aided Design **23** (1991), 492–497.
28. Sapidis, N., and G. Farin, Automatic fairing algorithm for B-spline curves, Computer Aided Design **22** (1990), 121–129.
29. Schumaker, L. L., *Spline Functions: Basic Theory*, Wiley, New York, 1981.
30. Strang, G., *Introduction to applied mathematics*, Wellesley-Cambridge Press, 1986.