Automatic Modeling of Smooth Spline Surfaces

Günther Greiner and Hans-Peter Seidel
IMMD IX, Graphische Datenverarbeitung
Universität Erlangen–Nürnberg
Am Weichselgarten 9
91058 Erlangen, Germany

Summary: Spline surfaces are very popular for the modeling of free form surfaces. This is because they are very flexible, conceptually simple and the shape can be easily controlled by manipulating control points. However, for complex shapes one has to deal with an overwhelming number of control points. It is therefore increasingly important to build tools that enable the designer to specify only a few geometric constraints while automatically determining the explicit representation of the surface. In this paper we describe the basic concepts of how such tools can be build.

1 Introduction

In many areas where free form surfaces have to be designed, the problem arises of constructing spline surfaces satisfying certain constraints. In addition one requires that these surfaces are fair, i.e., they have a pleasant looking shape. An example is the construction of blend surfaces (see [19]). The task is to fill a hole in an already existing fair surface or to construct a transition between two given surfaces. The aim is that the composite surface has an overall pleasant looking shape. Another example, occurring e.g. in reverse engineering, is the interpolation of scattered data (see e.g. [6]). One has to construct a smooth surface interpolating a set of discrete (measured) data. In other situations, an already existing fair spline surface has to be modified in certain areas. In the case that the surface is determined by a large number of control points, these manipulations may produce wrinkles or other flaws in parts of the surface. In a second, post processing step, one has to smooth these regions. This procedure is called fairing (see [12] for basics on the subject).

In CAGD one approach has gained a lot of interest recently. It is called called *variational design* and can handle these kind of problems very efficiently (see e.g. [1, 2, 3, 7, 8, 11, 13, 15, 16, 18]). In the present paper we want to describe the basic ideas and will explain how this method can be applied to the above mentioned problems.

The paper is organized as follows. In Sec. 2 we describe the general procedure, and explain which kind of constraints are permissible. In Sec. 3 the problem of how to measure the fairness of a surface will be addressed in detail. In Sec. 4 we outline a number of applications to concrete problems. We explain how blend surfaces can be constructed, we describe a fairing procedure, and we look at a method for interpolating scattered data. In the final section we give the conclusion of our investigations.

2 The general approach

The task is to construct surfaces having pleasant shape that satisfy certain geometric constraints. This is rather open ended, so we have to explain what "pleasant shape" means and what kind of "geometric constraints" we are considering. In this section we try to be as general as possible. In the following sections more concrete issues are discussed.

First of all, we will describe what type of surfaces we plan on working with. We consider some class $\mathcal{S}$ of parameterized surfaces. Each $F \in \mathcal{S}$ is a mapping from a fixed parameter space $\Omega \subset \mathbb{R}^2$ into $\mathbb{R}^3$. Moreover, we exclusively deal with spline surfaces, i.e. we consider piecewise polynomial surfaces (see [5]). These surfaces usually are characterized in the following way. One specifies a fixed partition of the parameter space Ω in patches (triangles or quadrangles). Then a spline surface F is uniquely determined by a set of *control points*. For modeling purposes these control points act as degrees of freedom (dofs), which finally have to be adjusted. $\mathcal{S}$ may consist of tensor product splines (over a rectangular grid), or of piecewise Bézier patches (over a triangulation), or of triangular splines (over a triangulation and an associated cloud of knots, see [4]) or a scheme of split surfaces (e.g., Clough–Tocher or Powell–Sabin). It is very important that the class has sufficiently many degrees of freedom (dofs). We require many more dofs than are needed to satisfy the geometric constraints that we describe next. This generally will be achieved by *a priori* specifying a sufficiently fine partition of the parameter space Ω.

The constraints may have the following form:

Interpolation $F(u_{1k}, u_{2k}) = y_k$ for finitely many parameter values $(u_{1k}, u_{2k}) \in \Omega$ and specified values $y_k \in \mathbb{R}^3$.

Tangent planes $\left\langle \frac{\partial F}{\partial u_i}(v_{1k}, v_{2k}) \middle| p_k \right\rangle = 0 \quad (i = 1, 2)$ for finitely many parameter values (v_{1k}, v_{2k}) and specified normal vectors p_k.

Blending Along the boundary of the parameter space, F has prescribed values and prescribed cross-boundary derivatives.

Fairing An already existing surface may be changed only in a specified region (in such a way that its shape is improved).

In addition, one could prescribe curvature information (e.g. Dupin's indicatrix) at certain points, or require that a given curve must be interpolated.

"Pleasant shape" also called *fairness* has a variety of possible interpretations depending on the specific application. In any case, we will measure the fairness by assigning every parameterized surface F a scalar value $J(F)$ thus defining a *fairness functional* $J : F \mapsto J(F)$. The idea is that a smaller value of $J(F)$ indicates that F has a more pleasant shape. To get a more concrete idea of a fairness functional one can imagine that $J(F)$ represents the 'total curvature' of the surface. In the next section we will describe a variety of different fairness functionals that are used in the literature.

The whole procedure can be outlined as follows.

1. Specify the constraints to be met;
2. select a class of spline surfaces $\mathcal{S}$ having sufficiently many dofs;
3. Choose an appropriate fairness functional J defined for surfaces of $\mathcal{S}$.
4. Determine the surface $F \in \mathcal{S}$ such that
   - F satisfies the constraints and
   - $J(F) \leq J(G)$ for any $G \in \mathcal{S}$ satisfying the constraints.

3 Appropriate Fairness Functionals

There is no universal fairness functional that can be used for every application. On the contrary, a whole variety of functionals is used by different authors. Some are motivated by physical reasoning, describing a certain energy, others have a geometric meaning e.g. measuring total curvature. What is common to all of them is that they dependent on (partial) derivatives of the parameterization F . The highest order of the partial derivatives that enter the functional is called the *order of the functional.* Mostly used are second and third order functionals.

We classify commonly used fairness functionals by their (numerical) complexity. There are *simple quadratic* functionals, *non-linear exact* functionals and, somehow in-between, *data dependent* functionals.

In the following we give an overview of frequently used exact functionals and the corresponding simplifications. Then we describe in more detail the concept of data dependency.

3.1 The exact functionals and their simplifications

In Table 1, we give a list of fairness functionals that have been used in the literature. To every *exact* functional its *simplified* version is listed. κ_{Gi} $(i = 1, 2)$ denote the principal curvatures of the surface G and $d\omega_G$ is the infinitesimal area element. Moreover, ΔG denotes the Laplacian, $\Delta G = \frac{\partial^2 G}{(\partial u_1)^2} + \frac{\partial^2 G}{(\partial u_2)^2}$.

The simplified thin plate energy J_7 is used e.g. by Halstead, Kass, DeRose [11] and Kallay [13]. In certain instances, namely when considering problems having fixed boundary data (e.g. blending problems) the simpler functional J_8, yields the same optimal surfaces as J_7 (see Corollary to Prop. 2 in [7]). The simple third order functional J_5 has been used in recent work of Dietz & Hoschek. The exact (non-linear) functionals J_2 and J_4 have been investigated by Moreton & Sequin. They used them to design closed surfaces (of positive genus) having optimal shape (see [15, 16]).

The following more general quadratic second order functional is used by Welch & Witkin (see [18]) and Celniker & Gossard (see [3])

$$J : G \mapsto \int_\Omega \sum_{i,j=1}^{2} \alpha_{ij} \left\langle \frac{\partial G}{\partial u_i} \middle| \frac{\partial G}{\partial u_j} \right\rangle + \beta_{ij} \left(\frac{\partial^2 G}{\partial u_i \partial u_j} \right)^2 du_1 du_2 \ .$$

meaning	order	exact	simplification
area functional	1	$J_1 : G \mapsto \int_\Omega 1\, d\omega_G$	$J_6 : G \mapsto \int_\Omega \left(\frac{\partial G}{\partial u_1}\right)^2 + \left(\frac{\partial G}{\partial u_2}\right)^2 du_1 du_2$
thin plate energy or *total curvature f.*	2	$J_2 : G \mapsto \int_\Omega {\kappa_G}_1^2 + {\kappa_G}_2^2\, d\omega_G$	$J_7 : G \mapsto \int_\Omega \left(\frac{\partial^2 G}{(\partial u_1)^2}\right)^2 + 2\left(\frac{\partial^2 G}{\partial u_1 \partial u_2}\right)^2 + \left(\frac{\partial^2 G}{(\partial u_2)^2}\right)^2 du_1 du_2$
mean cuvature f. / Laplacian	2	$J_3 : G \mapsto \int_\Omega (\kappa_{G1} + \kappa_{G2})^2\, d\omega_G$	$J_8 : G \mapsto \int_\Omega \Big(\Delta G\Big)^2 du_1 du_2$
variation of curvature	3	$J_4 : G \mapsto \int_\Omega \left(\frac{\partial \kappa_{G1}}{\partial e_1}\right)^2 + \left(\frac{\partial \kappa_{G2}}{\partial e_2}\right)^2 d\omega_G$	$J_9 : G \mapsto \int_\Omega \left(\frac{\partial(\Delta G)}{\partial u_1}\right)^2 + \left(\frac{\partial(\Delta G)}{\partial u_2}\right)^2 du_1 du_2$
general case	3	$J_5 : G \mapsto J_4(G) + \alpha_2 J_2(G) + \alpha_1 J_1(G)$	$J_{10} : G \mapsto J_9(G) + \alpha_2 J_7(G) + \alpha_1 J_6(G)$

Table 1: Survey of exact and simplified fairness functionals

Celniker & Gossard choose the coefficients α_{ij}, β_{ij} by physical reasoning, while Welch & Witkin's choice is based on geometric properties (see [3, 18] for details). In the next section we describe a general and systematic approach for choosing the coefficients thus obtaining appropriate fairness functionals for a great variety of design problems.

3.2 Data dependent functionals

The exact (highly non-linear) functionals presented in the previous section are far too complex to produce results at a speed suited for interactive use. On the other hand, the simple quadratic functionals are capable of producing results in real time. However, the quality of the resulting surfaces is, in many cases, quite poor. Thus simple functionals on the one hand and the exact functionals on the other are extreme cases with respect to the computational cost and quality of the resulting surface. See [14] for a discussion of quality versus computational cost for different fairness functionals.

In the following we will describe how one can find functionals that combine the advantages of these two extremes. The goal is to find a quadratic functional that is locally a good approximation to the corresponding exact functional. For nearly flat surfaces the simple functionals presented in Sec. 3.1 will do that. For the arbitrary case this can be achieved by introducing *reference surfaces*, sometimes also called *non-planar parameter spaces*.

The procedure works as follows. A reference surface $G_0 : \Omega \to \mathbb{R}^3$ is chosen that roughly has the shape of the surface we are looking for. One may consider this surface G_0 as a non-planar parameter space for the final surface.

A quadratic functional that will be a good approximation for J_3 is defined by replacing the usual Laplacian in J_8 by the *Laplace-Beltrami*

$$J_{13} : G \mapsto \int_\Omega \Big(\mathbf{\Delta}_{G_0} G\Big)^2 \, d\omega_{G_0} \ . \tag{1}$$

The Laplace-Beltrami is the Laplacian on the surface G_0, defined by

$$\mathbf{\Delta}_{G_0} f = \mathbf{div}_{G_0}(\mathbf{grad}_{G_0} f)$$

with *gradient* and *divergence* with respect to the metric induced by G_0,

$$\mathbf{grad}_{G_0} f = \Big(\tfrac{\partial f}{\partial u_1} \ , \ \tfrac{\partial f}{\partial u_2}\Big) \begin{pmatrix} g_{011} & g_{012} \\ g_{021} & g_{022} \end{pmatrix}^{-1}$$

$$\mathbf{div}_{G_0}(f_1, f_2) = \tfrac{1}{\sqrt{g_0}} \Big(\tfrac{\partial}{\partial u_1}(\sqrt{g_0} f_1) + \tfrac{\partial}{\partial u_2}(\sqrt{g_0} f_2)\Big) \ .$$

The functional J_{13} is quadratic and of second order. Moreover, for surfaces G that roughly follow the a priori chosen reference surface G_0 we have $J_{13}(G) \sim J_3(G)$. A more precise formulation of this statement can be found in [9]. The choice of the reference surface G_0 will depend on the constraints that the final solution has to satisfy. In most cases these constraints will indicate what the final result is going to look like. This then allows us to find a reference surface G_0. Note that G_0 does <u>not</u> have to satisfy the constraints. The construction shows that the functional J_{13} is not a universal functional. Rather it depends on G_0, hence ultimately on the constraints that need to be met. Thus it is a *data dependent* functional!

In the same way one can construct data dependent functionals of first and third order. Again one has to choose a reference surface G_0. Then the data dependent version of the first order functional J_1 and J_6 resp. is given by

$$J_{11} : G \mapsto \int_\Omega \Big\langle \mathbf{grad}_{G_0} G \Big| \, \mathbf{grad}_{G_0} G \Big\rangle \, d\omega_{G_0} \tag{2}$$

and the data dependent version of the third order functional J_4/J_9 is given by

$$J_{14} : G \mapsto \int_\Omega \Big\langle \mathbf{grad}_{G_0}(\mathbf{\Delta}_{G_0} G) \Big| \, \mathbf{grad}_{G_0}(\mathbf{\Delta}_{G_0} G) \Big\rangle \, d\omega_{G_0} \tag{3}$$

Finally we want to point out, that it is also possible to define a data dependent, quadratic version of J_7 , which may serve as a good local approximation to the minimal energy functional J_2 . To do this one has to introduce the generalized Hessian $\mathbf{Hess}_{G_0}(f)$ of a scalar function f with respect to a reference surface G_0 . Similar to the usual Hessian, $\mathbf{Hess}_{G_0}(f)$ is a 2×2 matrix. The data dependent version of the thin plate energy functional J_2 and J_7 resp. is then given by:

$$J_{12} : G \mapsto \int_\Omega \textstyle\sum_{w=x,y,z} trace\Big((\mathbf{Hess}_{G_0}(G_w))^2\Big) \, d\omega_{G_0} \tag{4}$$

Here G_x, G_y and G_z are the three components of the parameterization G.

We conclude the presentation of the data dependent fairness functional by remarking that all the functionals introduced in this final section are of the following general form.

$$J(G) = \int_\Omega \sum_{1\le|\alpha|,|\beta|\le 3} w_{\alpha,\beta} \langle D^\alpha G | D^\beta G \rangle \, du_1 du_2 \qquad (5)$$

Here we use multi-index notation: α and β are multi-indices and $D^\alpha G$ denotes the corresponding partial derivative of G . The weight functions $w_{\alpha,\beta}$ depend only on the reference surface G_0. This shows, in particular, that all these functionals are quadratic. This implies that finding the solution to the optimization problem reduces to solving a linear system.

Figure 1: Blending: simple configuration of two primary surfaces.

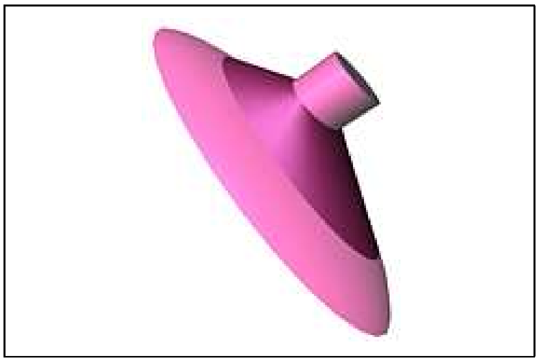

Figure 2: The reference surface obtained by linearly interpolating the boundary curves.

4 Applications

Having specified the constraints, the class $\mathcal{S}$ of spline surfaces and a fairness functional J, we have to solve the constraint optimization problem

Determine $F \in \mathcal{S}$ such that
- F satisfies the constraints and
- $J(F) \le J(G)$ for any $G \in \mathcal{S}$ satisfying the constraints.

Before we consider three concrete problems we give some general remarks. In the case that J is one of the highly non-linear functionals presented in Sec. 3.1, nonlinear optimization techniques have to be applied to find the minimum. Moreton & Sequin use a conjugate gradient method (details are given in [15]). As a consequence, the numerical computation for even small problems needs several hours to complete; more complex problems require days of computation. For most applications this is simply not acceptable. We therefore restrict ourselves in the following to quadratic fairness functionals. As mentioned earlier, the minimization process then leads to solving a linear system. Note that the constraints we are considering are also of linear nature! Therefore the optimization process only needs from a fraction of a second (for simple quadratic functionals and moderately sized data sets) to several minutes (for large data sets and data dependent fairness functionals).

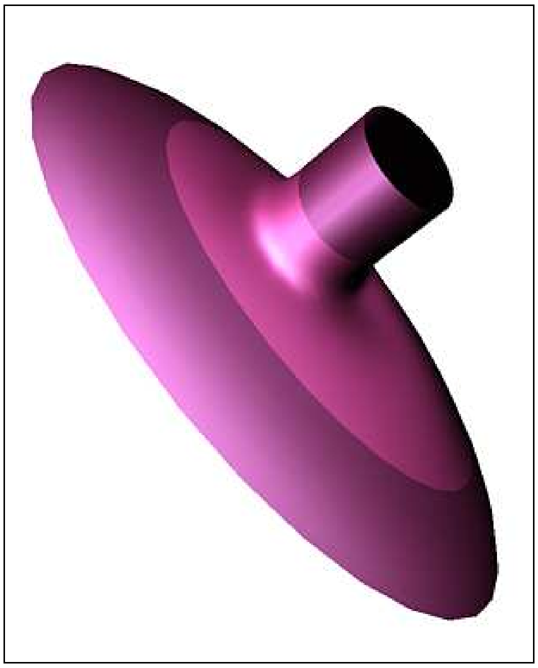

Figure 3: Blend surface obtained by optimization (reference surface : linear interpolation of the boundary curves).

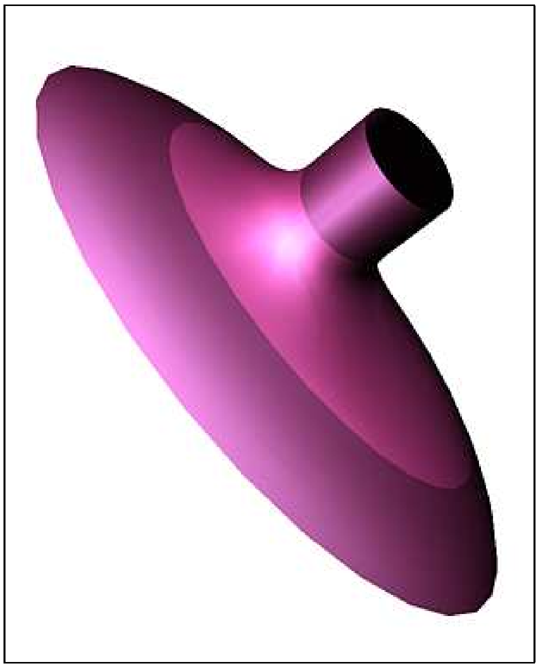

Figure 4: Blend surface obtained by optimization (reference surface : the blend surface from Fig. 3).

There are two possible ways of dealing with the constraints.

1) When one is able to single out a basis of the affine subspace $\mathcal{S}_0$ of all the parameterizations that satisfy the constraints, the whole problem reduces to quadratic unconstrained minimization problem in the finite dimensional affine space $\mathcal{S}_0$.

2) If it is not possible to single out a basis of $\mathcal{S}_0$, or if this is too complicated, one uses Lagrangian multipliers to handle the constraints. Due to the linearity of the constraints, the resulting system is still linear!

4.1 Blending

We want to construct a smooth transition between two primary surfaces. A simple configuration, cylinder and plane with circular hole is shown in Fig. 1. We assume that the primary surfaces are tensor product B-spline surfaces and we construct the blend surface as tensor product B-spline surface as well. Thus $\mathcal{S}$ is the set of all tensor product B-spline surfaces over a rectangular grid. Since we want smooth transitions between the primary surfaces and the blend surface, two layers of control points along each of the boundary curves will be fixed. It is easy to specify a basis of the affine subspace $\mathcal{S}_0$ consisting of those B-spline surfaces that satisfy the required conditions along the boundary curves. In fact, all the B-spline basis functions that correspond to interior grid points (grid distance to the boundary points at least two) form a basis (of the associated linear subspace).

We use as fairness functional J_{13}, a data dependent second order functional. The reference surface G_0 (shown in Fig. 2) is obtained by linearly connecting the boundary curves of the two primary surfaces. The blend surface obtained by minimizing this functional is shown in Fig. 3. It is already a fairly good, but can be improved in the following way. The blend surface we just constructed may serve as new reference surface, thus defining a new quadratic functional. The minimization of this functional yields the surface shown in Fig. 4. Of course one can perform an additional step; we may even iterate. However, it turns out that a possible third step hardly changes the shape of the blend surface.

A more detailed description of the construction just described can be found in [7].

4.2 Fairing

To avoid further technical details we again consider the simplest class of spline surfaces, namely tensor product B-spline surfaces defined over a rectangular grid.

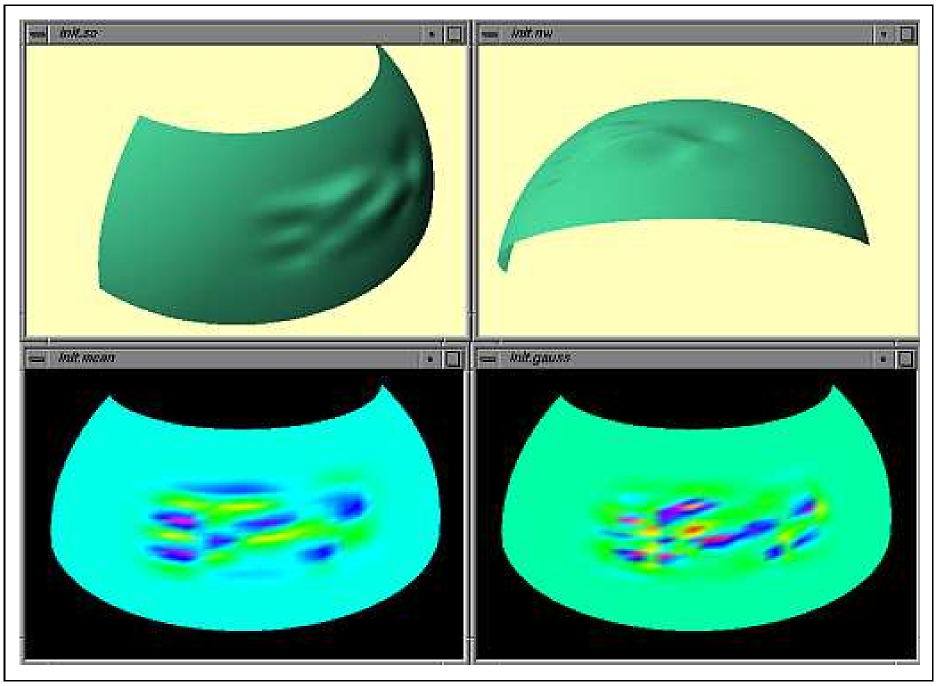

Figure 5: A sphere-like surface that has to be faired. Two views of the shaded image (upper images), the mean curvature (lower left) and the Gaussian curvature (lower right)

We assume that there is a parameterized tensor product B-spline surface $F_0 : \Omega \to \mathbb{R}^3$ having some wrinkles in a sub-domain Ω_0. We consider as example the sphere like surface shown in Fig. 5. In a first step we identify all the control points that influence the region Ω_0. Only these control points are allowed to be modified. They act as dofs in the optimization process. The remaining control points are kept fixed, thus forming the constraints for the minimization process. This time J_{14}, the data dependent fairness functional of third order is used. As reference surface G_0 we use a least square fit to the set of all control points of F_0.

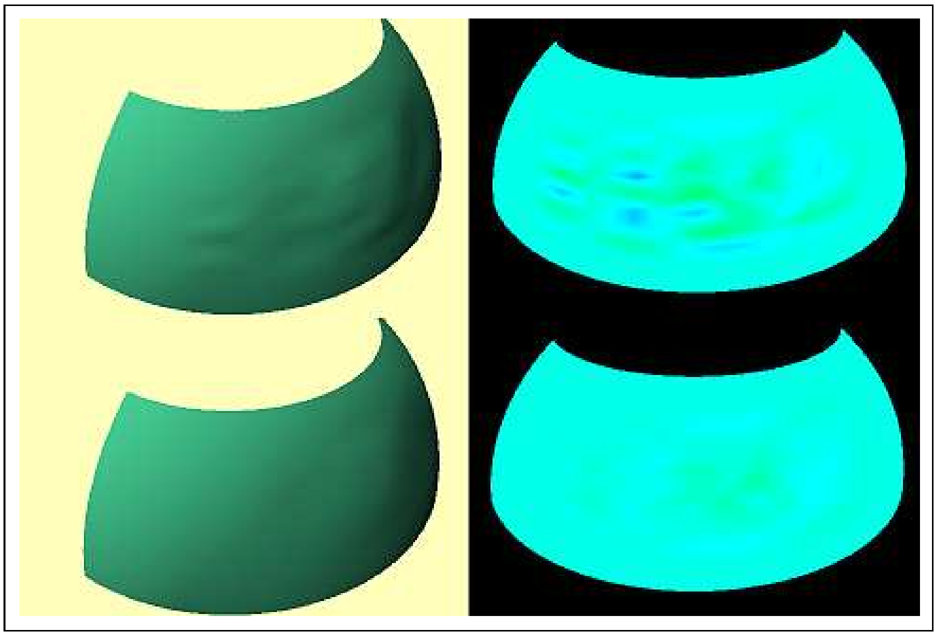

Figure 6: Fairing by optimizing data dependent fairness functionals. Upper images: First step: Fairing of the surface given in Fig. 5 (shaded image and mean curvature plot); Lower images Second step: Fairing of the surface shown in the upper image

The surface minimizing this functional (and satisfying the constraints) is shown in the upper part of Fig. 6: upper left a shaded image and upper right a plot of the mean curvature. The resulting surface is much fairer than the initial surface. It still has some wrinkles, which can be seen when inspecting the curvature plot. One can perform the same procedure again and again, thus iteratively improve the surface. In the lower part of Fig. 6, the surface obtained after a second step is shown.

As in Sec 4.1 one can perform the whole procedure once more, thus obtaining an even better result. In [8] a more sophisticated, iterative procedure for fairing surfaces by minimizing suitable data dependent fairness functionals is described. We refer the interested reader there for further details.

4.3 Scattered Data Interpolation

In its simplest form the problem can be stated as follows. Given data points $(x_i, y_i),\ i = 0, 1, .., N$ in the plane and corresponding scalar data values $z_i,\ i = 0, 1, .., N$ construct a function $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ that interpolates the values z_i, i.e., $f(x_i, y_i) = z_i$. In addition one requires that the graph of the function f is a fair surface.

The variational approach to this problem is briefly described as follows: Choose an equidistant rectangular 'fine grid' say of size $n \times m$ which contains in its interior all the data points. Typically the number of grid lines n and m is determined by the condition $n \cdot m > 10N$. In order to define a data dependent fairness functional we construct a reference surface by a least square fit to the data values. The least square fit is a tensor product B-spline function over a ‘coarse grid’. The graph of this function serves as the reference surface which is then used to define the data dependent fairness functional.

Figure 7: Interpolating Franke’s surface. Upper image: two views of the test surface with 33 and 100 sampled points; middle image: reconstruction by interpolating 33 points; lower image: reconstruction by interpolating 100 points.

In the examples shown in Fig. 7 the data dependent version of the thin plate energy functional has been used, i.e. the functional J_{12}. The upper part of Fig. 7 shows one of the standard test functions for scattered data interpolation methods. The surface in the middle has been obtained by interpolating the test function at 33 points (the ones marked upper left). It is a cubic tensor product B-spline function, grid size 20×20. The reference surface was the cubic tensor product B-Spline function with grid size 5×5 obtained by a least square fit. The surface shown in the lower part was obtained by interpolating the test function at 100 points (the ones marked upper right). It is a cubic tensor product B-spline function, grid size 40×40. The reference surface was the tensor product B-Spline with grid size 8×8.

More details on the interpolation method we briefly described can be found in [10]. A variational approach to scattered data interpolation which is based on triangular B-splines is described in [17].

References

[1] G.-P. Bonneau, H. Hagen, and St. Hahmann. Variational surface design and surface interrogation. *Computer Graphics Forum*, 12(3):447–459, 1993.

[2] G. Brunnett, H. Hagen, and P. Santarelli. Variational design of curves and surfaces. *Surv. Math. Industry*, 3:1–27, 1993.

[3] G. Celniker and D. Gossard. Deformable curve and surface finite–element for free–form shape design. *ACM Computer Graphics*, 25:257–266, 1991.

[4] W. Dahmen, C. A. Micchelli, and H.-P. Seidel. Blossoming bedgets B–spline bases built better by B–patches. *Math. Comp.*, 59:97–115, 1992.

[5] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Boston, 1993.

[6] R. Franke and G. M. Nielson. Scattered data interpolation and applications: A tutorial and survey. In H. Hagen and D. Roller, editors, *Geometric Modelling, Methods and Applications*, pages 131–160. Springer Verlag, 1991.

[7] G. Greiner. Surface construction based on variational principles. In P. J. Laurent, A. LeMéhauté, and L. L. Schumaker, editors, *Wavelets, Images, and Surface Fitting*, pages 277–286, Wellesley MA, 1994. AKPeters.

[8] G. Greiner. Variational design and fairing of spline surfaces. *Computer Graphics Forum*, 13(3):143–154, 1994.

[9] G. Greiner. Data dependent fairness functionals. to appear in CAGD, 1995.

[10] G. Greiner, A. Kolb, and Riepl A. Scattered data interpolation using data-dependent optimization techniques. Technical report, Universität Erlangen, 1995.

[11] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. *ACM Computer Graphics*, 27(35–44):35–44, 1993.

[12] J. Hoschek and D. Lasser. *Grundlagen der geometrischen Datenverarbeitung.* B. G. Teubner, Stuttgart, 1992.

[13] M. Kallay. Constrained optimization in surface design. In B. Falcidieno and T. L. Kunii, editors, *Modeling in Computer Graphics*, Berlin, 1993. Springer–Verlag.

[14] M. Lounsberry, St. Mann, and T. DeRose. Parametric surface interpolation. *IEEE Comp. Graphics Appl.*, 12:45–52, 1992.

[15] H. P. Moreton and C. Séquin. Functional optimization for fair surface design. *ACM Computer Graphics*, 26:167–176, 1992.

[16] H. P. Moreton and C. H. Séquin. Scale-invariant minimum-cost curves: Fair and robust design implements. *Computer Graphics Forum*, 12(3):473–484, 1993.

[17] R. Pfeifle and H.-P. Seidel. Fitting triangular B-splines to functional scattered data. In *Proceedings of Graphics Interface '95*, pages 80–88. Canadian Human-Computer Communications Society, Morgan Kaufmann, 1995.

[18] W. Welch and A. Witkin. Variational surface modeling. *ACM Computer Graphics*, 26:157–166, 1992.

[19] L. R. Woodwark. Blends in geometric modelling. In R. R. Martin, editor, *The mathematics of surfaces II*, pages 255–297. Oxford University Press, 1987.