# Local Energy Fairing of B–Spline Surfaces

Jan Hadenfeld

**Abstract.** An algorithm for local fairing of B-spline surfaces of arbitrary order is presented. This stepwise algorithm is based on the idea of changing only one control point in each iteration step. Which point has to be changed is determined by a ranking-list where all control points are sorted in regard to their corresponding values of a suitable chosen fairness criterion. The criterion is chosen in such a way that the problem leads to linear equations. This method is an extension of an approach for curve fairing [3].

## §1. Introduction

In many cases a B-spline surface is the result of an interpolation or approximation of a set of points. Unfortunately in many cases there has to be an after-treatment, because the surface is *nonsmooth*. One of the main reasons for this after-treatment is that the given set of points can contain defects because of the digitizing process. Also the higher the degree of the approximation surface, the higher the tendency to be wavy.

Therefore a fairing method is required which assists the designer in producing a *smooth* surface. Although the fairness of a surface can only be described in subjective manner, there are always two steps of fairing a given surface which have to be done:

1) Find the undesirable regions.
2) Eliminate these regions.

According to these two steps there are many methods which can be found in the literature on fairing surfaces. To detect the undesirable regions of a surface the usage of reflection lines [11,14] is very common in modern CAD systems. And with the help of isophotes [16] the continuity of spline surfaces at the segment boundaries can be visualized. Another way is to use the *k-orthotomics* [10] which are known from optics. Once the regions which have to be faired are found, one of these methods can be used for fairing. Unfortunately not all of them work automatically.

Our approach was motivated by a method of Farin and Sapidis [6,17] for fairing cubic B-spline curves by subsequently removing and reinserting knots.

Mathematical Methods for Curves and Surfaces
M. Dæhlen, T. Lyche, and L. L. Schumaker (eds.), pp. 203–212.

ISBN 8265-1268-2.

In doing so, they raise the continuity at specified knots by one and reduce in this way the discontinuities in the slope of curvature between adjacent cubic pieces. Another local fairing method for cubic B-spline curves was developed by Farin [5] whereby the main part is a *degree reduction – degree re-elevation* step.

They extended their methods described in [5,6,17] to tensor product B-spline surfaces in the following way: First interpreting all rows of the control net as control polygons of B-spline curves and fair this curves. Then do the same for the columns of the resulting control net.

Another algorithm for smoothing was developed by Kjellander who extended in [13] his method for cubic curves [12] to surfaces. This method moves a data point to a better location and then interpolates the new data set with a $C^2$ cubic spline. Unfortunately, a large linear system has to be solved and the undesirable regions of the surface are not found automatically. Moreover, this method is *global*.

In this paper we generalize a strategy which already has been successfully used for curves [3] to B-spline surfaces. For curves we locally minimize the linearized *strain energy* of a thin elastic beam by modifying only one control point in each iteration step. For surfaces we choose in this paper the thin plate energy as the fairness criterion. Moreover we use a so-called *ranking-list* to detect the undesirable regions which have to be faired.

In most papers, a surface is said to be smooth if the energy of a thin plate described by the surface is minimal. In [2], the potential energy of a thin elastic plate is given by

$$\Pi_P = \iint_S a(\kappa_1^2 + \kappa_2^2) + 2(1-b)\ \kappa_1 \kappa_2\ dS, \tag{1}$$

where $\kappa_1$ and $\kappa_2$ are the principle curvatures of the plate and $a, b$ are constants which depend on the material.

Choosing $a = 1$ and $b = 0$ and following the technical assumption (which is also common practice) that the plate has only small deflections we get an approximation for the thin plate energy

$$\Pi = \iint_A \mathrm{X}_{uu}^2 + 2\ \mathrm{X}_{uv}^2 + \mathrm{X}_{vv}^2\ du\, dv. \tag{2}$$

For more details on this linearization, we refer to e.g. [7]. In the next sections we will use this linearized energy (2) as our *fairness criterion* for fairing B-spline surfaces.

First, in Section 2 the different stages of the surface during the algorithm are described. In Section 3 the *ranking-number* is introduced and in Section 4 the new control point is calculated with help of this ranking-number. Then the *ranking-list* and the distance tolerance are discussed to create in Section 7 an algorithm for fairing B-spline surfaces. Finally, in Section 8 some examples are shown.

## §2. One Iteration Step

The idea of the fairing algorithm presented here is to minimize a given functional by changing only one control point in each step. For this iterative procedure, let us introduce the following notations:

1) The given (*nonfair*) B-spline surface of the order $(k,l)$ with the knot sequences $U = \{u_i\}_{i=0}^{n+k}$ and $V = \{v_j\}_{j=0}^{m+l}$ with $(u,v) \in [u_{k-1}, u_{n+1}] \times [v_{l-1}, v_{m+1}]$ is described by

$$\mathrm{X}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{d}_{ij}\, N_{i,k}(u)\, N_{j,l}(v), \tag{3}$$

with the normalized basis functions $N_{i,k}(u)$ and $N_{j,l}(v)$ and control points $\mathbf{d}_{ij}$ in general dimension.

2) The B-spline surface which has already been faired by a certain number of iterations by

$$\bar{\mathrm{X}}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \bar{\mathbf{d}}_{ij}\, N_{i,k}(u)\, N_{j,l}(v). \tag{4}$$

3) And the surface in the next step with the new control point $\widetilde{\mathbf{d}}_{r_1 r_2}$ by

$$\widetilde{\mathrm{X}}(u,v) = \sum_{\substack{i=0 \\ (i,j)\neq(r_1,r_2)}}^{n} \sum_{j=0}^{m} \bar{\mathbf{d}}_{ij}\, N_{i,k}(u)\, N_{j,l}(v) + \widetilde{\mathbf{d}}_{r_1 r_2}\, N_{r_1,k}(u)\, N_{r_2,l}(v). \tag{5}$$

In (5) there are no restrictions to the index $(r_1, r_2)$ of the new control point $\widetilde{\mathbf{d}}_{r_1 r_2}$. But the B-spline surface is often a part of a set of surfaces and because of that it is undesirable to change the boundary curves. And if the surface already has any continuity with respect to its neighbors, it is also necessary to fix more control points to achieve the old continuity. So we restrict the index $(r_1, r_2)$ by $\alpha_1 \le r_1 \le n - \beta_1$ and $\alpha_2 \le r_2 \le m - \beta_2$.

The problem now is which control point has to be changed, and how to find the new location for this point. To solve this problem we will now introduce the so-called *ranking-number*.

## §3. The Ranking-Number

If we change a control point we want to get the best fairing result in this step, i.e., we are looking for the largest improvement of the energy integral. This improvement can be expressed by the following *ranking-number*

$$z_{r_1,r_2} = \Pi(\bar{\mathbf{d}}_{r_1 r_2}) - \Pi(\widetilde{\mathbf{d}}_{r_1 r_2}), \qquad \begin{array}{l} \alpha_1 \le r_1 \le n - \beta_1, \\ \alpha_2 \le r_2 \le m - \beta_2. \end{array} \tag{6}$$

This ranking-number describes the improvement of the energy integral if we change the control point $\bar{\mathbf{d}}_{r_1 r_2}$ to $\widetilde{\mathbf{d}}_{r_1 r_2}$.

Now, maximizing the number $z_{r_1,r_2}$ with respect to $\widetilde{\mathbf{d}}_{r_1 r_2}$ leads to the problem of minimization of the energy integral $\Pi(\widetilde{\mathbf{d}}_{r_1 r_2})$.

## §4. The New Control Point

The new location $\widetilde{\mathbf{d}}_{r_1 r_2}$ of the control point $\bar{\mathbf{d}}_{r_1 r_2}$ has to be chosen in such a way that the new surface $\widetilde{\mathrm{X}}$ minimizes the energy functional (2). The quadratic form has an unique minimum $\widetilde{\mathbf{d}}_{r_1 r_2}$ which is determined by

$$\frac{\partial \Pi(\widetilde{\mathbf{d}}_{r_1 r_2})}{\partial \widetilde{\mathbf{d}}_{r_1 r_2}} \stackrel{!}{=} 0. \tag{7}$$

The equation (7) can be solved explicitly for the control point $\widetilde{\mathbf{d}}_{r_1 r_2}$ and we obtain

$$\widetilde{\mathbf{d}}_{r_1 r_2} = \sum_{\substack{i=i_0 \\ (i,j)\neq(r_1,r_2)}}^{i_1} \sum_{j=j_0}^{j_1} \gamma_{ij}\, \bar{\mathbf{d}}_{ij}, \tag{8}$$

with the factors

$$\gamma_{ij} = -\frac{U^{2,2}_{i,r_1} V^{0,0}_{j,r_2} + 2\, U^{1,1}_{i,r_1} V^{1,1}_{j,r_2} + U^{0,0}_{i,r_1} V^{2,2}_{j,r_2}}{U^{2,2}_{r_1,r_1} V^{0,0}_{r_2,r_2} + 2\, U^{1,1}_{r_1,r_1} V^{1,1}_{r_2,r_2} + U^{0,0}_{r_1,r_1} V^{2,2}_{r_2,r_2}}, \tag{9}$$

and the abbreviations

$$U^{m,n}_{i,j} = \int_{u_0}^{u_1} N^{(m)}_{i,k}(u)\, N^{(n)}_{j,k}(u),$$
$$V^{m,n}_{i,j} = \int_{v_0}^{v_1} N^{(m)}_{i,l}(v)\, N^{(n)}_{j,l}(v),$$

and for the limits of the integrals and sums

$$\begin{aligned} i_0 &= \max\{0, r_1 - k + 1\}, & i_1 &= \min\{r_1 + k - 1, n\}, \\ j_0 &= \max\{0, r_2 - l + 1\}, & j_1 &= \min\{r_2 + l - 1, m\}, \\ u_0 &= \max\{u_{r_1}, u_{k-1}\}, & u_1 &= \min\{u_{r_1+k}, u_{n+1}\}, \\ v_0 &= \max\{v_{r_2}, v_{l-1}\}, & v_1 &= \min\{v_{r_2+l}, v_{m+1}\}. \end{aligned}$$

It can be shown that the control point $\widetilde{\mathbf{d}}_{r_1 r_2}$ is an affine combination of the involved control points because

$$\sum_{\substack{i=i_0 \\ (i,j)\neq(r_1,r_2)}}^{i_1} \sum_{j=j_0}^{j_1} \gamma_{ij} = 1. \tag{10}$$

This property (10) can be used to verify the calculation of the integrals. They can be exactly calculated with help of quadrature methods like the Gaussian quadrature. For further information about integrating products of B-splines see [19].

## §5. The Ranking-List

Up to now, we know how to change a control point to get a smoother surface. But which point is the best choice?

In Section 3 we have defined the ranking-number $z_{r_1,r_2}$. Inserting the surfaces (3) and (5) in (6), this (positive) ranking-number (6) can be expressed by

$$z_{r_1,r_2} = (\bar{\mathbf{d}}_{r_1 r_2} - \widetilde{\mathbf{d}}_{r_1 r_2})^2 \cdot \left[ U^{2,2}_{r_1,r_1} V^{0,0}_{r_2,r_2} + 2\, U^{1,1}_{r_1,r_1} V^{1,1}_{r_2,r_2} + U^{0,0}_{r_1,r_1} V^{2,2}_{r_2,r_2} \right], \quad (11)$$

which is a weighted function of the squared change of the control point $\bar{\mathbf{d}}_{r_1 r_2}$. The weights are just the nominator of the factors (9).

Then we want to change the control point with the largest ranking-number. To get the index $(r_1, r_2)$ of this control point we have to calculate the ranking-numbers for all control points which are used. Then this *ranking-list* has to be sorted.

It should be remarked that the whole ranking-list has to be calculated only in the first step for all points which can be changed. The reason is that the control point $\widetilde{\mathbf{d}}_{r_1 r_2}$ influences only the ranking-numbers $z_{i,j}$ with the index $r_1 - k + 1 \le i \le r_1 + k - 1$ and $r_2 - l + 1 \le j \le r_2 + l - 1$ and only this ranking-numbers have to be recalculated in the next step.

In the previous two sections we can see the reason why it is advantagous to change only one control point in every step. The control point is determined by an affine combination of the involved neighbouring control points and the factors (9) are very simple to calculate. But if we want to change not only one but e.g. two control points in every step, then we have to solve (7) for both control points. This leads to a linear system which can be solved explicitly for both control points and also these control points are an affine combination of the involved control points. But the disadvantage is that the representation for the points is much more complicated than (8) and (9). Moreover we have to calculate the ranking-list for all combinations of control point if no property (like neighbourhood) is given.

## §6. Distance Tolerance

It is often necessary to fulfill a prescribed tolerance $\delta$ between the original surface X and the new surface $\widetilde{\mathrm{X}}$, so we have to take care in every step of the constraint

$$\max \left\{ |\mathrm{X}(u,v) - \widetilde{\mathrm{X}}(u,v)| : (u,v) \in [u_{k-1}, u_{n+1}] \times [v_{l-1}, v_{m+1}] \right\} \le \delta. \quad (12)$$

But this leads to a nonlinear problem, and so we use an upper bound for (12) (c.f. [18])

$$|\mathbf{d}_{r_1 r_2} - \widetilde{\mathbf{d}}_{r_1 r_2}| \le \delta. \quad (13)$$

Now, if the point $\widetilde{\mathbf{d}}_{r_1 r_2}$ does not satisfy the constraint (13), we have to look for an alternative position $\widetilde{\mathbf{d}}^*_{r_1 r_2}$ which minimizes the energy functional

(2) under the constraint (13). This new control point is determined by

$$\widetilde{\mathbf{d}}^{*}_{r_1 r_2} = \mathbf{d}_{r_1 r_2} + \delta \frac{\widetilde{\mathbf{d}}_{r_1 r_2} - \mathbf{d}_{r_1 r_2}}{|\widetilde{\mathbf{d}}_{r_1 r_2} - \mathbf{d}_{r_1 r_2}|}. \tag{14}$$

The condition (13) is a good upper bound for (12) if the degree of the spline is small. For more details about the distance tolerance, we refer to [3].

## §7. The Algorithm

With the results developed in the last sections we are now able to build an automatic fairing algorithm for B-spline surfaces:

1) Compute the ranking-numbers $z_{ij}; \alpha_1 \le i \le n - \beta_1, \alpha_2 \le j \le m - \beta_2$.
2) Find $z_{r_1 r_2} = \max\{z_{ij}; \alpha_1 \le i \le n - \beta_1, \alpha_2 \le j \le m - \beta_2\}$.
3) Compute the new location $\widetilde{\mathbf{d}}_{r_1 r_2}$ (resp. $\widetilde{\mathbf{d}}^{*}_{r_1 r_2}$ satisfying the $\delta$-constraint (13)).
4) If a criterion to stop is fulfilled, then exit, else goto step 1.

The first criterion to stop is that the ranking-numbers are less than a certain value. And secondly, we restrict the number of iterations of the fairing algorithm.

The calculation of the ranking-numbers (resp. the ranking-list) is the most time-consuming part of the algorithm. A faster strategy is to change more than one point from the sorted ranking-list. After this, the ranking-list is rebuilt and the process is repeated once again until the criterion to stop is fulfilled.

In the case that the B-spline surface is to be faired by fulfilling a $\delta$-constraint, it is possible that the algorithm *hangs up* at a certain control point, that means this point is always the first one in the ranking-list. So we limit the times each point can be changed during the algorithm.

## §8. Examples

Now we will show three *nonfair* examples to demonstrate the smoothing effect of the algorithm described in the last section. In contrast to the first example which is an academic one, the second and the third surface belong to practical applications.

In the following pictures the *(non-)fairness* is visualized with help of isophotes. Also the measured error $\max_{ij}\{|\mathrm{X}(u_i, v_j) - \widetilde{\mathrm{X}}(u_i, v_j)|\}$ (calculated at 40000 parameter locations $(u_i, v_j)$) is given. All the calculations have been done on an HP 9000/735 workstation.

For the first example presented here we chose $52 \times 52$ control points which lie on the unit sphere and equidistant knot vectors with $k$-fold knots at the boundaries to define the given bi-quartic B-spline surface. Note that the B-spline surface is not a part of the unit sphere. Then 1366 inner control points

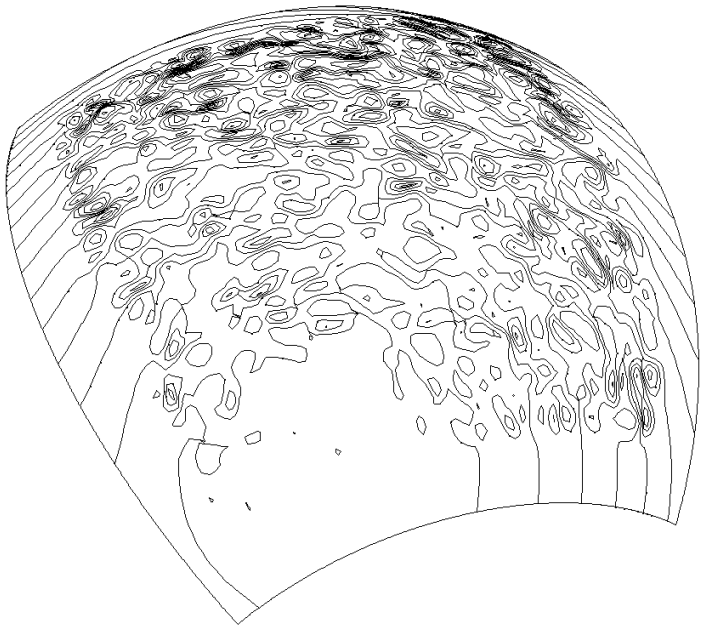

**Fig. 1.** The isophotes of the perturbed B-spline surface.

are perturbed with a maximal deviation of 0.02 with help of random numbers (see Fig. 1). Here a strip of 6 control points on every side is left unchanged.

With a given tolerance of 0.02, the smoothing result is shown in Fig. 2. Here, the control points are changed in 23.73 CPU-seconds and a strip of 5 control points on every side is left unchanged. In this example the measured error between the old and the faired surface is 0.012.

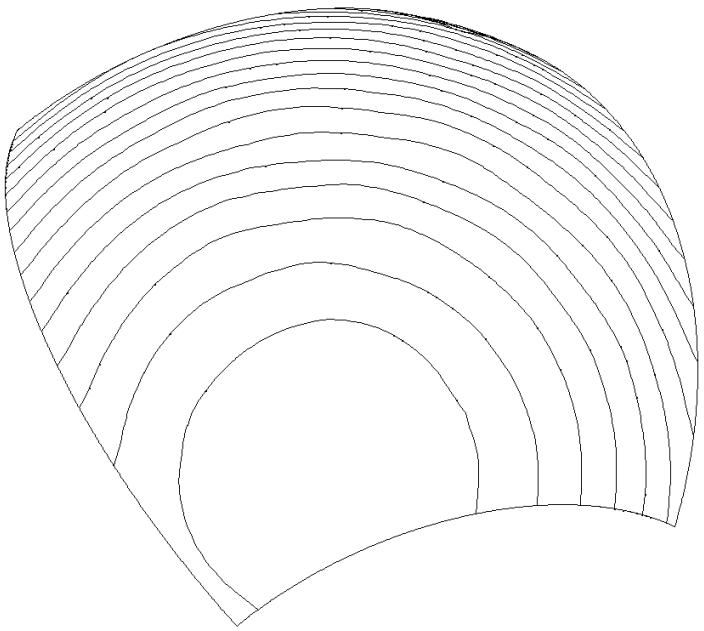

**Fig. 2.** The isophotes after fairing with a tolerance of 0.02.

The second example is a result of a typical *least-square* approximation of a set of points. The bi-cubic B-spline surface has $10 \times 12$ segments and uniform knot vectors with $k$-fold knots at the boundaries. The length of the largest boundary curve is approximatively 350, and the average perpendicular error between the surface and the points is 0.726 (see Fig. 3).

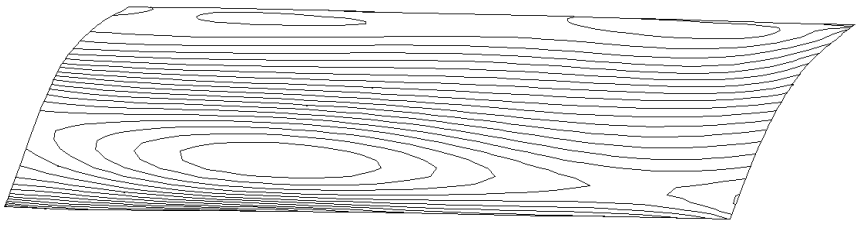

**Fig. 3.** The isophotes of the given bi-cubic B-spline surface.

Fairing with no tolerance and with fixed corner control points, the fairing result is shown in Fig. 4. Here, the algorithm has changed the control points in 0.29 CPU-seconds. The measured error between the old and the faired surface is 2.157 and the average perpendicular error between the faired surface and the data points is 1.469.

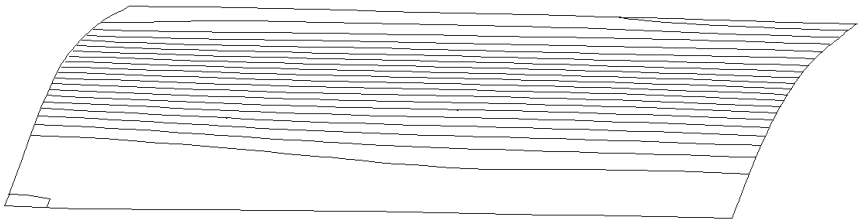

**Fig. 4.** The smooth surface faired without a tolerance.

The third example is a part of a car-body shape. As in the second example a set of points has been approximated with a bi-cubic B-spline surface. The surface has $52 \times 13$ control points and all inner knots have multiplicity $(k-1)$. Then we have removed some inner knots with help of an algorithm described in [4] to get a minimum continuity of $C^1$ at the inner knots (Fig. 5). The length of the largest boundary curve is approximatively 150.

With no tolerance, the smoothing result is shown in Fig. 6. Here, the control points are changed in 0.18 CPU-seconds and only the boundary control points resp. the boundary curves are left unchanged. The measured error was 1.754.

## §9. Conclusion

In the present paper an algorithm is described for fairing B-spline surfaces of arbitrary order. Here we have used the linearized thin plate energy as the fairness criterion. But the developed algorithm is not restricted to this

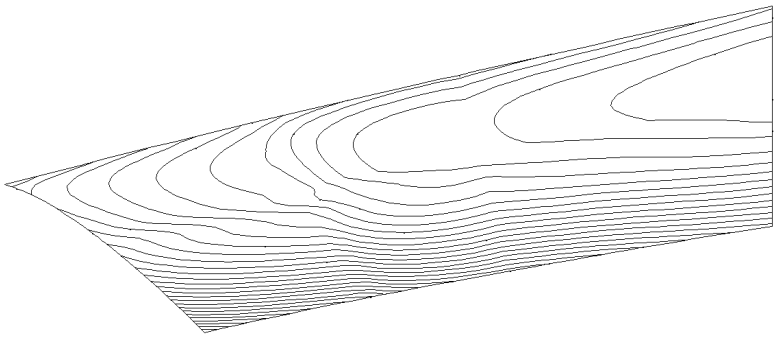

**Fig. 5.** The given B-spline surface.

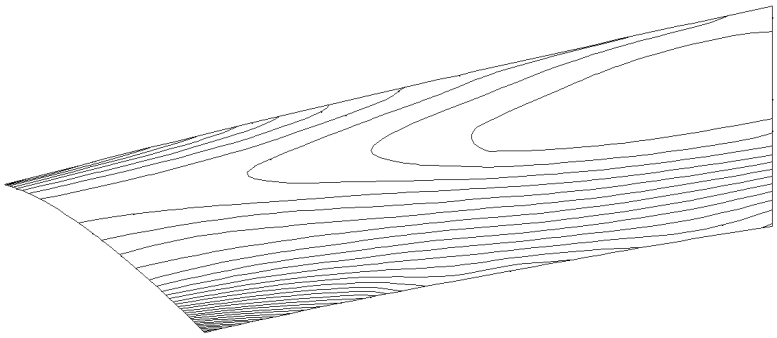

**Fig. 6.** The surface after fairing with no tolerance.

functional, but also other integrals which are a quadratic form of $\mathrm{X}(u, v)$ can be used. Such a functional is e.g. (see [1])

$$H = \iint_A \mathrm{X}_{uuu}^2 + \mathrm{X}_{vvv}^2 \, du \, dv, \tag{15}$$

or a functional suggested by Greiner [8]

$$G = \iint_A (\text{grad div grad X})^2 \, du \, dv. \tag{16}$$

These integrals are not discussed here.

**Acknowledgements.** I would like to thank the *Bundesministerium für Forschung und Technik* for their financel support and also Dr. Sarvajit S. Sinha from *Imageware*, who made it possible for me to visualize the quality of the surfaces with help of their CAD system *Surfacer*.

## References

1. Brunnett, G., H. Hagen, and P. Santarelli, Variational design of curves and surfaces. Surveys on Mathematics for Industry **3** (1993), 1–27.
2. Courant, R., and D. Hilbert, *Methoden der Mathematischen Physik.* Zweite Auflage. Springer 1931.

3. Eck, M., and J. Hadenfeld, Local Energy Fairing of B-Spline Curves. To appear in G. Farin, H. Hagen, and H. Noltemeier (eds.): Computing, Supplementum **10**, Springer (1995).
4. Eck, M., and J. Hadenfeld, A Stepwise Algorithm for Converting B-Splines, in *Curves and Surfaces in Geometric Design*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A K Peters, Wellesley, 1994, 131–138.
5. Farin, G., Degree Reduction Fairing of Cubic B-Spline Curves, in *Geometry Processing for Design and Manufacturing*, R. E. Barnhill (ed.), SIAM (1992), 91–103.
6. Farin, G., and N. Sapidis, Curvature and the Fairness of Curves and Surfaces. IEEE Computer Graphics & Applications **9** (1989), 53–57.
7. Greiner, G., Surface Construction Based on Variational Principles. In P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.): Wavelets, Images, and Surface Fitting, A K Peters, Wellesley (1994), 277–286.
8. Greiner, G., Variational Design and Fairing of Spline Surfaces. Computer Graphics Forum 13:3 (1994), 143–154.
9. Hoschek, J., and D. Lasser, *Fundamentals of Computer Aided Geometric Design*. A K Peters, Wellesley, 1993.
10. Hoschek, J., Smoothing of curves and surfaces. Comput. Aided Geom. Design **2** (1985), 97–105.
11. Kaufmann, E., and R. Klass, Smoothing surfaces using reflection lines for families of splines. Comp. Aided Design **20** (1988), 312–316.
12. Kjellander, J. A. P., Smoothing of cubic parametric splines. Comp. Aided Design **15** (1983), 175–179.
13. Kjellander, J. A. P., Smoothing of parametric surfaces. Comp. Aided Design **15** (1983), 289–293.
14. Klass, R., Correction of local surface irregularities using reflection lines. Comp. Aided Design **12** (1980), 73–77.
15. Moreton, H. P., and C. H. Séquin, Functional Optimization for Fair Surface Design. Computer Graphics **26** (1992), 167–176.
16. Poeschl, T., Detecting surface irregularities using isophotes. Comput. Aided Geom. Design **1** (1984), 163–168.
17. Sapidis, N., and G. Farin, Automatic Fairing Algorithm for B-Spline Curves. Comp. Aided Design **22** (1990), 121–129.
18. Schaback, R., Error estimates for approximations from control nets. Comput. Aided Geom. Design **10** (1993), 57–66.
19. Vermeulen, A. H., R. H. Bartels, and G. R. Heppler, Integrating Products of B-Splines. SIAM Journal of Scientific and Statistical Computing **13** (1992), 1025–1038.

Jan Hadenfeld
Technische Hochschule Darmstadt, Fachbereich Mathematik, AG 3
Schloßgartenstraße 7, D–64289 Darmstadt, GERMANY
hadenfeld@mathematik.th-darmstadt.de