ACM Symposium on Solid Modeling and Applications (2004)
G. Elber, N. Patrikalakis, P. Brunet (Editors)

Constraint-based Design of B-spline Surfaces from Curves

P. Michalik[1] and B. D. Brüderlin[2]

[1]R&D Dept., Tebis AG, Munich, Germany
[2]Computer Graphics Group, Dept. of Computer Science Technical University of Ilmenau, Germany

Abstract

In this paper we describe the design of B-spline surface models by means of curves and tangency conditions. The intended application is the conceptual constraint-driven design of surfaces from hand-sketched curves. The solving of generalized curve surface constraints means to find the control points of the surface from one or several curves, incident on the surface, and possibly additional tangency and smoothness conditions. This is accomplished by solving large, and generally under-constrained, and badly conditioned linear systems of equations. For this class of linear systems, no unique solution exists and straight forward methods such as Gaussian elimination, QR-decomposition, or even blindly applied Singular Value Decomposition (SVD) will fail. We propose to use regularization approaches, based on the so-called L-curve. The L-curve, which can be seen as a numerical high frequency filter, helps to determine the regularization parameter such that a numerically stable solution is obtained. Additional smoothness conditions are defined for the surface to filter out aliasing artifacts, which are due to the discrete structure of the piece-wise polynomial structure of the B-spline surface. This leads to a constrained optimization problem, which is solved by Modified Truncated SVD: a L-curve based regularization algorithm which takes into account a user defined smoothing constraint.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Splines; G.1.2 [Approximation]: Spline and piecewise polynomial approximation; G.1.3 [Numerical Linear Algebra]: Linear systems (direct and iterative methods)

1. Introduction

B-spline curves and surface have been recognized as an important representation for free-form shapes early on, in [GR74]. It is our observation that current free-form modeling tools are still far away from generating sculpted shapes with desired properties by few pen-strokes or mouse-clicks: the creation of complex free-form surface models is a time-consuming and tedious procedure which requires a knowledge about the underlying curve and surface representation. We claim, that in the context of surface modeling the power of constraint-based methods have not yet been fully exploited. This fact and the experience we have gained from investigating methods for interactive manipulations of solids with constraints for planar and analytic surfaces [BDKM00, DMB98] has led to the idea to extend this approach to free-form surface models: The designer creates or modifies B-spline surfaces by sketching one or several 3D curves. These curves are seen as design parameters which determine the shapes of the surfaces. The designer is free to further modify these curves by re-styling their shapes, add new curves, or, delete the existing ones. The curves are automatically constrained as incident on the surface which leads to a set of constraints determining the surface.

In our previous paper [MKB02] we have investigated methods for correct and intuitive interpretation of hand sketched 3D curves. In this paper, which can be seen as a continuation of our previous work, we are concerned with two fundamental issues:

1. Techniques to manage networks consisting of B-spline surfaces, curves and constraints among them.
2. Interpolation of B-spline surfaces from curve-surface incidence and tangency constraints for arbitrary (non-isoparametric) curves.

The curve-surface incidence constraints are expressed as linear systems of equations between *degrees of freedom* of the surface and the design parameters – the control points of sketched curves. How to obtain these equation efficiently and with high numerical precision was described in [Mic03]. Our current paper contains further details about the user interaction and solving of inverse problems which were not mentioned in previous publications [MB99, MB00, MB01]. The equation systems are, in general, under-determined and ill-conditioned. Hence, special numerical techniques are necessary to compute a numerically stable solution. We present a powerful and reliable method which automatically extracts an optically pleasing surface from an ill-conditioned and under-determined system of linear equations. A surface exactly satisfying given curve constraints does not always exist – in this case designers are guided by the system towards a compromise between their intention and the satisfaction of constraints.

1.1. Structure of the paper

The remainder of this paper is divided into five sections organized as follows: section 2 reviews the related research and points out the differences between our work and previously achieved results. In section 3 we describe the concept of an interactive, constraint-driven surface modeler. Section 4 describes the interpolation of a B-spline surface from non iso-parametric curves. The theory is rounded off by several design examples. Finally, section 5 summarizes the research achievements of this work and compiles an outlook on future research.

1.2. Notation

We shall deal with geometric elements such as surfaces, curves or points. These are denoted by italic upper case letters, S, C or P. Our central tool is linear algebra of matrices and vectors: Bold upper case letters **A** denote matrices and bold lower case letters **a** denote vectors.

2. Related work

The problem of defining surfaces by incident curves can be expressed as a system of linear equations between *degrees of freedom* (DOF) of the surface and the design parameters – the control points of 3D curves. Assume that a parameterization of a B-spline surface F and a curve G in the domain of F are given and fixed. Further assume that a 3D curve H' (which serves as a shape parameter) is mapped to a surface curve $H = F(G)$. Then the control points of the surface which satisfies $H' = F(G)$ can be determined as a solution of a system of linear equations.

The B-spline surface interpolation is a standard procedure in CAGD if H is an iso-parametric line of F, i.e. $G = const$. The area of constraint-based surface design is dominated by surface reconstruction from a compatible network of iso-parametric curves (the so-called surface "skinning"). Also, relatively well understood are, so-called, *multi-patch methods* and *the scattered data interpolation*. There is a huge amount of literature which deals with these topics, therefore, we point only to standard books on CAGD [Far92, HL89, PT95] and to Jörg Peter's survey of fundamental methods for multi-patch surface design [Pet01].

If G is a general domain curve, the problems are two-fold: first, it is not obvious how to formulate the equations which constrain the incidence of an arbitrary curve. Second, in contrast to traditional methods, usually, the constraints do not uniquely determine the surface – the problem is under-determined. In addition, it turns out that interpolation problems of this type belong to the category of, so-called, *ill-posed* problems [Han98]. These problems are characterized by the property that the equation system is not simply singular: there is no clear distinction between linearly dependent and independent subsets of equations and it is very hard to determine the rank of the system matrix. Solving of such problems is not possible without sophisticated numerical analysis tools.

2.1. Obtaining the equations

The first problem – obtaining the equations – was approached in three different ways. In [CW92] and [WW92], Welch et al. propose to formulate such constraints as a continuous approximation problem: given F, G and H' as above a linear system of equations in the unknown control points of the surface is obtained by minimizing the quadratic distance functional $\int_G (H' - F(G))^2$ which yields a square matrix of equations linear in control points of F. The second approach uses the fact that a 2D curve G is mapped to a 3D surface curve H incident on F by taking linear combinations of Fs control points. In other words, each control point of H can be written as a linear combination of control points of F yielding a linear system of equations in unknown control points of F. In [EC97] Elber and Cohen have shown how to obtain the equations for the special case when F is a Bezier patch and G is one or several (arbitrarily oriented) line segments. Similar approach was chosen by Ferguson and Grandine in [FG90]: They have already pointed out that using line segments might not be sufficient and a generalization to arbitrary curves is desired. The third approach, pursued by Dietz in [Die98], for example, is the discretization method: the continuous curve-surface incidence problem is discretized by considering many point-surface constraints ordered along given 3D curve.

We have generalized the second method to B-spline surfaces and arbitrary B-spline domain curves in [Mic03] and [MB99, MB00, MB01]. Briefly, we have introduced the, so-called, "unevaluated" version of blossom based polynomial composition algorithm, the original (unevaluated) version of which was described be DeRose et.al in [DGHM93]. Given G and parametric structure of F, we compute a matrix **A** with the property that the surface control points of the curve $H = F(G)$ are obtained as a linear transformation of the control points of F. This yields an matrix equation $\mathbf{H} = \mathbf{AF}$ where **H** and **F** are matrices containing the control points of the curve and the surface, respectively.

The disadvantage of the "continuous approximation" approach is that the computation of the associated matrices is inefficient and numerically not very stable (esp. because of the integration of high degree splines). Furthermore, the system matrix is obtained by formulating the normal equations. This is not the optimal approach: it is known, that the condition number of normal equations is the square of the actual condition of the problem [GvL89]. The unevaluated composition (if implemented carefully) is more efficient and numerically much more stable. It delivers a matrix **A** with smaller overall errors it its elements and, in general, of smaller size. Furthermore the matrix equation $\mathbf{H} = \mathbf{AF}$ can be easily extended to define other linear curve-surface constraints; e.g., in [Mic03] we describe an efficient method how to constrain normals along an arbitrary surface curve.

2.2. Obtaining the solution

The second part of the problem is to compute the control points of the surface **F** given the linear system of equations $\mathbf{AF} = \mathbf{H}$. The equation system is generally *ill-posed*. In case of an ill-posed problem we have $k = rank(\mathbf{A}) < \min(m,n)$ but the singularity is not easy to detect. This is the fundamental difference to problems which are "only" singular: there is no obvious selection for the "right" k. Methods for solving such problems are known in linear algebra, see e.g. [Han98] for further backgrounds and a profound overview on *regularization* methods. We have applied the so-called "L-curve" filter [Han01, CHR01], in connection with Singular Value Decomposition (SVD) of the system matrix. The SVD is a superior tool for solving of singular systems of linear equations. However, in its usual setting, it is not suited to solve an ill-posed problem because even the explicit knowledge of singular values of **A** is not sufficient to determine the rank of **A**. This information is central in order to obtain a satisfactory solution. We briefly summarize the SVD and its properties along with a more precise definition of ill-posedness in section 4. The regularization algorithms we have exploited are the Truncated Singular Value Decomposition (TSVD) [Han90] and Modified Truncated SVD (MTSVD) [HSS92] algorithms. The MTSVD algorithm is perfectly suited to solve ill-posed constrained optimization problems. Problems of this kind are the mathematical core of the variational surface design which was pursued by many researchers, see [BHS93, BHH93, Gre94, KPS95, GS97, SU90], for example. The basic idea is to compute the surface as a minimum of some application dependent objective function with respect to specified boundary constraints. However, except of Schumaker and Utretas in [SU90] the ill-posedness of the constraints is never mentioned. Indeed, almost all publications deal with well-conditioned constraints – such as incidence of iso-parametric curves or equally distributed points on a surface. Nevertheless, variational constraints are well suited to determine values of those control points of the surface which are not consumed by the positional constraints – in our case the curve-surface incidence or tangency along a curve. In that case one chooses the objective function such that certain differential properties of the surface are minimized. There is a standard set of candidate functions which in most cases deliver satisfactory results (i.e. optically pleasing or smooth surfaces): one minimizes thin plate energy of the surface, variation of curvature or similar properties. An overview of frequently used objective functions and their properties can be found in [GS97], for example.

Celniker [WW92, CW92], Dietz [Die98] and Ferguson [FG90] have used variational methods in order to completely determine the control points of the surface. However, non of the publications have accessed the problem of obtaining a numerically stable solution from the ill-posed equations. Although Celniker and Welch have recognized that the linear systems of equations may be ill-conditioned, it is not clear how they remedy the ill-conditioning. They propose the usage of pivoted Gaussian elimination. However, in most cases Gaussian elimination is not sufficient to reveal the rank of the matrix, see e.g. [Han98]. We have carried out numerous experiments which confirm this statement. In [EC97], Elber and Cohein have used Bezier patches with carefully chosen number of degrees of freedom. In that case the ill-posedness is not that serious problem, especially, when the domain curves do not differ much from iso-parametric lines. However, this method results in surface patches of very high degree. Ferguson and Grandine have mentioned that computing solutions to such linear systems can be difficult. Although they have have used the Singular Value Decomposition of the system matrix, they do not mention how to select the rank of the badly conditioned matrix.

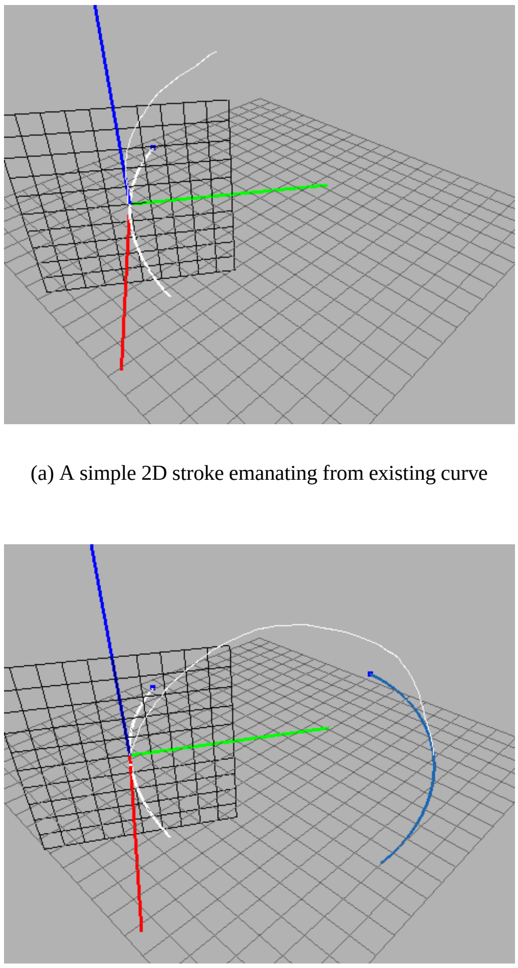
3. Constraint-based surface sketching

In the following we present our view of a constraint-based surface modeling system. First, we show how we have interpreted free-hand sketches in 3D to obtain B-spline curves. We do that only briefly because this topic was described in our 2002 paper [MKB02]. Second, we demonstrate on an example how the constraints are managed and how the system reacts to user interaction.

3.1. Interpretation of sketch strokes in 3D

As a starting point, the system provides one default planar surface. The initial pen stroke is projected into that plane and approximated by a planar B-spline curve. The system uses interference of subsequent pen-strokes with existing curves to embed the sketch into an auxiliary projection plane: assume that the user continues with another pen stroke starting at (or close to) a previously sketched curve. Here, a pen-stroke is a set of ordered points in the current image plane. At this stage, the system recognizes two alternatives to compute the projection plane:

1. The stroke emanates from an existing curve and ends in the "air" (no other curve was hit). The 3D curve is projected to current image plane. The parameter value corresponding to the closest point on that projected curve is found and the projection plane is computed from point and tangent at that point, see figure 1(a).
2. If the stroke connects two existing curves, see figure 1(b), two curve points and the bi-normal at the first curve point are used to set up the projection plane.



(a) A simple 2D stroke emanating from existing curve

(b) A 3D stroke connecting two curves

Figure 1: *Interpretation of pen-strokes in 3D*

Proceeding this way one or several 3D curves can be obtained. The interpretation of the curve data depends on the selected *design mode*:

Sweeping mode: whenever the input stroke emanates at a planar curve, the resulting curve is interpreted as an "axis-curve" for sweeping the curve the stroke emanates at.

Interpolation mode: the 3D curves are interpreted as input curves for the surface interpolation. For each curve an incidence constraint is automatically generated and the surface which interpolates sketched 3D curves is computed.

Sculpting mode: The user draws or selects one curve incident on an existing surface. Other curves are either "fixed" (their shape and position must not change) – or "free" (their shape or position are allowed to change). The selected curve is sculpted by further pen-strokes inside of an auxiliary "orthogonal sketch space" – an auxiliary surface which is locally orthogonal to the sculpted surface along the original curve,see [MKB02] for details.

Assuming that surface interpolation from arbitrary curves is available, the former two modes are relatively straightforward. The sculpting mode, however, requires additional effort to manage the information about the current "state" of the curve and surface data. In the following paragraph we describe a concept of a constraint-management for this kind of modeling.

3.2. Constraint-based surface sculpting

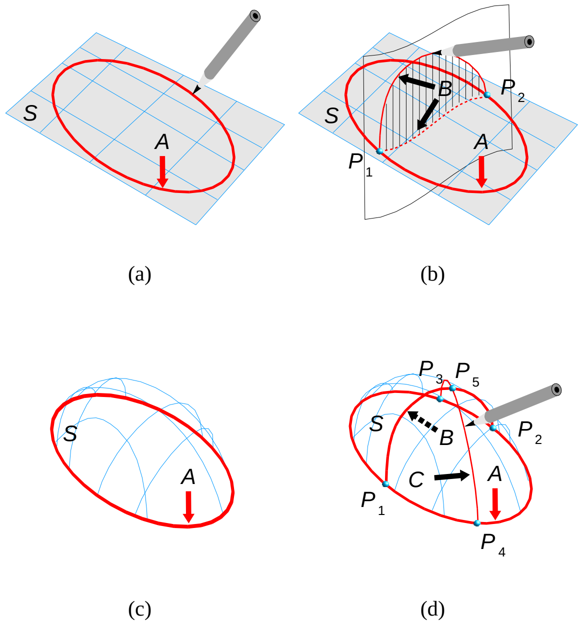


Figure 2: *Schematic AI demonstration of using the modeler in "sculpting" mode*

Assume, the user wishes to create a dome shaped surface as shown in figures 2(a)-(d). The user sketches the closed curve A first. The curve is projected into the parametric space of the selected surface S which yields a pre-image of A in the domain of the surface. A curve-incidence constraint is automatically generated. If desired, the region outside the closed curve is trimmed from the surface S. Since we want to preserve the shape of A it is marked as "fixed". Now, in order to bulge out the surface, the curve B is drawn, fig. 2(b). The sketch is first interpreted in an auxiliary plane (see §3.1) and then projected onto the surface. Then, the curve is adjusted to be incident in appropriate sketching space determined as described in §3.1. In our example the sketch space is denoted by the hatched region in fig. 2(b). The pre-image of B is checked for interference with other elements. In this case, B intersects with A at points P_1 and P_2. It is necessary to capture these conditions by additional constraints. Here, 4 point-curve incidence constraints among both curves and points are recognized and generated automatically which yields a constraint-graph shown in fig. 3(a).

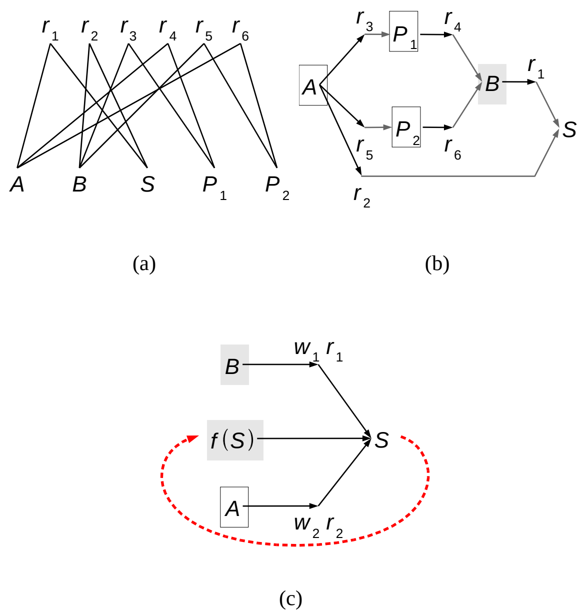


Figure 3: *Constraint-graph and construction plan for the cap surface example.*

Whenever a sketching transaction is finished, the system offers a shape for surface S, see fig. 2(c). The order of evaluation is determined from current distribution of degrees of freedom (DOF) in the constraint graph: in our example, the "construction plan" is as shown in fig. 3(b): the curve A is fixed which enforces fixed points P_1 and P_2. These consistency constraints consume 2 DOF from B – thus, the dependent DOF of B must remain fixed. Every sketch needs to be checked against these conditions. This implies that there must be a two-way communication between the constraint solver and the sketching system: at the time when the sketch is interpreted, only this reduced number of DOF may be considered.

Finally, the surface is determined from curves A and B. The resulting shape of S will depend on the structure of its B-spline spaces and, of course, on the quality of the applied smoothing constraint, see below. One possibility is to start with relatively simple surface and insert new control points at appropriate positions if errors at the constraints exceed defined limits. However it is not yet clear at what positions the new control points should be inserted in order to reduce the errors in the continuous curve constraints. The error for each constraint is measured by back-substitution of the pre-image curve into S and evaluation of the difference among this exactly incident curve and the curve present in the model. In the following $|r_i(X,Y)|$ will denote this error for a specific constraint. At the last stage of the construction plan, shown in figure 3(c), the surface is determined (possibly in several iterations) as a solution of a constrained variational problem

$$\min f(S) \quad \text{subject to} \quad |r_1(A,S)| = 0 \wedge |r_2(B,S)| = 0 \qquad (1)$$

where $f(S)$ is a smoothness constraint which regulates the overall shape of the surface. Here, the objective function $f(S)$ is selected as described in section 2.

The so determined construction plan stays valid as long as the user does not change the distribution of DOF (i.e. does not fix or free an another element). A new plan needs to be evaluated if elements are created and inserted into the model, or if existing elements are removed. This is illustrated in fig. 2(d). Another curve, C, is introduced. New consistency conditions arise, if the user decides to keep the shape of B: since B intersects with C in P_5 and A remains fixed, the number of Cs DOF is a-priori reduced by 3.

In summary we note: the model is represented by a constraint-graph which is updated depending on user's design actions. The construction plan is determined after an element of the graph (curve, surface, or point) has undergone a change of the state (i.e., the user has marked an element as one of "fixed", "free" or "changed"). For analytic and planar surfaces algorithms to obtain a construction plan from "steady state" constraint graph have been proposed, see e.g. [BDKM00, DMB98]. However, these algorithms rely on the a-priori knowledge of how many degrees of freedom of an object are consumed by a specific constraint. Since this is impossible to predict in case of ill-posed linear problems (see section 4), these methods have to be modified or completely re-evaluated; this is subject to future work.

4. Surface interpolation from general curves

The central operation required in the above example is the constrained approximation of a surface given one or several curves. In the following we will explain how to obtain a numerically stable and optically pleasing surface which minimizes errors at curve constraints.

4.1. Demonstrating the ill-posedness

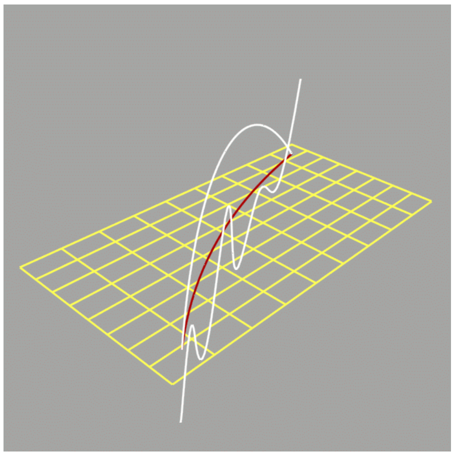
The equation system is set up as noted in section 2. Thus one curve constraint (or a concatenation of several curve constraints for one surface) result in an linear system of equations

$$\mathbf{H} = \mathbf{A}\mathbf{F} \qquad (2)$$

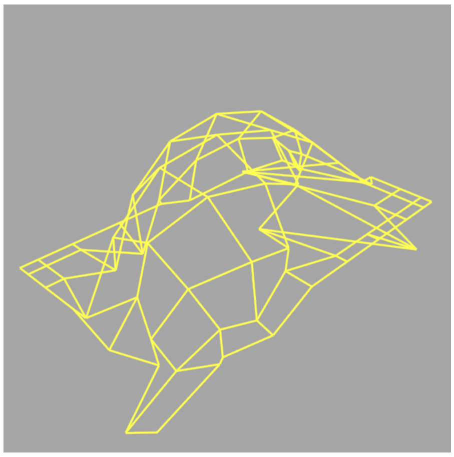
The matrix $\mathbf{H}$ contains the control points of constrained curves and $\mathbf{F}$ denotes matrix of the unknown control points of the surface F. Given one or several 3D curves we look for a surface F such that 2 is satisfied.

In order to demonstrate the problems one encounters when solving the system 2 consider figure 4: Let be given a bi-cubic B-spline surface with one curve constraint and two sketched "request" curves. The surface has 8×10 control points, the domain curve G is a cubic B-spline with 8 control points. The exact surface curve $H = F(G)$ has degree 12 and 102 control points. Clearly, this highly overdetermined curve is not well suited for interactive editing. In [Mic03] and [MKB02] we have proposed to maintain a level of detail representation between the sketched curve and the exact curve: briefly, we compute "compatibility" matrices which perform the necessary degree elevation and knot insertion on the request curve yielding a linear system with compatible number of rows on both sides.

Now we turn to solving the inverse problem in eq. 2. The system exhibits the typical features of an ill-posed problem:



(a)



(b)

Figure 4: *Demonstration of the ill-posedness for a single curve constraint on a bi-cubic B-spline surface with* 8×10 *control points. (a): Initial shape of surface, the original shape of the curve H (dark). Two request curves are shown in bright color: a rapidly varying curve and a smooth curve. (b): The unregularized solution. Although the surface satisfies the constraint – the smooth bright curve from figure (a) – from the shown control mesh we conclude that this surface is useless.*

- *The condition number is high*. The condition of a linear equation system is expressed as amount of change in the normed solution $\|\mathbf{F}\|$ in dependence of the changes in parameters $\mathbf{H}$, see [TB97]Part III, for example. Figure 4(b) demonstrates that: even a small deformation of the constrained curve, in this case, shown as the smoother bright curve in figure 4(a), causes randomly occurring "wiggles" in the control mesh of the surface which is shown in 4(b).
- *The rank of the system is a '"noisy" number.* The first property does not necessarily imply that a linear system is ill-posed. The characteristic property of an ill-posed problem is that there is no obvious choice for the rank of the system matrix. For example: when applying pivoted Gaussian elimination one will not encounter a pivot which is particularly smaller than the pivots in previous elimination steps. The values of pivots gradually decay to a small number; hence, there is no obvious criterion when to stop the elimination.

The ill-posedness of the linear system $\mathbf{AF} = \mathbf{H}$ is caused by nearly linearly dependent rows (and columns) in $\mathbf{A}$; in other words, the chosen linear model, here obtained from the composition of the tensor-product B-splines spaces of F and the domain curve G, is not able to safely determine the influence of the surface curve H on the control point of F.

4.2. The L-curve regularization

The superior tool for analysis of ill-posed problems is the Singular Value Decomposition (SVD). The SVD of system matrix $\mathbf{A}$ has the format $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$. The matrices $\mathbf{U},\mathbf{V}$ are orthonormal and Σ is a diagonal matrix such that $\sigma_i \geq \sigma_{i+1} \ldots \geq 0$ where $m \times n$ is the size of $\mathbf{A}$. For well-conditioned non-singular problems all "singular values" σ_i are non-zero and the condition number $\kappa = \sigma_1/\sigma_{\min(m,n)}$ is small. In case of well-conditioned singular problem, there is a particular k such that $\sigma_k \gg \sigma_{k+1}$ and $\sigma_i \approx 0$ for $i \geq k+1$. The "truncation" parameter k corresponds to the rank of the singular problem. Consider the following portioning of the SVD matrices:

$$\mathbf{U}_k = [\mathbf{u}_1 \ldots \mathbf{u}_k]; \;\; \Sigma_k = \text{diag}\,(\sigma_1 \ldots \sigma_k); \;\; \mathbf{V}_k = [\mathbf{v}_1 \ldots \mathbf{v}_k]$$

Accordingly, introduce the matrix $\mathbf{V}_{k+1}$ which consists of all $(m-k)$ remaining columns of $\mathbf{V}$. The matrix $\mathbf{A}_k = \mathbf{U}_k\Sigma_k\mathbf{V}_k^T$ represents the closest rank k approximation of the original matrix $\mathbf{A}$, see, e.g. [GvL89] for further details. Thus, instead of solving the original ill-posed problem we solve $\mathbf{A}_k\mathbf{F}_k = \mathbf{H}$ by computing

$$F_k = \mathbf{V}_k\Sigma_k^{-1}\mathbf{U}_k^T\mathbf{H} \tag{3}$$

for some selected "truncation parameter" k. Note that $\mathbf{F}_k$ has no component in the null-space of $\mathbf{A}$; the generalized solution space (for cases $k < \min(m,n)$) is obtained by adding a translation factor from $\mathbf{A}$s null-space

$$F = \mathbf{F}_k + \mathbf{F}_n = \mathbf{F}_k + \mathbf{V}_{k+1}\mathbf{f} \tag{4}$$

where $\mathbf{f}$ is a $(m-k)$-size vector of free parameters. Setting $\mathbf{f} = \mathbf{V}_{k+1}^T\mathbf{F}_0$ ($\mathbf{F}_0$ denotes coefficients of a previously known surface) and inserting $\mathbf{f}$ into above equation one obtains a solution which:

1. exactly solves $\mathbf{A}_k\mathbf{F}_k = \mathbf{H}$
2. minimizes the residual $\rho = \|\mathbf{AF}_k - \mathbf{H}\|$ and
3. minimizes the normed difference $\eta = \|\mathbf{F}_k - \mathbf{F}_0\|$

This procedure is called Truncation of the SVD (TSVD) and it is the common choice whenever a singularity is expected in a linear system, see [PTVF92]chapter 15, for example.

The difficulty with an *ill-posed* problem is that the singular values of an ill-posed problem decay gradually to zero without a significant jump in magnitude at a specific σ_k. Hence, determining the "numerical rank" k is more involved. The basic idea of *regularization* is to locate a "nearby" rank deficient but well-conditioned problem which approximates the original problem with sufficient accuracy. The choice of the truncation parameter k is apparently not arbitrary, since one seeks a solution $\mathbf{F}_k$ which minimizes the residual and the norm of the solution. Too small k will cause large error in ρ_k and too large k will increase η_k, since computing Σ_k^{-1} in eq. 3 involves divisions by small numbers.

It was observed by several researchers that if the values ρ_k and η_k are plotted in logarithmic scale for several ks a curve with characteristic "L-shape" results. We point to Hansen's paper [Han01] for historical overview and further background of "L-curves". Obviously, the "optimal" k will be such that the error in constraints (horizontal "leg" of the L-curve) is as small as possible and the solution is not yet influenced by nearly singular equations which is reflected by increasing norm of the solution vector (vertical leg of the L-curve). It is shown in [Han01] that this property is satisfied for truncation parameter which corresponds to (or is close to) the sharp corner of the L-curve. There is a more precise statement about the L-shape of the resulting curve the "Picard condition", see, e.g. [Han98]. Simply stated, for an ill-posed problem $\mathbf{H} = \mathbf{AF}$ the modified "left-hand side" $\mathbf{H} + \Delta\mathbf{H}$ must be "smooth enough to survive the inversion to $\mathbf{F}$". Otherwise, the L-curve possesses several distinct corners; it moves from one to the next in a "cascade"-like manner. Hence, we cannot expect that a meaningful solution can be computed for arbitrary curve requests; for example, returning to the surface from fig. 4(a), it is very unlikely that a reasonable surface (with that dimension and structure of the B-spline space) which interpolates the rapidly varying curve will exist. Following Hansen's advice in [Han90] in such cases we select the corner with smallest η. The effect of this choice is immediately forwarded to the user of our interactive system: the surface does not vary as strong as the sketched curve – the fast variation of the sketched curve is suppressed. This is a signal to the user that the required changes are not possible on given surface and in presence of specified constraints.

For the surface example in fig. 4 we obtain the L-curve shown in figure 5: The optimal truncation parameter corresponds to the "sharp" concave corner emphasized by an arrow; a closer look at this region (shown in a magnified scale in the right upper corner of the figure) reveals that the "corner" consists of a cluster of rapidly varying values (ρ_k, η_k). This leads to an idea to approximate the discrete points of the L-curve by a sufficiently smooth B-spline curve $L(t)$ and to locate a parameter value t_k where the curve exhibits largest negative curvature, see [Han01]. We have applied a more effective procedure: we start with a piecewise linear approximation of the ρ, η point set, increase the polynomial degree to cubic and apply a slightly modified version of Lyche-Mørken algorithm [LM87] to remove as many knots as possible. We build a statistics on errors caused by removal of a knot and increase the threshold which regulates the Lyche-Mørken recursion until the statistics contains only errors such that $|\log(\varepsilon_{\max}) - \log(\varepsilon_{\min})| \leq 1$. This heuristics relies on the observation that the L-curve has one

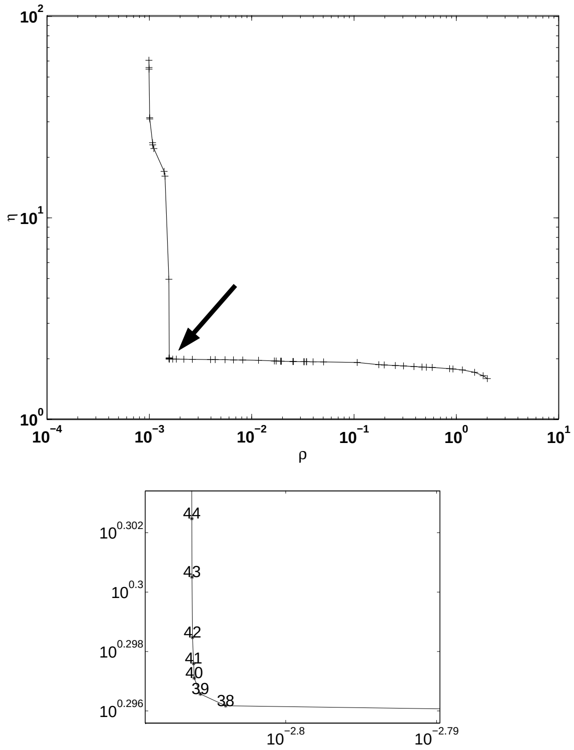


Figure 5: *The L-curve with the magnified "corner" region for constraint shown in fig. 4.*

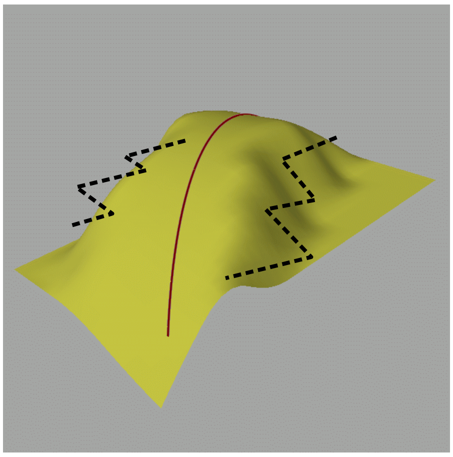


Figure 6: *The surface corresponding to the corner of the L-curve from fig. 5. The "aliasing" of the surface is emphasized by staircase shaped lines.*

(optimal case), or several, distinct regions of large curvature and is "smooth" elsewhere. Also, this method copes better with the irregular parameterization of the discrete L-curve data. At the output, we obtain a curve which has one (in the optimal case) or several C^0 discontinuities in the knot vector. The curve points corresponding to these discontinuities are (or are close to) the corner of the L-curve.

4.3. The "surface aliasing" effect

The solution surface corresponding to the L-curve corner at $k = 38$ is shown in fig. 6: observe, that the surface deforms along the curve in a "staircase" like manner. We call this the "surface aliasing effect", rf. [MB00] for its similarity with aliasing as known in computer graphics. A attempt to map a non-horizontal or non-vertical line on a discrete mesh of pixels is comparable to mapping a non iso-parametric line to the rectangular mesh of control points. The truncation parameter $k = 38$ means that the solution is stable when 38 out of 80 control points are determined. The remaining control points were determined according to side constraint $\min_k \|\mathbf{F} - \mathbf{F}_0\|$. This is apparently not sufficient: the transition among the dependent and independent control points is too abrupt. Such surface, although in algebraic sense correct (the example in fig. 6 renders the residual of approximately 10^{-3}, cf. to figure 5) will surely not be accepted by the designer. One has to consider a kind of "antialiasing".

4.4. L-curve regularization with side-constraint

In [MB00] we have proposed to locally re-parameterize the original surface F by a new surface F' such that the constrained curve H becomes an iso-parametric line in the domain of F'. All manipulations of H are transfered to F' which completely avoids aliasing and ill-conditioning of the problem, see the abovementioned paper for further details. Although the reparametrization is an effective method to remove most of the numerical problems, it is inconvenient: it can be carried out only for one curve constraint inside of a (curved) rectangular region on F. This is not always possible, e.g., if there are several interfering curve constraints. Furthermore, the underlying surface must remain "fixed" in order to avoid aliasing in F along the edge curves between F and F'.

Note that we have $(m-k)$ free parameters $\mathbf{f}$ and the solution space (eq. 4) obtained from the TSVD at our disposal. Thus, we may use these free parameters for stating additional constraints which help against the "aliasing" effect and improve the shape of the surface without destroying the already minimal residual of the TSVD solution. Well-suited for this purpose are the quadratic surface functionals frequently used in variational surface design, see [GS97], for example. It suffices to say that the most frequently used functionals are quadratic forms $\mathbf{F}^T\mathbf{L}\mathbf{F}$ in control points of the surface and, hence, possess one well-defined minimum at the solution of $\mathbf{L}\mathbf{F} = 0$. The matrices $\mathbf{L}$ can be computed efficiently by symbolical or numerical methods see [VBH92] or [Mic03]. In the following we will consider a linear combination of matrices $\mathbf{L} = \sum_{i=1}^{3} \alpha_i \mathbf{L}_i$ for $\sum_{i=1}^{3} \alpha_i = 1$, where $\mathbf{L}_i$, denote the normal equations of area, thin-plate energy and variation of curvature minimizing functionals for $i = 1, 2, 3$.

If the SVD of $\mathbf{A}$ is available, an efficient numerical method to solve such constrained approximation problem is the so-called, Modified TSVD (MTSVD) [HSS92]. It allows to define more general side constraints. Instead of $\|\mathbf{F} - \mathbf{F}_0\|$ one minimizes $\|\mathbf{L}(\mathbf{F} - \mathbf{F}_0)\|$ for some regularization matrix $\mathbf{L}$. In our case the selection of $\mathbf{L}$ is as described above. We know that the linear combination of the surface functionals is minimal at

$$\mathbf{L}(\mathbf{F} - \mathbf{F}_0) = 0$$

with the previously known solution $\mathbf{F}_0$ before the deformation.

Substituting eq. 4 for $\mathbf{F}$ in above equation yields

$$\begin{aligned} \mathbf{L}(\mathbf{F}_k + \mathbf{V}_{k+1}\mathbf{f} - \mathbf{F}_0) &= \mathbf{0} \\ \mathbf{L}\mathbf{V}_{k+1}\mathbf{f} &= \mathbf{L}(\mathbf{F}_0 - \mathbf{F}_k) \end{aligned} \quad (5)$$

Solving eq. 5 for $\mathbf{f}$ and substituting back to eq. 4 we obtain a "corrected" MTSVD solution $\mathbf{F}_{\mathbf{L},k}$ which in addition to properties 1-2 from §4.2 minimizes the expression $\|\mathbf{L}(\mathbf{F}_k - \mathbf{F}_0)\|$.

In [HSS92], Hansen proposes to compute the values of $\rho_k = \|\mathbf{A}\mathbf{F}_{\mathbf{L},k} - \mathbf{H}\|$ and $\eta_k = \|\mathbf{F}_{\mathbf{L},k}\|$ for each k and locate the corner of the "weighted" L-curve as in the case of TSVD. Consider, that there may be several hundred up to several thousand truncation parameters, thus, we must be able to obtain the penalty term $\mathbf{f}$ fast. The linear system in eq. 5 can be solved in a particularly efficient way by pre-computing the worst case QR-decomposition of $\mathbf{L}\mathbf{V}_{k+1}$, i.e., for the smallest k ever used and update it for the current k accordingly, see [HSS92] and [GvL89]§12.6 for details. Thus obtaining the proper "correction" term $\mathbf{f}$ for each k is a computationally cheap procedure. Note that the pre-condition for this procedure is that the matrix $\mathbf{L}\mathbf{V}_{k_{\min}+1}$ must be non-singular. Otherwise, the trick of updating the QR-decomposition does not work and the eq. 5 must be solved for each k from scratch. Fortunately, if $\mathbf{L}$ is the linear combination of matrices as described above, the matrix $\mathbf{L}\mathbf{V}_{k_{\min}+1}$ tends to be non-singular for $\alpha_i \geq 0$, reasonable parameterization of the B-spline surface F and $k_{\min}$ larger than three, see [Die96] or [Die98], for example. Theoretically, matrices gained from quadratic surface functionals always posses a trivial solution: the objective function equals zero if all control points are zero. One has to assure that at least three or more linearly independent constraints exist. Our experience shows that, in practice, even more linearly independent constraints may be necessary – we assume that this is due to the numerical errors which are contained in the regularization matrices.

We have found that the optimal truncation parameter obtained by MTSVD methods rarely differs from k obtained by TSVD; moreover, since the system matrix in eq. 5 is not particularly well conditioned (the typical condition numbers obtained in our experiments were between 10^3-10^4), the discrete L-curve points are considerably more "scattered". Hence, we recommend to compute k by the TSVD method and solve for $\mathbf{f}$ in eq. 5 only once. Figure 7 shows three different modifications of the same surface and the same constraint as in figure 4 after the MTSVD correction was applied. No aliasing can be observed, the error in the residual is less than 10^{-3}. The example in figure 8 shows the computed result of the dome shaped example from figure 2 with two curve constraints.

5. Conclusions and future work

In this paper we presented part of our free-form sketching system intended to provide the designers with more intuitive ways in designing free-form shapes. In particular, combining the concept of free-hand sketching with constraint-based surface sculpting, we hope to achieve reasonable compromises between real-world design tools (e.g. pencil and paper) and virtual computer tools. With regard to surface interpolation as proposed in section 4 we summarize the problems and subjects to future work as follows:

Efficiency

In previous work we have put much effort into speed up of setting up the equation system using "unevaluated" polynomial composition. The result is that for examples presented throughout this papers, setting up the equations takes only a fraction of second. The subject of future work will be to improve the efficiency of solving the equation system: Although the SVD provides a perfect insight into an ill-conditioned system obtaining the SVD for a general matrix is a computationally expensive procedure. The amount of work required by currently known symbolic algorithms (the Golub-Kahan method [GvL89]Sec. 5.4 and 8.3) to perform the full SVD is approximately $O(m^2 + n^3)$. Note that the SVD needs to be computed only once in a "life-time" of a system of linear constraints: computing a surface after a curve modification from available SVD is fast and delivers a new surface shape in real time. Nevertheless, for large data sets computing the SVD may cause quite unpleasant delay-times during the design work: for moderate sizes of $\mathbf{A}$, such as used in above examples, i.e. few hundred columns and rows the LAPACK functions "*gesvd*" or "*gesdd*" require about 1-3 seconds on an 750 MHz PC. This is acceptable; however, consider that the model may consist of several surfaces and many curve constraint. Also, the run-time grows rapidly for larger examples: for surfaces with about 500 control points and more than thousand constraint equations the SVD already requires prohibitive 15-20 seconds. Unfortunately, the alternatives to SVD are by far not that straightforward and reliable. Other, cheaper to compute, rank-revealing factorization is the so-called Rank-revealing QR-factorization (RR-QR) as proposed by Chan and Hansen in [CH90]. In order to apply this method, at least approximate guess on the location of the optimal truncation parameter is required. Otherwise, the effort becomes comparable to computing the SVD. A very promising method seems to be the Sparse Multifrontal Rank Revealing QR Factorization described by Pierce and Lewis in [PL97]. However the application to our problem has to be further examined.

We also intend to investigate the usage of so-called iterative methods for large and sparse linear systems of equations. The matrices occurring in our application are sparse with predictable structure which can not be utilized by the symbolic SVD algorithms. The gain on efficiency when using iterative solvers is considerable. Moreover, the so-called "Krylov subspace" methods exhibit similar behavior as the SVD: they tend to filter out the stable components of the solution first. Thus, the discrete L-curve method can be applied to obtain the optimal truncation parameter. On the other hand, for ill-posed problems, the convergence of most iterative methods is poor: methods such as the famous "Conjugate Gradient" or "GM-RES" are guaranteed to converge for non-singular equations only. We have run few tests with these methods and in the cases when they converged obtaining the solution took only fraction of the time required for SVD. The problems to be solved are to define the right "pre-conditioning" method which is suited for this type of equation systems, see e.g. [GvL89]Chap. 10 for further details on this topic.

The smoothing constraints

The "smoothed" MTSVD solution tends to generate "too stiff" surfaces whenever there are not sufficiently many constraint which restrict the shape of the surface. Considering the previously known surface $\mathbf{F}_0$ when we solve for the penalty term $\mathbf{f}$ in eq. 5 prevents the surface from collapsing into a narrow strip somewhere around the curve constraint. However, for certain curves, the surface functionals used in this work and the MTSVD method generate almost

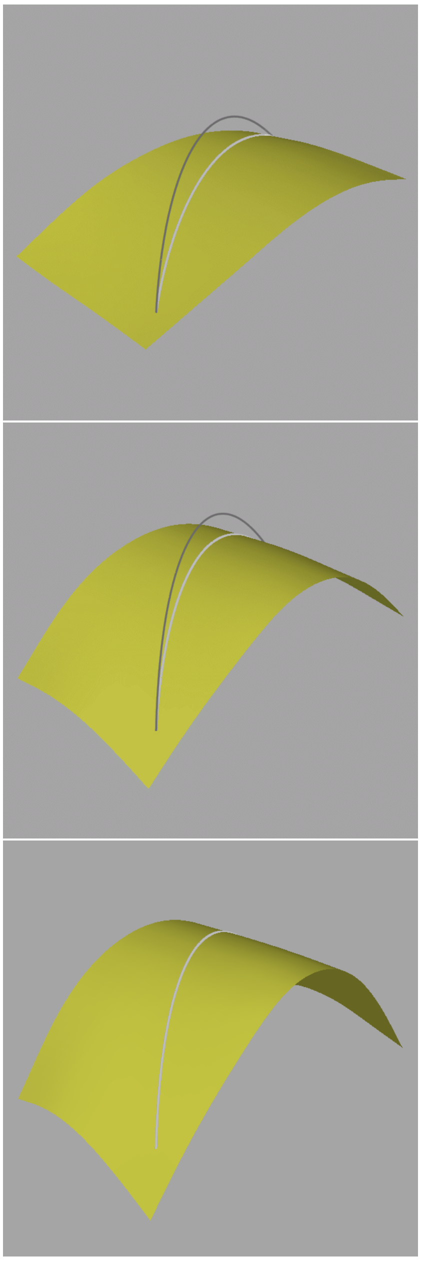


Figure 7: *Bi-cubic surface obtained by the L-curve regularization with side-constraint. The modifications of the surface proceed from top to bottom, the "next" sketched curve is shown in darker color. Adding the side constraint which acts as a "smoothing term" to the L-curve solution removes the aliasing artifacts and delivers an optically and geometrically smooth surface.*

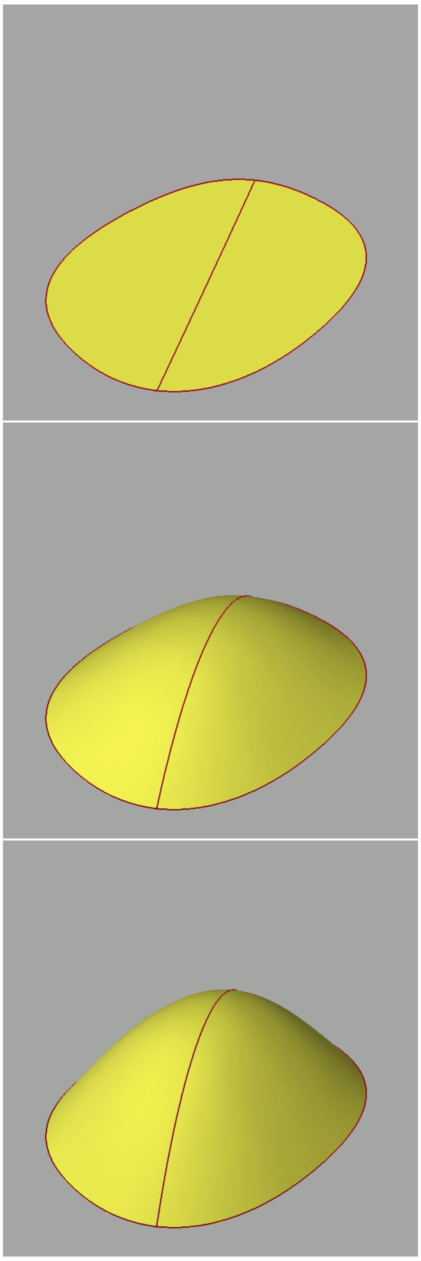


Figure 8: *The example shows the design of a "dome" surface with two curve constraints. The closed loop remains "fixed", the other curve is modified by subsequent pen-strokes. The bi-cubic surface with* 10×10 *control points was trimmed outside of the fixed loop.*

"ruled" surfaces and may smooth out desired features of the original surface. This is rarely the shape a designer would expect. One possibility is to use so-called "data-dependent" functionals introduced by Greiner in [Gre94]. They take the shape of a reference surface (assumed to be known) into account more strongly than eq. 5 does. The surface sculpting is an iterative process, thus, a surface from previous design step is always available and should be used as such reference. If working in "interpolation mode", see §3.1, a reference surface can be computed from sketched curves as a raw least squares fit to a bi-linear surface, for example. Another possibility, simpler to implement, is to use incomplete QR-factorization when solving eq. 5: this allows to regulate the amount of smoothing by considering only a "lower energy" factor **f**. The disadvantage is that the QR-factorization has to be carried out with pivoting and hence, has to be computed from scratch for each new shape of the constrained curve.

Acknowledgments

We would like to thank the reviewers for providing various hints for improvements and pointing out additional relevant literature. This project was funded, in part, by a grant from the German Ministry of Education and Research within the "VRIB" project (contract Nr. 01 IR A05E).

References

[BDKM00] BRÜDERLIN B., DÖRING U., KLEIN R., MICHALIK P.: Declarative geometric modeling with constraints. In *Conference proceedings CAD 2000* (Berlin, March 2000), Iwainsky A., (Ed.), GFAI. 1, 5

[BHH93] BONNEAU G. P., HAGEN H., HAHMANN S.: Variational Surface Design and Surface Interrogation. *Computer Graphics Forum 3*, 12 (1993), 447–459. 3

[BHS93] BRUNNETT G., HAGEN H., SANTARELLI P.: Variational Design of Curves and Surfaces. *Surv. Math. Industry 3*, 27 (1993). 3

[CH90] CHAN T., HANSEN P. C.: Computing Truncated Singular Value Decomposition Least Square Solutions by rank revealing QR-Factorizations. *SIAM J. Sci. Stat. Comput. 11*, 3 (May 1990), 519–530. 8

[CHR01] CALVETTI D., HANSEN P. C., REICHEL L.: L-curve curvature bounds via Lanczos bidiagonalization. *Electronic Transactions on Numerical Analysis* (May 2001). To obtain under www.imm.dtu.dk. 2

[CW92] CELNIKER G., WELCH W.: Linear constraints for deformable B–spline surfaces. *Computer Graphics 25*, 2 (March 1992), 171–174. 2, 3

[DGHM93] DEROSE T., GOLDMAN R., HAGEN H., MANN S.: Functional composition via blossoming. *ACM Transactions on Graphics 12*, 2 (April 1993). 2

[Die96] DIETZ U.: B-Spline Approximation with Energy Constraints. In *Reverse Engineering*. B.G. Teubner, Stuttgart, 1996. 8

[Die98] DIETZ U.: Creation of Fair B-Spline Surface Fillets. In *Creating Fair and Shape Preserving Curves and Surfaces*. B.G. Teubner, Stuttgart, 1998. 2, 3, 8

[DMB98] DOERING U., MICHALIK P., BRÜDERLIN B.: A constraint–based shape modeling system. In *Geometric Constraint Solving & Applications*. Springer Verlag, June 1998. 1, 5

[EC97] ELBER G., COHEN E.: Filleting and rounding using trimmed tensor product surfaces. In *Proceedings The fourth ACM/IEEE Symposium on Solid Modeling and Applications* (May 1997), pp. 201–216. 2, 3

[Far92] FARIN G.: *Curves and Surfaces for CAGD*, 3. ed. Computer Science and Scientific Computing. Academic Press, Inc., 1992. 2

[FG90] FERGUSON D. R., GRANDINE T. A.: On the construction of surfaces interpolating curves: I. a method for handling non-constant parameter curves. *ACM Transactions on Graphics 9*, 2 (1990), 213–215. 2, 3

[GR74] GORDON W. J., RIESENFELD R. F.: B-spline curves and surfaces. In *Computer Aided Geometric Design*, Barnhill R. E., Riesenfeld R. F., (Eds.). Academic Press, 1974. 1

[Gre94] GREINER G.: Variational design and fairing of spline surfaces. *Computer Graphics Forum (Proc. Eurographics '94)*, 3 (1994), 143–154. 3, 9

[GS97] GREINER G., SEIDEL H.-P.: Automatic modeling of smooth spline surfaces. In *Proc. WSCG '97* (1997), N. Magnenat-Thalmann V. S., (Ed.), pp. 665–675. 3, 7

[GvL89] GOLUB G. H., VAN LOAN C. E.: *Matrix Computations*, 2 ed. John Hopkins Series in the Mathematical Sciences. The John Hopkins University Press, 1989. 2, 6, 8

[Han90] HANSEN P. C.: Truncated Singular Value Decomposition Solutions to Discrete Ill–posed Problems With Ill–determined Numerical Rank. *SIAM J. of Scientific Computing 11*, 3 (May 1990), 519–530. 3, 6

[Han98] HANSEN P. C.: *Regularization Tools*. Department of Mathematical Modeling, Technical University of Denmark, www.imm.dtu.dk, March 1998. 2, 3, 6

[Han01] HANSEN P. C.: The L-curve and its use in the numerical treatment of inverse problems. In *Computational Inverse Problems in Electrocardiology* (Southampton, 2001), invited paper for P. Johnston, (Ed.), WIT Press, pp. 119–142. 2, 6

[HL89] HOSCHEK J., LASSER D.: *Fundamentals of Computer Aided Geometric Design*, 2. ed. A K Peters Ltd., 1989. 2

[HSS92] HANSEN P. C., SEKH T., SHIBAHASHI H.: The Modified Truncated SVD Method for Regularization in General Form. *SIAM J. of Scientific Computing 13*, 5 (September 1992), 1142–1150. 3, 7, 8

[KPS95] KOLB A., POTTMANN H., SEIDEL H.-P.: Fair surface reconstruction using quadratic functionals. In *Eurographics Proc. '95* (1995), Eurographics, Blackwell Publishers, pp. 469–479. 3

[LM87] LYCHE T., MØRKEN K.: A Discrete Approach to Knot Removal and Degree Reduction Algorithms for Splines. In *Algorithms for Approximation*, Mason J., Cox M., (Eds.). Oxford University Press, 1987, pp. 67–82. 6

[MB99] MICHALIK P., BRÜDERLIN B.: Computing curve–surface incidence constraints efficiently. In *Proceedings Swiss Conference on CAD/CAM* (February 1999). 1, 2

[MB00] MICHALIK P., BRÜDERLIN B.: A constraint-based method for sculpting free-form surfaces. In *Computing, special issue of the Dagstuhl Seminar on Ge-*

ometric Modelling, Brunnett G., Bieri H. P., (Eds.). Springer Verlag, 2000. 1, 2, 7

[MB01] MICHALIK P., BRUEDERLIN B.: Introducing parametrization in surface models by means of geometric constraints. In *Proceedings Scanning and Human Body Modeling* (Paris, France, May. 4-5 2001). 1, 2

[Mic03] MICHALIK P.: *Methods for Constraint-based Conceptual Surface Design*. PhD thesis, Technical University of Ilmenau, Dept. of Computer Science, 2003. Available online from `www.bibliothek.tu-ilmenau.de`. 1, 2, 5, 7

[MKB02] MICHALIK P., KIM D., BRUDERLIN B.: Sketch- and Constraint-based Design of B-spline Surfaces. In *Proceedings of International Conference on Solid Modeling* (Saarbrücken, Germany, 2002). 1, 3, 4, 5

[Pet01] PETERS J.: Geometric continuity. Dept C.I.S.E, University of Florida, February 2001. 2

[PL97] PIERCE, LEWIS: Sparse multifrontal rank revealing qr factorization. *SIAM Journal on Matrix Analysis and Applications*, 18 (1997). 8

[PT95] PIEGL L., TILLER W.: *The Nurbs Book*. Springer Verlag, 1995. 2

[PTVF92] PRESS W., TEUKOLSKY S., VETTERLING W., FLANNERY B.: *Numerical Recipies in C: The art of scientific computing*. Cambridge Unversity Press, 1992. 6

[SU90] SCHUMAKER L. L., UTRERAS F. I.: On Generalized Cross validation for Tensor Smoothing Splines. *SIAM J. Sci. STAT. Comput. 11*, 4 (July 1990), 713–731. 3

[TB97] TREFETHEN L. N., BAU D.: *Numerical Linear Algebra*. Sociaty for Industrial and Applied Mathematics, Philadelphia, PA, 1997. 6

[VBH92] VERMEULEN A. H., BARTELS R. H., HEPPLER G. R.: Integrating Product of B-Splines. *SIAM J. Sci. Stat. Computing 13*, 4 (July 1992), 1025–1038. 7

[WW92] WELCH W., WITKIN A.: Variational surface modeling. *Computer Graphics 26*, 2 (July 1992), 157–165. 2, 3