Interpolating and Approximating Scattered 3D-data with Hierarchical Tensor Product B-Splines

Günther Greiner and Kai Hormann

Abstract. In this note we describe surface reconstruction algorithms based on optimization techniques. The input are scattered 3D-data with specified topology. The surfaces constructed are tensor product B-splines. To achieve local detail and/or local fairness we make use of hierarchical tensor product B-splines. Interpolation as well as approximation problems are discussed. The problem of parameterizing a discrete point set which is crucial in this approach will be discussed in more detail.

§1. Introduction

In the last 30 years, many interesting approaches for reconstructing smooth surfaces from discrete scattered data have been presented. In [5, 10] surveys on the classical approaches for interpolating scattered data are presented. More recent approaches are described in [1, 2]. In most approaches, the surface description is in a format not suitable for further processing in CAD/CAM systems. There are few exceptions: the procedures presented in [11] and [1] (see also these proceedings), which are similar to our approach.

In the following we describe an approach how scattered 3D-data can be approximated or interpolated by either a single TP-B-spline patch or by hierarchical TO-B-splines. There is one restriction, the topology of the point set has to be simple, i.e. it must be homeomorphic to a plane set. Thus e.g., complete spheres or tori cannot be reconstructed, without an a priori segmentation. An approach which can handle arbitrary topologies is presented in [2].

The paper is organized as follows. In Sec. 2 we recall some facts on interpolation and approximation. Then, in Sec. 3 we outline the general approach which proceeds in four steps. In the following sections these steps are described in more detail: In Sec. 4 we focus on the parameterization of the discrete point set. How to define an appropriate TP-B-spline space, from which the interpolating/approximating surface will be chosen, will be discussed in Sec. 5. In Sec. 6 we describe fairness functionals and in Sec. 7 we report how the linear systems will be solved. Finally in Sec. 8 we present some examples.

Proceedings of Chamonix 1996
A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), pp. 1–3.

§2. Interpolation and approximation

When reconstructing a "continuous" object starting from a number of discrete "samples" (e.g. measurements) one has basically two choices: interpolation or approximation. Since interpolation is exact at the samples, one might assume that it yields a more accurate reconstruction. However, if the samples carry some kind of noise (e.g. due to measurement errors) approximation will be more adequate. Which one of the basic approaches one should use will depend on the background of the problem.

In both cases we start having a point set $\{P_i\} \subset \mathbb{R}^3$. In order to formulate a well-posed interpolation/approximation problem, one additionally has to specify the *topology* of the point set. That is, one has to specify the neighbors of every point P_i. Usually this is done by specifying a triangulation of the point set, leading to a set of "edges". Fig. 1 indicates, that a point set in general has many topologies, leading to different interpolation (or approximation) problems.

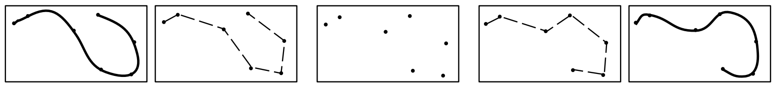

Fig. 1. A point set (middle) with two different topologies .

The problem of interpolation can be formulated as follows:
Given points $\{P_i\} \subset \mathbb{R}^3$ (with a triangulation that is homeomorphic to a plane set), find a smooth, parameterized surface $F : \Omega \to \mathbb{R}^3$ that interpolates all points P_i, preserving the topology. I.e. there are (consistently triangulated) parameter values $Q_i \in \Omega \subseteq \mathbb{R}^2$ such that $F(Q_i) = P_i$.

Moreover, the interpolating surface has to belong to an a priori specified class $\mathcal{S}$ of parametric surfaces.

There is an important special case, the *scalar case*, in which points (x_i, y_i) in the plane and corresponding scalar values z_i are given. A scalar functions $f : \Omega \to \mathbb{R}$ has to be found such that $f(x_i, y_i) = z_i$. The decisive difference between the general and the special case is the following:
In the special case each 3D-data point $(x_i, y_i, z_i) \in \mathbb{R}^3$ has a natural parameter value $Q_i = (x_i, y_i) \in \mathbb{R}^2$. In addition, any triangulation of the plane set $\{Q_i\} \subset \mathbb{R}^2$ describes the (unique) topology. In the general case however, the specification of parameter values to every data point is a major problem. It will be addressed in Section 4.

When formulating an approximation problem, one has to specify, how the distance of the set of sample points $\{P_i\}$ to the surface F will be measured: One can either consider the maximum distance of the samples, or sum up the errors of every sample, or one can consider the least square distance. We only consider the *least square distance* $\sum_i \text{dist}(P_i, F)^2$. Here $\text{dist}(P_i, F)$ denotes the distance of P_i to F. It may be the *parametric distance*: $\|P_i - F(Q_i)\|$ or the *geometric distance* $\inf_{(u,v)} \|P_i - F(u, v)\|$.

The *pure approximation* problem can be formulated as follows:

Given points $\{P_i\} \subset \mathbb{R}^3$ (with a triangulation that is homeomorphic to a plane set), find in a specified class $\mathcal{S}$ of parameterized surfaces the surface $F : \Omega \to \mathbb{R}^3$ that minimizes $\sum_i \text{dist}(P_i, F)^2$.

Since the handling of the geometric distance is very difficult, the parametric distance will be used instead. The determination of the parameter values will be one of the main problems. In addition to being topological correct, one tries to approximate the geometric distance well by carefully choosing or modifying the parameter values.

Besides the *pure approximation*, there is *approximation with fairing.* For this approach a fairing functional J_{fair} has to be specified (see Sec. 6 for examples). The problem then consists of determining the surface $F \in \mathcal{S}$ such that $\sum_i \text{dist}(P_i, F)^2 + \omega J_{fair}(F)$ will be minimal. The positive constant ω somehow measures the tradeoff between fairing and fitting. A larger ω will lead to more fair surfaces but also will cause a bigger approximation error.

There is a close relationship between interpolation and approximation with fairing. The following result can be found in [6]. First we fix the notation:

$$J_0(F) := \sum_i \|P_i - F(Q_i)\| \ , \qquad J_\omega := J_0 + \omega J_{fair} \tag{1}$$

Theorem 1. *Assume that J_{fair} is a quadratic, positive definite functional and let $\mathcal{S}$ be a finite dimensional subspace of parameterized surfaces.*
Let F_ω be the unique minimum of J_ω, i.e. $\quad J_\omega(F_\omega) \le J_\omega(F) \quad$ *for all* $\ F \in \mathcal{S}$.
Then

- $F_0 := \lim_{\omega \to 0} F_\omega$ *exists,*
- $J_0(F_0) = \mathbf{min}_F J_0(F)$ $\qquad$ *and*
- F_0 *is among all minima the one that minimizes* J_{fair}.

We have $J_0(F) = 0$ if and only if F interpolates the data. More general, $J_0(F)$ in some sense measures the failure of interpolation. Minimizing $J_0 + \omega J_{fair}$ yields the same result as minimizing $\frac{1}{\omega} J_0 + J_{fair}$. Thus making ω smaller, the contribution of the error functional J_0 becomes more important.

An immediate consequence of the theorem is the following corollary. It is the basis to a numerical solution of the interpolation problem: the so-called *penalty method* (see [6]).

Corollary. *If in $\mathcal{S}$ there is an interpolating surface, then F_0 is the interpolant in $\mathcal{S}$ with optimal fairness!*

§3. The General Approach

The following basic steps have to be performed:

1) Assign parameter values $Q_i = (u_i, v_i) \in \mathbb{R}^2$ to every data point $P_i \in \mathbb{R}^3$.
2) Choose a rectangle $[a, b] \times [c, d]$ that contains all parameter values. Consider the class $\mathcal{S}$ of bicubic tensor product spline surfaces over $[a, b] \times [c, d]$ by specifying an equidistant rectangular grid $\{(t_j, s_k)\}$ with $t_j - t_{j-1} = \frac{b-a}{n_u} = const$, and $s_k - s_{k-1} = \frac{d-c}{n_v} = const$;

3) Choose a fairness functional $J_{fair} : \mathcal{S} \to \mathbb{R}$;

4_I) Solve the constrained optimization problem
- $J_{fair}(F) = \mathbf{min}$, $F \in \mathcal{S}$ subject to
- $F(u_i, v_i) = P_i$ for all i .

4_A) Solve the unconstrained optimization problem
- $J_0(F) + \omega J_{fair}(F) = \min$, $F \in \mathcal{S}$.

Then the quality of the resulting surface will be checked, i.e., the fairness and the approximation error. If the result is not satisfactory, modifications of steps 1)-4) will be performed and, if necessary, repeated. In the subsequent more detailed discussion of the method, these steps are called iteration steps or refinement steps. More precisely, in each iteration we have to

i) improve the parameter values;
ii) refine the knot spacing (either globally or locally) thus obtaining a larger class $\mathcal{S}$ of TP-spline surfaces;
iii) modify the fairness functional;
iv) solve the (un)constrained optimization problem.

§4. Parameter Selection

To select parameters $Q_i = (u_i, v_i)$ for every data point we use a *spring model.* The idea is the following: We replace each edge of the 3D-triangulation by a spring. The boundary points are projected into a plane and will be fixed there. The remaining vertices, when forced to stay in this plane, arrange themselves in such a way that the spring energy will be minimized. The position of the vertices in this equilibrium is used as an initial parameterization.

More precisely we proceed as follows. Let P_i $(i = 1, \ldots, p_0, p_0 + 1, \ldots, p)$ be the data points and assume that $\{P_1, \ldots, P_{p_0}\}$ are the boundary points of the triangulation. First we determine the plane which fits to all the boundary vertices $\{P_1, \ldots, P_{p_0}\}$ best in the least square sense. Then we orthogonally project P_i $(i = 1, \ldots, p_0)$ into this plane, thus obtaining parameter values Q_i $(i = 1, \ldots, p_0)$ for all the boundary points. The parameter values Q_i for the remaining data points are obtained by minimizing the following functional:

$$\sum_{(i,j)\in \text{edges}} D_{ij} \left(\|Q_i - Q_j\| - L^0_{ij}\right)^2. \tag{2}$$

This functional describes the potential energy of the system of springs that is forced to stay in the plane we obtained by projecting the boundary vertices. D_{ij} represents the spring constant of the spring connecting P_i and P_j and is chosen to be $\frac{1}{\|P_i - P_j\|^r}$ for some $r \geq 0$. $L^0_{ij} = \alpha \|P_i - P_j\|$ is the initial length of that spring. When applied to the problem of reconstructing curves, minimizing this functional will result in well-known parameterizations. For $r = 0$ we will get the *uniform*, for $r = 0.5$ the *centripetal* and for $r = 1$ the *chord length parameterization.* While the parameter α has no effect on the solution in this case, its choice is important in the surface case:

Only for $\alpha = 0$, the functional (2) is quadratic and positive definite in the unknowns Q_i $(i = p_0 + 1, \ldots, p)$. Therefore, to find the minimum, a positive definite linear system has to be solved. This system is also sparse: In each row (representing some Q_k) only those entries are $\neq 0$ that correspond to neighboring vertices of P_k. By choosing different values for r, we will receive equivalents to the parameterizations in the curve case.

Since springs of zero-length are not realistic we also investigated the functional (2) for $\alpha > 0$. The minimum was found by an iterative process similar to the *Gauss-Seidel iteration* for solving linear systems.

Having calculated the Q_i we can now define a rectangular parameter space $R = [a, b] \times [c, d]$ containing all parameter values. This is achieved by determining the bounding boxes of the Q_i $(i = 1, \ldots, p)$ for several orientations (= orthogonal coordinate systems) in the plane and choosing the one having smallest area. Fig. 2 shows the results for a data-set representing a distributor cap: *uniform* refers to $\alpha = 0$, $r = 0$, *chord length* to $\alpha = 0$, $r = 1$ and *real spring* to $\alpha = 0.5$, $r = 1$.

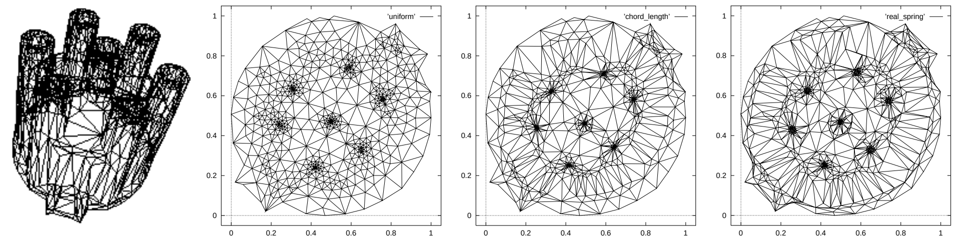

Fig. 2. A triangulated point set with different parameterizations.

When the procedure is iterated the parameters will be improved (step i)). This is done by standard *parameter correction* as described for example in Hoschek's book [8].

§5. Defining an appropriate TP-B-Spline space

We consider a class $\mathcal{S}$ of bicubic TP-B-Splines defined over the rectangle $[a, b] \times [c, d]$ obtained in the previous section. To specify such a set we have to describe the grid of knots. We only consider equidistant knot spacing. This simplifies many of the calculations to be done later on.

Looking for an appropriate knot spacing one has to take into account that there are sufficiently many degrees of freedom. More precisely, if an interpolating surface has to be determined, we have to make sure that interpolation is possible. For interpolation we heuristically start with a knot spacing such that in every 4×4 sub grid of knots there are at most 7 data points (parameter values).[1] Having specified the equidistant grid $\{(t_j, s_k)\}$ the normalized bicubic B-spline basis functions $N_j(u)N_k(v) \quad (-1 \leq j \leq n_u+1,\ -1 \leq k \leq n_v+1)$ span the class $\mathcal{S}$ of all C^2-continuous functions in $[a, b] \times [c, d]$ being piecewise bicubic over every sub-rectangle $[t_j, t_{j+1}] \times [s_k, s_k+1]$. Hereby, $N_j(u)$ and

[1] In case one has to be sure that interpolation is possible, all local interpolation problems (with 4×4) knots are checked whether they are solvable.

$N_k(v)$ denote the (translates) of the univariate cubic B-Spline basis function, centered at t_j and s_k respectively (i.e. the interval $]t_{j-2}, t_{j+2}[$ is the support of $N_j(u)$). Since these basis functions are linearly independent, every $F \in \mathcal{S}$ has a unique representation $F(u,v) = \sum_{j=-1}^{n_u+1} \sum_{k=-1}^{n_v+1} c_{jk} N_j(u) N_k(v)$. The *control points* $c_{jk} \in \mathbb{R}^3$ determine the shape of F.

When the process of surface reconstruction is iterated (step ii)), a simple strategy for selecting a new class of B-splines is to half the grid size. This *global refinement* increases the dimension of the problem by a factor of 4. A better strategy is a *local refinement*, based on the hierarchical B-splines introduced by Forsey and Bartels (see [3, 4]).

In fact, when the reconstructed surface is examined, in general, in some parts of the domain everything will be fine, i.e. the approximation error will be below the tolerance and the surface is sufficiently fair. Only in some critical areas either the approximation error is above the tolerance or the quality of the surface is poor. Only in these parts we need additional degrees of freedom for better fitting and/or fairing.

This is achieved as follows: First, the critical areas (grid rectangles) are marked. Then we consider TP-B-splines surfaces having the following representation:

$$F(u,v) = F_{old}(u,v) + \sum_{j,k} \tilde{c}_{jk} \tilde{N}_j(u) \tilde{N}_k(v)$$

where $\tilde{N}_*$ denote the basis function corresponding to half the grid size and the summation is only over indices j,k for which, the support of $\tilde{N}_j(u)\tilde{N}_k(v)$ meets the marked area.

§6. Fairness Functionals

When considering fairness functionals, one has to find a compromise between low numerical complexity and high quality of the resulting surfaces. Low complexity is achieved by using quadratic functionals. High quality can be obtained by using good approximations to the curvature functionals.

For example, one has low numerical complexity for the *simple thin plate energy* functional given by

$$J_{simple}(F) = \int_\Omega {F_{uu}}^2 + 2{F_{uv}}^2 + {F_{vv}}^2 \, dudv \tag{3}$$

and one gets surfaces of high quality when using the *exact thin plate energy* (see [9]) (based on the principle curvatures κ_{F_i}) given by

$$J_{exact}(F) = \int_\Omega \kappa_{F_1}^2 + \kappa_{F_2}^2 \, d\omega_F \tag{4}$$

A compromise between these to extreme cases is the *data dependent thin plate energy* introduced in [7].

$$J_S(F) = \int_\Omega \mathrm{trace}(\mathbf{Hess}_S(F)^2) \, d\omega_S \tag{5}$$

Let us briefly recall the basic ideas, definitions and properties:
The basic object is a *reference surface* S, that roughly represents the geometry of the surface F to be constructed by the (constrained) optimization process. We consider F being defined over S. In other words, S serves as a non-planar parameter space for F. One can differentiate functions defined on S. The second order derivative of F with respect to S (the Euclidean analogue of the Hessian), is denoted by $\mathbf{Hess}_S(F)$. The integrand in (5) is the trace of the square of the Hessian. The functional J_S defined by (5) is quadratic and (nearly) positive definite.

The relation to (3) and (4) is the following: When the usual Euclidean Hessian is used in (5) (i.e. S is planar), we are back to (3). While when F itself serves as reference surface, then (5) and (4) coincide (for details see [7]).

J_S will be a good functional, provided that S roughly represents the geometry of F. More precisely, S must allow a parameterization of F with only little metric distortion. We try to achieve this in the following way.
In the initial step: The grid points lying within the triangulation of the parameter values Q_i are transformed by linear interpolation to 3D-space. For the grid points not lying in the triangulation extrapolation is used instead. The situation is outlined in Figure 3.

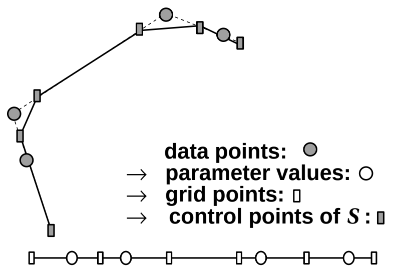

Fig. 3. Construction of the initial reference surface $\mathcal{S}$ (1D analogue).

In the iteration steps: The surface of the previous step serves as the reference surface in every refinement step.

§7. Solving the optimization problem

The fairness functional J we use is quadratic and positive semi-definite hence $J_{fair}(F) = \langle F|F\rangle_J$ for a suitable positive semi-definite, symmetric, bilinear form $\langle\cdot|\cdot\rangle_J$.

7.1. Lagrangian multipliers When the interpolation problem has to be solved, the constrained optimization problem has to be solved exactly. We achieve this by introducing Lagrangian multipliers, one λ_i for every interpolation condition. Solving

- $J_{fair}(F) = \mathbf{min},\ F = \sum_{\mu=1}^{M} c_\mu N_\mu$ subject to
- $F(u_i, v_i) = P_i$ for all i

then leads to a linear system in block matrix form.

$$\begin{pmatrix} A_0 & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} c \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ P \end{pmatrix} \tag{6}$$

For simplicity, we omit double indices ${}_{jk}$ for the basis functions and control points resp. Instead we use the single index ${}_{\mu}$, $\mu = 1, ..., M = (n_u+3)(n_v+3)$. A_0 is an $M \times M$-matrix, B is an $p \times M$-matrix given by

$$A_0 = (\langle N_\mu | N_\nu \rangle_J)_{1\le\mu,\nu\le M} \text{ , and } \ B = (N_\mu(u_i, v_i))_{1\le i\le p, 1\le\mu\le M}$$

The unknowns are $c = (c_\mu)_{1\le\mu\le M}$ and $\lambda = (\lambda_i)_{1\le i\le p}$. On the right hand side the data points $P = (P_i)_{1\le i\le p}$ appear.

The normal equation (6) is solved iteratively by a block Gauss-Seidel method. The unknowns (c_μ and λ_i) are grouped in such a way that in each step a local interpolation problem over a sub grid of size 4×4 has to be solved. That is, each subproblem has the following unknowns: 4×4 control points c_μ plus the λ_i's corresponding to interpolation conditions lying in the sub grid.

7.2 Penalty method Often exact interpolation is not necessary and approximation will be sufficient. In these cases the penalty method is more appropriate. In the initial step we consider the functional $J_\omega(F) = J_0(F) + \omega J_S(F)$, with J_0 being defined in (1), J_S the data dependent thin plate energy defined in (5). The penalty parameter ω typically is chosen between 0.1 and 0.01. If there are at least three data points, J_ω is positive definite (see [7]), hence has a unique minimum, call it F_ω. Its control points $\{c_\mu\}$ are obtained by solving the linear system

$$(A_0 + \omega B^t B)c = B^t P$$

with A_0, B and P as defined above. Depending on the size (which is $M \times M$), this linear system is solved either by Cholesky method or by cg-method.

In each iteration step of the whole reconstruction process (consisting of: parameter correction — (local) refinement of the space $\mathcal{S}$ — selection of the new reference surface S — solving the linear system) the penalty parameter ω is decreased by the factor 0.1.

§8. Examples and Conclusion

We conclude with a selection of examples. In all our examples we use bicubic TP-B-splines defined over the unit square with equidistant knot spacing. We start our iterative approximation process with a TP-B-spline having 6, 8 or 12 control points in either direction, depending on the number of data points.

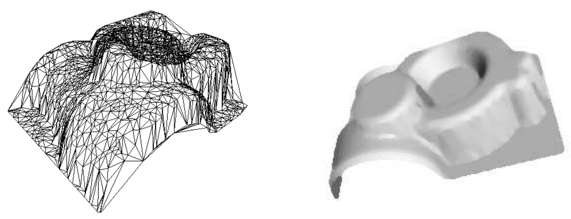

Fig. 4. A triangulated data-set (left) and a smooth approximation (right).

Fig. 4 shows the result if we use the *real spring parameterization* for the parameter values, the *data dependent thin plate energy* as fairness functional and the penalty method with an initial $\omega = 0.05$, two iterations and global refinement.

The effect of different fairness functionals is shown in Fig. 5, where we solved the constrained optimization problem to obtain interpolating surfaces. While the use of the *simple thin plate energy* leads to a surface with lots of wrinkles, the result becomes smooth with the *data dependent* version of this fairness functional. In this case, the reference surface was a least square fit to the data.

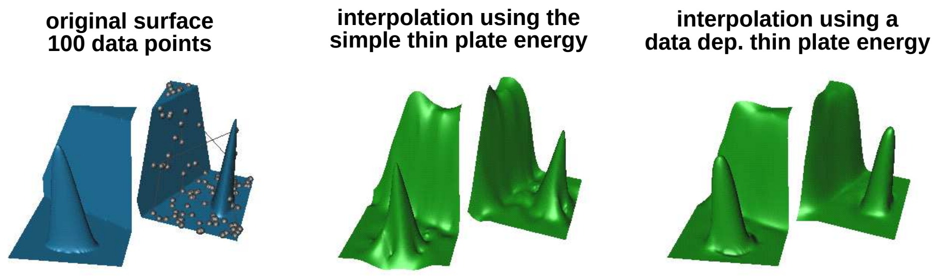

Fig. 5. Reconstruction using different fairness functionals.

After solving the optimization problem we calculate the approximation errors to the data points. Patches containing data with an approximation error $\geq 0.1\%$ will be locally refined, taking into account that neighboring patches are grouped to rectangular regions. If more than 50% of the parameter space are to be refined locally, we advise global refinement instead. Fig. 6 shows the result of such an approximation process.

To sum up, one can say that the use of hierarchical tensor product B-splines leads to a promising approach to reconstruct smooth surfaces. Combined with the techniques of finding parameter values for the data points representing the geometric arrangement of the triangulated data, it leads to surfaces having a low number of control points due to the fact that the degrees of freedom are concentrated at the problematic regions. The use of data dependent fairness functionals results in smooth surfaces and the penalty method allows the user to control the tradeoff between fairness and approximation.

References

1. Dietz, U. and J. Hoschek, Smooth B-spline surface approximation to scattered data, in *Reverse Engineering*, J. Hoschek and W. Dankwort (eds.), B. G. Teubner, Stuttgart, 1996, 143–151
2. Eck, M. and H. Hoppe, Automatic reconstruction of B-spline surfaces of arbitrary topological type, ACM Computer Graphics (SIGGRAPH 1996), 325–334
3. Forsey, D. R. and R. H. Bartels, Hierarchical B-spline refinement, ACM Computer Graphics **22**(4), 1988, 205–212

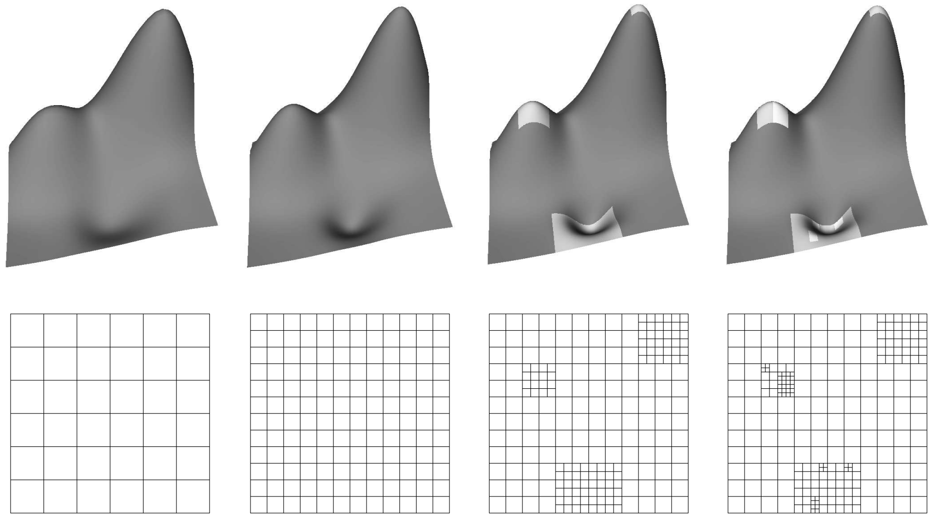

Fig. 6. Approximation with one global and two local refinement steps.

4. Forsey, D. R. and R. H. Bartels, Surface fitting with hierarchical splines, ACM Trans. on Graphics **14**, 1995, 134–161

5. Franke, R. and G. M. Nielson, Scattered data interpolation and applications: A tutorial and survey, in *Geometric Modeling*, H. Hagen and D. Roller (eds.), Springer, Berlin, 1991, 131–161

6. von Golitschek, M. and L. L. Schumaker, Data fitting by penalized least squares, in *Algorithms for approximation II*, John C. Mason (ed.), Shrivenham, 1988, 210–227.

7. Greiner, G., Loos, J. and W. Wesselink, Surface modeling with data dependent energy functionals, Comp. Graphics Forum, **15**(3), 1996, 175–186

8. Hoschek, J. and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, AKPeters, Wellesley, 1994

9. Moreton, H. P. and C. H. Séquin, Functional optimization for fair surface design, ACM Computer Graphics (SIGGRAPH 1994), 167–176

10. Schumaker, L. L., Fitting surfaces to scattered data, in *Approximation Theory II*, G. G. Lorentz, C. K. Chui and L. L. Schumaker (eds.), Academic Press, 1976, 203–268

11. Schumaker, L. L. and G. E. Fasshauer, Minimal energy surfaces using parametric splines, Comput. Aided Geom. Design **13**(1), 1996, 45–79

G. Greiner, K. Hormann,
Graphische Datenverarbeitung,
Universität Erlangen–Nürnberg,
Am Weichselgarten 9,
91058 Erlangen,
Germany
$\left.\begin{array}{l}\texttt{greiner}\\ \texttt{kihorman}\end{array}\right\}$`@informatik.uni-erlangen.de`