Approximation with Active B-spline Curves and Surfaces

Helmut Pottmann, Stefan Leopoldseder, Michael Hofer
Institute of Geometry
Vienna University of Technology
Wiedner Hauptstr. 8–10, Vienna, Austria
{pottmann,leopoldseder,hofer}@geometrie.tuwien.ac.at

Abstract

An active contour model for parametric curve and surface approximation is presented. The active curve or surface adapts to the model shape to be approximated in an optimization algorithm. The quasi-Newton optimization procedure in each iteration step minimizes a quadratic function which is built up with help of local quadratic approximants of the squared distance function of the model shape and an internal energy which has a smoothing and regularization effect. The approach completely avoids the parametrization problem. We also show how to use a similar strategy for the solution of variational problems for curves on surfaces. Examples are the geodesic path connecting two points on a surface and interpolating or approximating spline curves on surfaces. Finally we indicate how the latter topic leads to the variational design of smooth motions which interpolate or approximate given positions.

1. Introduction

Curve and surface approximation is a central topic of Computer Aided Geometric Design and there is a large body of literature dealing with it. It is beyond the scope of this paper to give an overview of the many available algorithms. Thus we will just point to the work which is in close connection with our algorithms and even there we will mainly focus at the literature which comes not directly from the CAGD community, but from areas like Computer Vision and applications of partial differential equations.

At first, let us consider approximation with parametric curves or surfaces, which are usually expressed by means of B-splines [23, 44]. The usual approach uses a least squares formulation with a regularization term that expresses the fairness of the final result (see e.g. [23, 24]).

The principle, explained for surfaces, is as follows. Let $\mathbf{p}_k,\ k = 1, \ldots, N$, be the given data points or samples on a given model surface. We are looking for an approximating B-spline surface or another parametric surface with a representation of the form

$$\mathbf{x}(u, v) = \sum_{i=1}^{n} B_i(u, v)\,\mathbf{d}_i. \tag{1}$$

The basis functions $B_i(u, v)$ are usually polynomial, piecewise polynomial, or piecewise rational. We assume that the functions B_i are given or precomputed from the input data; thus weights or knots are already determined. Then one estimates the surface parameters $(u_k, v_k), k = 1, \ldots, N$, of those points $\mathbf{x}(u_k, v_k)$ on the approximant which should be close to the corresponding data points $\mathbf{p}_k$. The approximant is computed as minimizer of a functional

$$F = \sum_k \|\mathbf{x}(u_k, v_k) - \mathbf{p}_k\|^2 + \lambda F_s. \tag{2}$$

The first part is a quadratic function in the unknown control points $\mathbf{d}_i$,

$$\sum_k \|\mathbf{x}(u_k, v_k) - \mathbf{p}_k\|^2 = \sum_k \Big[\sum_{i=1}^{n} B_i(u_k, v_k)\mathbf{d}_i - \mathbf{p}_k\Big]^2.$$

The second part F_s in (2) is a smoothing term (see e.g. [6]). A frequently used example is the simplified thin plate energy, a quadratic function in the second partial derivatives,

$$F_s = \int\!\!\int (\mathbf{x}_{uu}^2 + 2\mathbf{x}_{uv}^2 + \mathbf{x}_{vv}^2)\,du\,dv. \tag{3}$$

It is also quadratic in the unknowns $\mathbf{d}_i$ and thus the minimization of F is the minimization of a quadratic function and amounts to the solution of a linear system of equations.

It is a difficult task to estimate the parameters (u_k, v_k). This parameter choice largely effects the result (see e.g. [33] and the references therein). Therefore, iterative parameter correction procedures have been suggested [23]. The final approximant should exhibit error vectors $\mathbf{x}(u_k, v_k) - \mathbf{p}_k$ which are orthogonal to the approximating surface $\mathbf{x}(u, v)$.

Fortunately, there is a way to overcome the parameterization problem. The idea comes from Computer Vision and Image Processing, where so-called *active contour models* are used in a variety of applications (see e.g. [3]). The origin of this technique is a paper by Kass et al. [27], where a variational formulation of parametric curves, coined *snakes*, is used for detecting contours in images.

An elegant formulation of curve and surface reconstruction and segmentation problems is the concept of *geodesic active contours* [8, 9, 49]. There, the curve to be reconstructed, e.g. from a medical image, is found as geodesic in a Riemannian space whose metric is derived from the input (image). Analogously, surface reconstruction is reformulated as minimal surface computation in a Riemannian space. The literature on this highly interesting topic is rapidly increasing. A good overview of the methods is found in the book by G. Sapiro [49].

Instead of a parametric representation of a planar curve or a surface, one may use an *implicit form* as zero set (level set) of a bivariate/trivariate function [5]. A major advantage of this approach is the simplicity with which one can model arbitrary topologies. Often, the function which the level sets are taken from is assumed to be polynomial. This yields algebraic surfaces. For algebraic surface fitting, especially the fitting of algebraic tensor-product spline surfaces to scattered data, we refer the reader to the recent paper of Jüttler and Felis [26]. Some approaches to surface reconstruction via implicit surfaces define a signed distance function to the data set and denote the zero contour of the signed distance function as the reconstructed implicit surface [1, 4].

The formulation of active contour models via level sets goes back to Osher and Sethian [42]. The *level set method* [41, 53] has been successfully applied to the solution of a variety of problems, e.g. for segmentation and analysis of medical images [36]. There are also several extensions to surfaces. An application to the surface fitting problem to scattered data sets has been given by Zhao et al. [61, 62]. For further work on the level set method which is related to surface reconstruction, see [17, 18, 60].

In the present investigation, we develop further a concept for parametric curve and surface fitting which has recently been developed by the authors [46]. We assume as input a rather dense set of points or even a given curve or surface representation. We refer to it as *model shape*. The first situation arises for example when we are processing data of modern 3D scanners, or if we would like to fit a surface to a dense mesh. A curve or surface Φ as input may arise when the representation of Φ is not in the desired parametric representation. For example, we may have as input an implicit representation or a B-spline curve/surface with a too high degree or too many knots. Thus, we are also contributing to the problems of *spline conversion, degree reduction* and *knot removal* [23, 44]. Moreover, surfaces derived from a given surface in various ways might not be in the desired form. A well-known example are offset surfaces of NURBS surfaces, which are usually not NURBS surfaces themselves. The present approach is very well suited for *offset surface approximation* [35].

The basic idea of our approach is the application of a Newton type algorithm for the solution of the nonlinear problem of curve and surface approximation. Thus we use an iterative method, i.e., an active curve or surface which adapts to the final model shape. Moreover, for a Newton algorithm we require local quadratic (Taylor) approximants of the function to be minimized. Since we want to minimize the sum of squared distances to the curve or surface to be approximated, we guide the shape change of the active curve/surface with help of local quadratic approximants of the squared distance function d^2 of the model shape. The local approximants help to move the active curve/surface $\mathbf{x}(u, v)$ to lower levels of d^2, without having to specify which point $\mathbf{x}(u, v)$ should move to which point of the model shape. In this way we are avoiding the parametrization problem.

Our method is applicable to approximation with any linear curve or surface scheme, even subdivision curves and surfaces. In this paper we restrict ourselves to a B-spline representation.

The organization of our paper is as follows. In section 2, we review our recent work on local quadratic approximants of the squared distance function d^2 of curves and surfaces [45], since it forms the basis for the new approximation technique. In section 3, the new concept is outlined for curve approximation and applied to degree reduction and offset approximation. In section 4, we describe surface approximation and present some examples. Section 5 deals with curves on surfaces. It is shown how to compute the shortest path (geodesic) between two points on a surface and how to compute spline curves on surfaces. Moreover, we describe how the new technique naturally leads to a variational formulation for the computation of smooth motions, which interpolate or appoximate a given set of positions. Finally, in section 6, we describe possible extensions and indicate directions for future research.

2. Local quadratic approximants of the squared distance function of curves and surfaces

The algorithmic concept we are proposing heavily relies on local quadratic approximants of the squared distance function of the curve/surface Φ or point cloud to which we would like to fit a B-spline curve/surface.

Let us first consider the distance function of a curve or surface Φ, which assigns to each point $\mathbf{p}$ the shortest distance of $\mathbf{p}$ to Φ. A variety of contributions deals with the computation of this function; in many cases this computation aims towards the singular set of the function, i.e., towards points where the function is not smooth since those points lie on the medial axis (or skeleton) of the input shape.

Early work on the geometry of the distance function comes from the classical geometric literature of the 19th century. One looks at its graph surface, which consists of developable surfaces of constant slope and applies results of classical differential geometry, line and sphere geometry (for a modern presentation, see e.g. [47]). For more recent work on distance transforms and the closely related medial axis transform, see [10, 11, 29, 40, 51, 52, 56].

The distance function is also the (viscosity) solution of the so-called eikonal equation. Its numerical computation is not trivial because the eikonal equation is a hyperbolic equation and an initially smooth front may develop singularities (shocks) as it propagates. Precisely the latter belong to the medial axis and are of particular interest. The computation of viscosity solutions with the level set method of Osher and Sethian [42, 41] proved to be a very powerful method (see e.g. [49, 54, 53]).

For our approach to surface approximation, not the distance function itself but the *squared* distance function is important. We are especially interested in local quadratic approximants of that function. For a derivation and proofs of the following results we refer the reader to [45]. For a better understanding, we first present local quadratic approximants of planar curves and then move to surfaces and space curves.

2.1. Local quadratic approximants of the squared distance function of a planar curve

In a Euclidean plane we consider a C^2 curve $\mathbf{c}(t)$ with parameterization $(c_1(t), c_2(t))$. The Frenet frame at a curve point $\mathbf{c}(t)$ consists of the unit tangent vector $\mathbf{e}_1 = \dot{\mathbf{c}}/||\dot{\mathbf{c}}||$ and its normal vector $\mathbf{e}_2(t)$. Those two vectors form a right-handed orthonormal basis in the plane.

We are interested in the squared distance function d^2 which assigns to each point $\mathbf{p}$ in π the square of its shortest distance to the curve $\mathbf{c}(t)$. In the following we give the formula for a local quadratic (Taylor) approximant of the squared distance function with respect to a local Frenet coordinate system. Note that the squared distance function is not smooth in points of the medial axis. Thus, we will not compute local quadratic Taylor approximants for points of the medial axis.

Consider a point $\mathbf{p}$ in π whose coordinates with respect to the Frenet frame at the normal footpoint $\mathbf{c}(t_0)$ are $(0, d)$. The curvature center $\mathbf{k}(t_0)$ at $\mathbf{c}(t_0)$ has coordinates $(0, \rho)$. Here, ρ is the inverse curvature $1/\kappa$ and thus has the same sign as the curvature, which depends on the orientation of the curve (see Fig. 1).

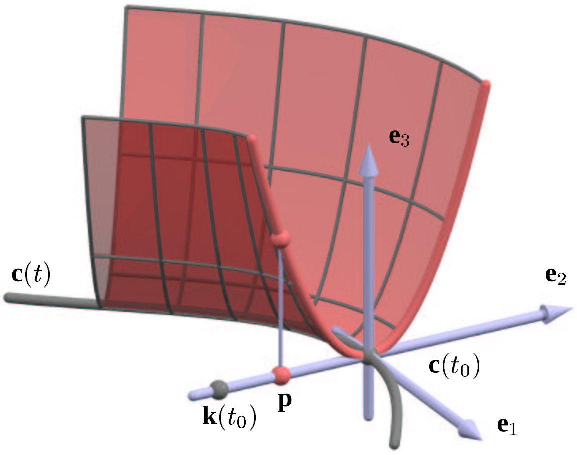

Figure 1. Graph of the squared distance function d^2 to a planar curve $\mathbf{c}(t)$.

With respect to the Frenet frame, the second order Taylor approximant F_d of the squared distance function d^2 at $(0, d)$ is given by

$$F_d(x_1, x_2) = \frac{d}{d-\rho}x_1^2 + x_2^2. \tag{4}$$

For a derivation of this result and a discussion of the different types of F_d we refer the reader to [45]. We just point out that the Taylor approximants may be indefinite. As shown in [45], we can use as appropriate nonnegative quadratic approximant

$$F_d^+(x_1, x_2) = \frac{d}{d+\rho}x_1^2 + x_2^2, \tag{5}$$

where d, ρ are taken as positive. Equ. (5) is not valid for points beyond the curvature centers, but they will not arise anyway when we consider global distances.

2.2. Local quadratic approximants of the squared distance function of a surface

Consider an oriented surface $\mathbf{s}(u, v)$ with a unit normal vector field $\mathbf{n}(u, v) = \mathbf{e}_3(u, v)$. At each point $\mathbf{s}(u, v)$, we have a local right-handed Cartesian system whose first two vectors $\mathbf{e}_1, \mathbf{e}_2$ determine the principal curvature directions. The latter are not uniquely determined at an umbilical point, but in that case we can take any two orthogonal tangent vectors $\mathbf{e}_1, \mathbf{e}_2$. We will refer to the thereby defined frame as *principal frame* $\Sigma(u, v)$. Let κ_i be the (signed) principal curvature to the principal curvature direction $\mathbf{e}_i$, $i = 1, 2$, and let $\rho_i = 1/\kappa_i$. Then the two principal curvature centers at the considered surface point $\mathbf{s}(u, v)$ are expressed in Σ

as $\mathbf{k}_i = (0, 0, \rho_i)$. The quadratic approximant F_d of d^2 at $(0, 0, d)$ is the following:

Proposition 1 *The second order Taylor approximant of the squared distance function of a surface at the point* $\mathbf{p} = (0, 0, d)$ *is given by*

$$F_d(x_1, x_2, x_3) = \frac{d}{d - \rho_1} x_1^2 + \frac{d}{d - \rho_2} x_2^2 + x_3^2, \quad (6)$$

when coordinates are given with respect to the principal frame at $\mathbf{s}(u, v)$.

Let us look at two important special cases.

- For $d = 0$ we obtain

  $$F_d(x_1, x_2, x_3) = x_3^2.$$

  This means that the second order approximant of d^2 at a surface point **p** is the same for the surface $\mathbf{s}(u, v)$ and for its *tangent plane* at **p**. Thus, if we are close to the surface, the squared distance function from the tangent plane at the footpoint is a very good approximant. At least at first sight it is surprising that the tangent plane, which is just a first order approximant, yields a second order approximant when we are considering the squared distance function d^2, to surface and tangent plane, respectively.

- In the limit $d \to \infty$ we obtain

  $$F_\infty(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2.$$

  This is the squared distance from the footpoint on the surface.

We see that distances from footpoints yield good approximations if we are in the 'far field' of the surface $\mathbf{s}(u, v)$. In the near field it is much better to use other local quadratic approximants. The simplest one is the squared distance from the tangent plane at the footpoint.

Analogous to the curve case, we may have an indefinite Taylor approximant. Then we can use as appropriate nonnegative quadratic approximant

$$F_d^+(x_1, x_2, x_3) = \frac{d}{d + \rho_1} x_1^2 + \frac{d}{d + \rho_2} x_2^2 + x_3^2. \quad (7)$$

Here, d, ρ_1, ρ_2 are taken as positive. Equ. (7) is not valid for points beyond the principal curvature centers. Such points do not arise anyway when we consider global distances.

2.3. Local quadratic approximants of the squared distance function to a space curve

Given a point **p** in $\mathbb{R}^3$, the shortest distance to a C^2 space curve $\mathbf{c}(t)$ occurs along a normal of the curve or at a boundary point of the curve. The latter case is trivial and thus we exclude it from our discussion. At the normal footpoint $\mathbf{c}(t_0)$ we define a Cartesian coordinate system with $\mathbf{e}_1$ as tangent vector and $\mathbf{e}_3$ in direction of the vector $\mathbf{p} - \mathbf{c}(t_0)$. This *canonical frame* can be viewed as limit case of the principal frame for surfaces, when interpreting the curve as pipe surface with vanishing radius. By this limit process, we can also show the following result.

Proposition 2 *The second order Taylor approximant of the squared distance function of a space curve* $\mathbf{c}(t)$ *at the point* $\mathbf{p} = (0, 0, d)$ *is given by*

$$F_d(x_1, x_2, x_3) = \frac{d}{d - \rho_1} x_1^2 + x_2^2 + x_3^2, \quad (8)$$

where coordinates are given with respect to the canonical frame. Here, $(0, 0, \rho_1)$ *are the coordinates of the intersection point of the curvature axis of* $\mathbf{c}(t)$ *at the footpoint* $\mathbf{c}(t_0)$ *with the perpendicular line* $\mathbf{p}\mathbf{c}(t_0)$ *from* **p** *to* $\mathbf{c}(t)$.

As expected, with $d = 0$ we obtain

$$F_d(x_1, x_2, x_3) = x_2^2 + x_3^2.$$

This means that the second order approximant of d^2 at a curve point is the same for the curve **c** and for its tangent.

3. Approximation with an active curve in the 'squared distance field'

Our approach to curve approximation has as input a model shape M. This can be a sufficiently dense point set along a curve. It can also be a curve in $\mathbb{R}^2$ or $\mathbb{R}^3$ which is given in any mathematical representation. For the sake of simplicity in our explanation, we confine ourselves at first to planar curves, but the concept works in higher dimensions as well.

From the model shape M, we compute — for example by means of second order Taylor approximants — local quadratic approximants of the squared distance function d^2 of M. Thus, for any point $\mathbf{p} \in \mathbb{R}^2$, we have a way to compute such a local quadratic approximant $F_{d,\mathbf{p}}$. In its simplest formulation, the method outlined below further assumes that $F_{d,\mathbf{p}}$ is a *nonnegative* quadratic function.

The *active curve model* we are using is of the following nature. It is governed by *control points* $\mathbf{d}_i,\ i = 1, \dots, n$, and there is a *linear* relation which computes from the control point set a larger set of curve points $\mathbf{s}_k,\ k = 1, \dots, N$, or points on a refined model. For example, we may have a Bézier or B-spline curve of the form

$$\mathbf{c}(t) = \sum_{i=1}^{n} B_i(t)\mathbf{d}_i. \tag{9}$$

We now evaluate the curve at parameters t_k and get curve points

$$\mathbf{s}_k = \mathbf{c}(t_k).$$

We could also work with a subdivision curve: The points $\mathbf{d}_i$ can be the vertices of a coarse subdivision level and the points $\mathbf{s}_k$ can be vertices of a refined model, after application of a few steps of the subdivision rule [59]. The set of points $\mathbf{s}_k$ must be large enough to capture the shape of the active curve well. In the following we use the notation

$$\mathbf{s}_k = L_k(\mathbf{d}_1, \ldots, \mathbf{d}_n) \tag{10}$$

to express the *linear* computation of $\mathbf{s}_k$ from the control points $\mathbf{d}_i$.

The key idea is to iteratively change the input control points $\mathbf{d}_i$ so that the active curve deforms towards the model shape M. We do not use the gradient flow in the squared distance field but in each step solve a minimization problem which ensures that we quickly move to lower levels of the function d^2.

The method now proceeds in the following steps.

1. Initialize the active curve and determine the boundary conditions. This requires the computation of an initial set of control points $\mathbf{d}_i,\ i = 1, \ldots, n$, the proper treatment of boundaries (such as fixing end points of a curve segment) and the avoidance of model shrinking during the following steps. More details on that are described in section 4.

2. Repeatedly apply the following steps a.–c. until the approximation error or change in the approximation error falls below a predefined threshold:

   a. With the current control points $\mathbf{d}_i$, compute, for $k = 1, \ldots, N$, the active curve point $\mathbf{s}_k = L_k(\mathbf{d}_1, \ldots, \mathbf{d}_n)$ and a local quadratic approximant $F_{d,\mathbf{s}_k} =: F_d^k$ of the squared distance function of the model shape M at the point $\mathbf{s}_k$. This has to be a nonnegative quadratic function, $F_d^k(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathbb{R}^2$.

   b. Compute displacement vectors $\mathbf{c}_i,\ i = 1, \ldots, n$, for the control points $\mathbf{d}_i$ by minimizing the functional

   $$F = \sum_{k=1}^{N} F_d^k(L_k(\mathbf{d}_1 + \mathbf{c}_1, \ldots, \mathbf{d}_n + \mathbf{c}_n)) + \lambda F_s. \tag{11}$$

   Thus, our goal is that the new curve points

   $$\mathbf{s}_k^* = L_k(\mathbf{d}_1 + \mathbf{c}_1, \ldots, \mathbf{d}_n + \mathbf{c}_n),$$

   which are linear combinations of the new control points $\mathbf{d}_i^* = \mathbf{d}_i + \mathbf{c}_i$, are closer to the model shape than the old active curve points $\mathbf{s}_k$. The functional F_s is a smoothing functional which shall be quadratic in the control points of the active curve. Thus, F_s is a quadratic function in the new control points $\mathbf{d}_i + \mathbf{c}_i$, and also quadratic in the unknowns $\mathbf{c}_i$.

   We see that step b. requires the minimization of a quadratic function F in the displacement vectors $\mathbf{c}_i$ of the control points. This amounts to the solution of a linear system of equations. Note that the factor λ in Equ. (11) determines the influence of the smoothing term F_s in the optimization. In all our examples we started with a high value of λ in the first iteration and let the influence of the smoothing term fade out in the later iterations ($\lambda \rightarrow 0$). In this way unwanted foldovers or loops of the active curve can be avoided in the beginning of the deformation process. As the active curve comes closer to the model shape the smoothing term gets less important.

   c. With $\mathbf{c}_i$ from the previous step, we replace the control points $\mathbf{d}_i$ by $\mathbf{d}_i^* = \mathbf{d}_i + \mathbf{c}_i$.

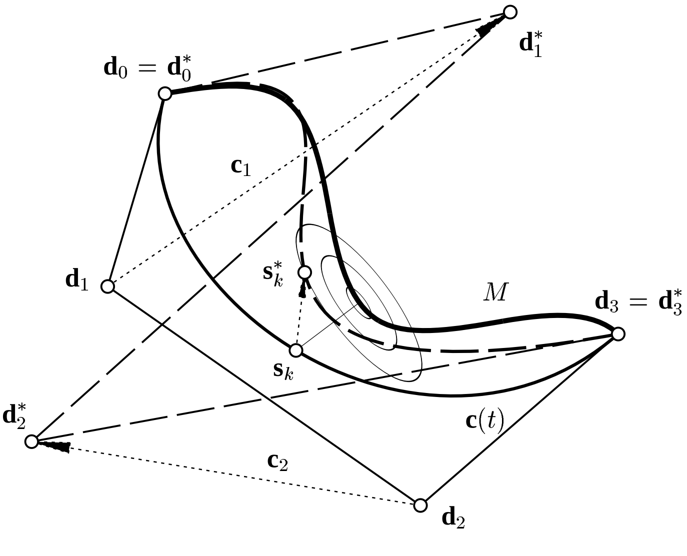

Figure 2. One step in the curve approximation procedure. The curve M is approximated by a B-spline curve.

Fig. 2 illustrates the algorithm. The model shape M is a curve which is to be approximated by a B-spline curve.

Fig. 2 shows an initial position of the B-spline curve $\mathbf{c}(t)$, with control points $\mathbf{d}_i$, and the updated B-spline curve, with control points $\mathbf{d}_i^*$, after one iteration step. For one of the sample points $\mathbf{s}_k = \mathbf{c}(t_k)$ the local quadratic approximant F_d^k of the squared distance function is indicated by three of its level sets, which are concentric ellipses.

Let us briefly discuss the choice of the *initial curve*. In all our tests it turned out to be not critical. Apparently the choice of the initial curve $\mathbf{c}^0$ is safe if $\mathbf{c}^0$ lies close enough to the model curve M so as not to intersect the medial axis of M. This implies that for each point of $\mathbf{c}^0$ there is a unique closest point on M. The squared distance function is smooth in a tubular neighborhood of M, which includes $\mathbf{c}^0$. In such a case, the initial shape rapidly deforms towards the final solution. However, it is not guaranteed that the solution is a global optimum. The landscape of the function F to be minimized needs to be investigated in more detail in future research. We need to get more information about the typical distribution of local minima. For a first result in this direction, see [12].

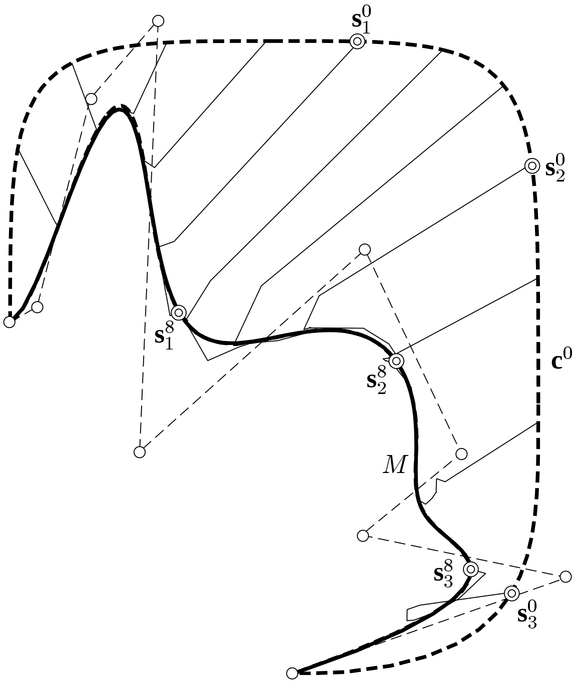

Figure 3. Active curve flow of the B-spline curve $\mathbf{c}^0$ towards target curve M.

Even if the initial curve intersects the medial axis of M, we obtain good results in most cases. For an example see Fig. 3 and Fig. 4. In both figures the model shape M is the same curve (bold solid line), but different initial positions $\mathbf{c}^0$ (bold dashed line) of the active curve $\mathbf{c}$ have been chosen. Both of the initial curves are very rough approximants of M and have intersection points with the medial axis of M (which is not plotted). Nevertheless, our iterative algorithm converges to a practical solution in both cases. The final position $\mathbf{c}^8$ of the approximating curve $\mathbf{c}$ is obtained after eight iterations (dashed curve near M) and its deviation from M is hardly perceptible in the figures.

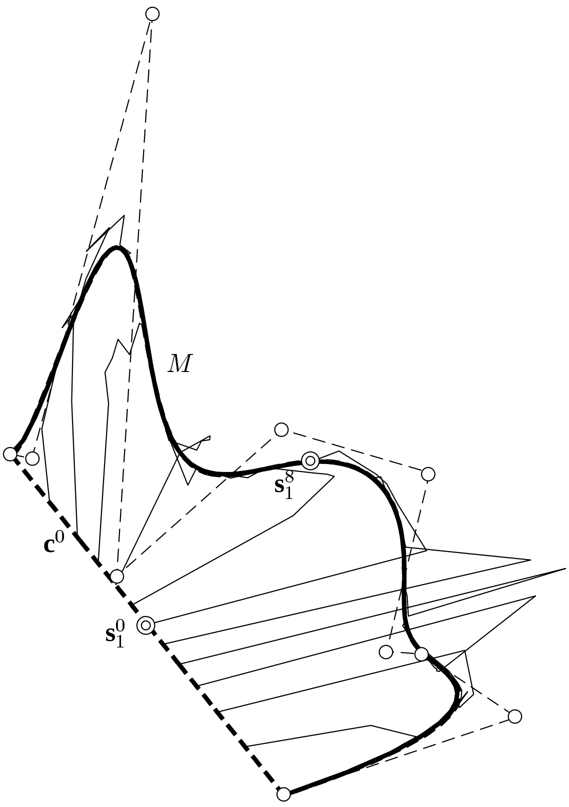

Figure 4. Active curve flow of the B-spline curve $\mathbf{c}^0$ towards target curve M.

The two solutions of Fig. 3 and 4 differ from each other, which can be best observed by means of the control polygons (thin dashed lines) of the final approximating curve $\mathbf{c}^8$. The fact, that we obtain different approximating curves is to be expected because the final result depends on the initial position $\mathbf{c}^0$ of the active B-spline curve $\mathbf{c}$. In general, we have to expect a certain number of local minima which are difficult to avoid.

In Fig. 3 and 4 also the paths of several sample points $\mathbf{s}_k = \mathbf{c}(t_k)$ are depicted, from the initial position $\mathbf{s}_k^0$ to the final position $\mathbf{s}_k^8$. It can be observed clearly that the sample points $\mathbf{s}_k$ of the active curve are not simply moved towards their closest point on M. Many of the sample points move tangentially to M, especially in later steps of the iterative algorithm.

3.1. Degree reduction

As a first example of an application of this method we deal with the problem of *approximate degree reduction*. There are several contributions on this topic, see e.g. [12, 15, 16]. For Bézier curves, the best degree reduction by one degree with respect to L_p-norm ($p = 1, 2, \infty$) has been derived by Eck [16]. In that paper explicit solutions are given, also for the degree reduction with C^q endpoint interpolation conditions. See Fig. 5 for the degree reduction of a Bézier curve $\bar{\mathbf{c}}$ of degree six by a Bézier curve $\mathbf{c}$ of degree four. The control points are denoted by $\bar{\mathbf{d}}_i$ and $\mathbf{d}_i$, respectively. In this example the L^2-norm was used, i.e.,

$$\|\mathbf{c} - \bar{\mathbf{c}}\|_2 = \left(\int_0^1 |\mathbf{c}(t) - \bar{\mathbf{c}}(t)|^2 dt \right)^{1/2}$$

was minimized with C^0 boundary constraints (endpoint interpolation). Note that because of the reduction of the degree by two, Eck's best approximation scheme had to be applied twice (degree $6 \to 5 \to 4$) and we obtain only a suboptimal solution.

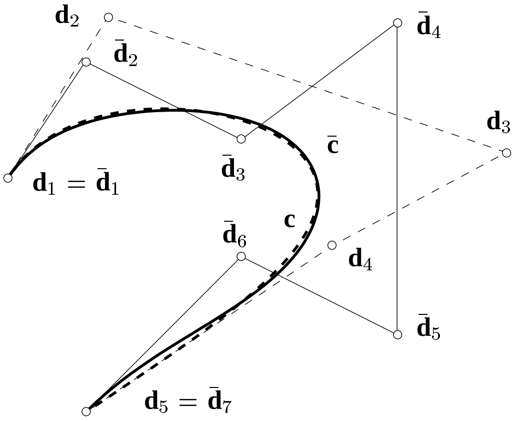

Figure 5. Best degree reduction according to Eck [16]. Bézier curve $\bar{\mathbf{c}}$ of degree 6 (solid line) is approximated by Bézier curve $\mathbf{c}$ of degree 4 (dashed line).

The general curve approximation scheme outlined in the present paper does not use the information that $\bar{\mathbf{c}}$ is a Bézier curve, or any other 'additional' information. If we are interested in the distance between curves as sets (not in the L^2 difference as functions), our algorithm yields much better results than the method of [16].

Fig. 6 and Fig. 7 show the result of our active curve approximation scheme applied to the example of Fig. 5. In Fig. 6 the control points $\mathbf{d}_i$ of the approximating Bézier curve of degree four have been chosen to lie on the target curve $\bar{\mathbf{c}}$. In general this yields a sufficiently good initial position $\mathbf{c}^0$ of the active Bézier curve $\mathbf{c}$. Fig. 7 shows the final result after just five iterations of the active curve flow.

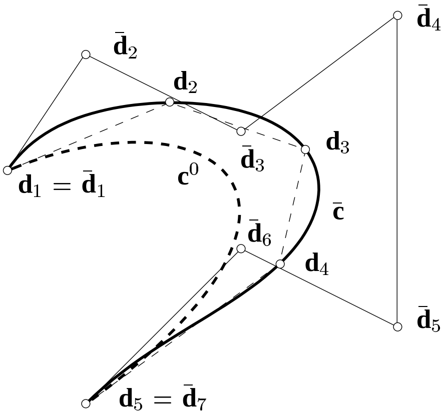

Figure 6. Bézier curve $\bar{\mathbf{c}}$ of degree 6 (solid line) and initial position $\mathbf{c}^0$ of approximating Bézier curve of degree 4 (dashed line).

Let us define the distance of the point set $\mathbf{c}$ to the point set $\bar{\mathbf{c}}$ by

$$d_{\mathbf{c},\bar{\mathbf{c}}} = \max(\|\mathbf{c}(t) - \bar{\mathbf{c}}(\sigma(t))\|), \tag{12}$$

where $\bar{\mathbf{c}}(\sigma(t))$ is the point of $\bar{\mathbf{c}}$ closest to $\mathbf{c}(t)$. Compared to the result of Eck's approximation the distance $d_{\mathbf{c},\bar{\mathbf{c}}}$ is reduced from 0.0985 (Fig. 5) to 0.0186 (Fig. 7), i.e., by a factor of 0.189. Similar results have been obtained for several other examples.

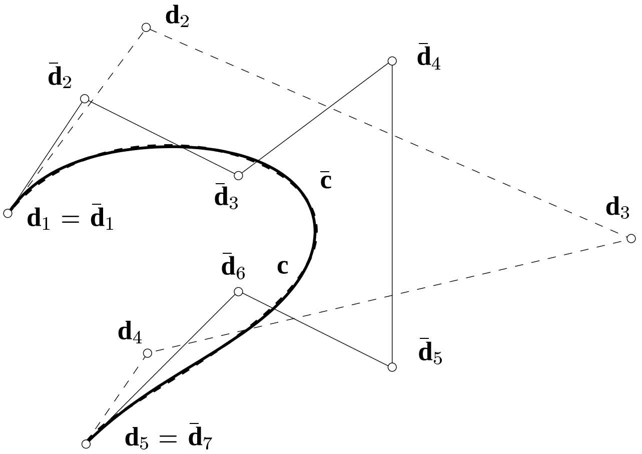

Figure 7. Bézier curve $\bar{\mathbf{c}}$ of degree 6 (solid line) and final position of approximating Bézier curve $\mathbf{c}$ of degree 4 (dashed line).

3.2. Offset curve approximation

The present concept of an active curve under the influence of the squared distance function of a model curve is also well suited for *offset curve approximation*. This is a widely investigated topic since offset curves have many applications (NC machining,...), see e.g. [23, 35] for surveys on offset curves.

For a point $\mathbf{p} \in \mathbb{R}^2$, we have Eq. (4) to describe a quadratic approximant of the squared distance function of a curve $\mathbf{c}(t)$, expressed in the principal frame $\mathbf{e}_1, \mathbf{e}_2$ at the normal footpoint. Let us consider **c**'s one-sided offset curve $\mathbf{c}^*(t) = \mathbf{c}(t) + \alpha \cdot \mathbf{e}_2(t)$ at distance α. In corresponding points $\mathbf{c}(t)$ and $\mathbf{c}^*(t)$ the principal directions $\mathbf{e}_i, i = 1, 2$, and the curvature center **k** are the same. If points have coordinates $\mathbf{p} = (0, 0, d), \mathbf{k} = (0, 0, \rho)$ with respect to the principal frame at $\mathbf{c}(t)$, then these points have coordinates $\mathbf{p} = (0, 0, d - \alpha)$, and $\mathbf{k} = (0, 0, \rho - \alpha)$ with respect to the principal coordinate frame at $\mathbf{c}^*(t)$. The second order Taylor approximant of the squared distance function of the offset curve $\mathbf{c}^*(t)$ at a point **p** is expressed (with respect to the principal frame at the normal footpoint) via

$$F_d^*(x_1, x_2) = \frac{d - \alpha}{d - \rho} x_1^2 + x_2^2.$$

With the quadratic approximant (4) of the squared distance from $\mathbf{c}(t)$ it is therefore simple to derive the corresponding quadratic approximant for its offset curve $\mathbf{c}^*(t)$ at distance α.

Fig. 8 shows an example where a closed curve is given as input shape. Two of its inner and three of its outer offset curves have been approximated by active cubic B-spline curves. These offset curve approximations and their respective control polygons are depicted in Fig. 8.

If an offset curve intersects the medial axis of the original shape, it develops a corner point, i.e., a tangent discontinuity, at this intersection point. This situation arises in our example for the outer two of the offset curves (see Fig. 9 for a closeup). In the implementation of the example of Fig. 8 and 9 the corner points have not been treated in any particular way. The computed offset curves just develop points of very high curvature where the corner points should lie. It is straightforward to detect such points of high curvature and enforce a tangent discontinuity via multiple insertion of an appropriate knot value. In this way corner points can be modelled exactly.

Note that our active curve approach as presented here uses parametric curves and does not allow topological changes. Thus it is not possible to approximate those inner offsets of our model curve which consist of two closed curves. Detection of a topology change and the corresponding adaption of the active curve is possible [13].

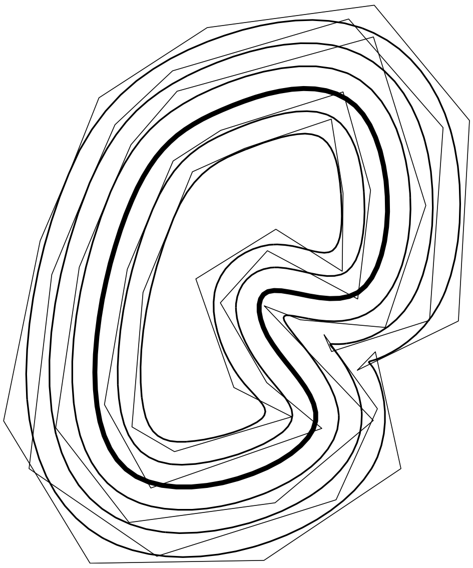

Figure 8. Closed planar curve and five of its offset curves approximated by active B-spline curves.

3.3. Approximation of a helix segment by a B-spline curve

For the illustration of *space curve approximation* by active B-spline curves we have chosen a helix segment as model shape M, see Fig. 10. A helix has constant curvature and torsion and possesses many applications in computer aided geometric design, kinematics, and computer graphics. Despite its geometric simplicity a helix segment cannot be exactly represented as a polynomial or rational curve. Thus there are many contributions on the approximation of a helix segment by (rational) Bézier or B-spline curves (see e.g. [38, 50]).

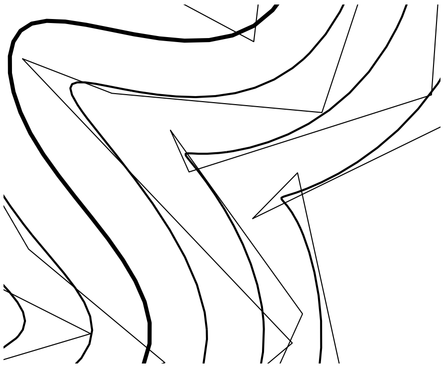

Figure 9. Detail of Fig. 8.

Fig. 10 shows the approximation of a helix segment by an active B-spline curve composed of four cubic segments. The initial position of the active curve was chosen as the straight line connecting the endpoints of the helix segment. The deviations of the approximating B-spline curve from the helix are depicted in Fig. 10, exaggerated by a factor of 500.

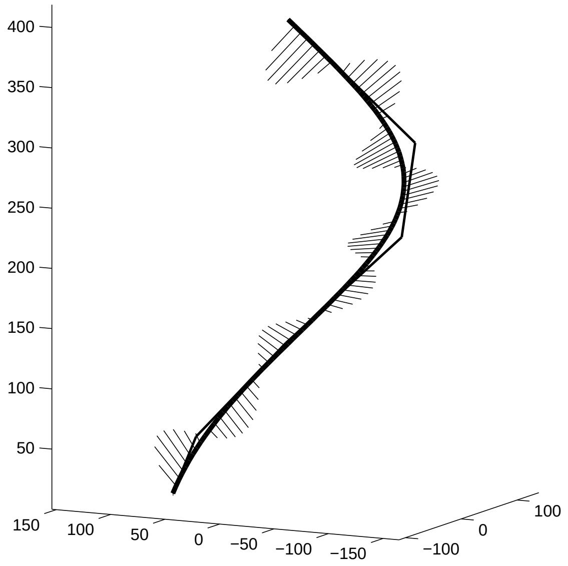

Figure 10. Approximation of a helix with a cubic B-spline, and visualization of error vectors of the approximation, exaggerated by a factor of 500.

4. Approximation with an active surface

Our approach to curve approximation has a straightforward extension to surface approximation.

The active surface model we are using shall be of the form (1), so that surface points to given parameter values depend on the control points in a linear way. Thus, we could also use a subdivision surface.

The quadratic function we are minimizing in each iteration step again consists of a distance part, set up via local quadratic approximants of the squared distance function, and a regularization term. Instead of repeating the algorithm, let us have a look at some details, improvements, and refinements, which are important for a successful implementation of the proposed method.

1. There are similar considerations about the choice of the *initial shape* as in the curve curve case. One difference concerns topology. For curves, even in parametric representation, the detection of a topology change and the corresponding adaption of the active curve is possible [13]. For surface approximation, the initial shape must already exhibit the *correct topology*. A change in topology is hardly possible with a parametric surface. It is simple, however, if one views the active surface as level set of a trivariate function; this is the approach taken in the level set method [53] which has been applied to surface approximation by Zhao et al. [61, 62].

2. One has to impose appropriate *boundary conditions*. For example, we may want to fix the vertices of a surface patch or want to approximate boundary curves of the model shape. One way to do this is to apply the curve analogue of the present method in a first step; then we keep corresponding control points fixed in the surface approximation procedure. However, it can be possible to reach an overall better surface quality by sacrificing some accuracy at the boundary. In this case it is better to add the functional for boundary approximation as a penalty term to F of equation (11). An example is shown in Fig. 12.

3. With totally unrestricted flow, the active surface may shrink to a single point of the model surface and then in a trivial way yield zero approximation error. Strategies for *shrinking avoidance* depend on the special situation, and usually involve appropriate boundary conditions which avoid shrinking, or an appropriate quadratic penalty function F_p added to F in equation (11). For a closed surface, we can start with an initial shape which lies entirely outside the model shape M. The active surface then deflates towards the model shape, if we forbid that it enters the interior of M.

4. If we have an active B-spline surface or another parametric surface and get model points $\mathbf{s}_k$ by evaluation, it is not necessary to keep the parameter values (u_k, v_k), at which we evaluate, fixed. An adaptive evaluation which guarantees a nearly uniform distribution over the active surface, or one which emphasizes especially important regions with help of more model points, will be useful. Moreover, we can introduce further knots and thus more control points during the algorithm if the desired accuracy cannot be achieved with a coarser model. One sees that the method naturally supports a *multiresultion modeling strategy*.

5. The assumption of *nonnegativity* of the approximants F_d^k of the squared distance function of M can be avoided if the change of control points (and thus model points) is restricted. This can be achieved by adding a term to F which expresses the distance of the new

control points to the old ones, e.g. $\sum ||\mathbf{c}_i||^2$. To make sure that the corresponding model points $\mathbf{s}_k$ do not move outside the region where their respective local quadratic approximants are positive, we have to solve a constrained minimization problem. Such algorithms are known in optimization as *trust regions algorithms* [28].

6. For F_s we can use any quadratic smoothing functional, which may change in each iteration step. Thus we can also build Greiner's method for the minimization of nonlinear fairness functionals [19] into our surface approximation technique.

4.1. Approximation with B-spline surfaces

In the following we give an example for the approximation of a given surface by a B-spline surface. In this example the initial position of the active B-spline surface patch (dark gray) is chosen as the bilinear patch connecting the boundary vertices of the model surface (light gray), see Fig. 11. The result of our algorithm after only 7 iterations is shown in Fig. 12. As a boundary condition, the vertices of the model patch have been fixed. The objective function that guides the active surface flow has been chosen as a weighted sum of the quadratic functionals for boundary and surface approximation together with a quadratic smoothing term.

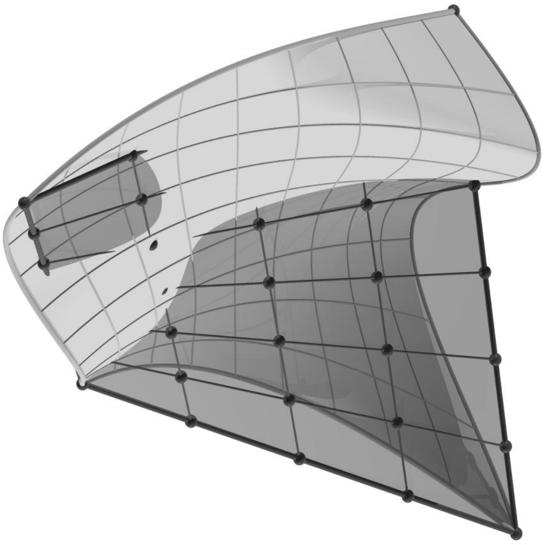

Figure 11. Model shape (light gray) and initial position of approximating B-spline surface (dark gray).

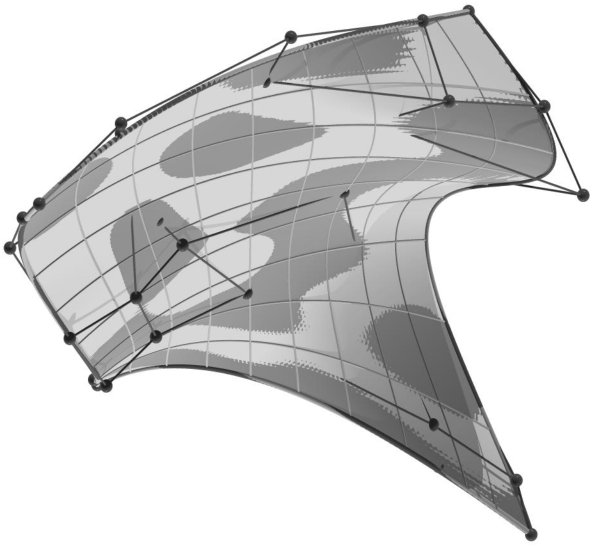

Figure 12. Model shape (light gray) and final position of approximating B-spline surface (dark gray) with boundary curve approximation.

4.2. Approximation of offset surfaces

Offset surface approximation is a widely investigated topic, see e.g. [23, 35, 43]. It has been noted in section 3.2 that our present concept of active curves is applicable to offset curve approximation. The same holds true for the approximation of offset surfaces by active surfaces.

Let us consider $\mathbf{s}$'s one-sided offset surface $\mathbf{s}^*(u,v) = \mathbf{s}(u,v) + \alpha \cdot \mathbf{e}_3(u,v)$ at distance α. In corresponding points $\mathbf{s}(u,v)$ and $\mathbf{s}^*(u,v)$ the principal directions $\mathbf{e}_i, i = 1,2,3$, and the two principal curvature centers are the same. If a point $\mathbf{p} \in \mathbb{R}^3$ has coordinates $\mathbf{p} = (0,0,d)$ with respect to the principal frame at $\mathbf{s}(u,v)$, then this point has coordinates $\mathbf{p} = (0,0,d-\alpha)$ with respect to the principal coordinate frame at $\mathbf{s}^*(u,v)$.

The second order Taylor approximant of the squared distance function of the offset surface $\mathbf{s}^*(u,v)$ at a point $\mathbf{p}$ is expressed (with respect to the principal frame at the normal footpoint) via

$$F_d^*(x_1,x_2,x_3) = \frac{d-\alpha}{d-\rho_1}x_1^2 + \frac{d-\alpha}{d-\rho_2}x_2^2 + x_3^2.$$

5. Variational problems for curves on surfaces

Let us now discuss the computation of surface curves which are solutions to certain variational problems. The general idea here is to use the embedding space for the evolution of the active contour. In an iterative procedure, we solve the given variational problem for a curve on a surface Φ by minimizing a functional F which is composed of two essential components. The first component F_s is the functional to be minimized within the variational problem. For example, in case of a geodesic curve, it is the total arc length. The second component expresses the distance of the curve to the surface Φ with help of local quadratic approximants of the squared distance function to Φ. Thus, as long as F_s is quadratic in the unknowns, or can be replaced by a quadratic function in the unknowns, we are within the framework discussed before. We will present three examples, the third of which — concerning motion design — is a higher dimensional one and is just briefly outlined. It will be described in more detail in a subsequent paper.

In our algorithms, the final curve will be given as a parametric curve **c** which lies very close to the surface. Thus, our algorithm computes, within the desired accuracy, the *shape* of the curve. This may be sufficient. If one wishes to have the preimage curve in the parameter domain of the surface Φ, we can project a sufficient number of points of **c** onto Φ, compute their preimages in the parameter domain and then fit a B-spline curve $\mathbf{c}_p$ to the resulting point set. The image of $\mathbf{c}_p$ under the surface parameterization then is the final result.

5.1. Geodesics

The computation of the shortest path (geodesic) between two points on a surface is a classical one, and also of importance in various problems of CAD and geometric modeling (see e.g. [43]). Typically, one uses the second order differential equations for geodesics known from differential geometry, and then solves the corresponding boundary value problem. Boundary value problems, however, are not so simple to solve and numerical strategies, such as shooting methods, have to be employed [43]. In Computer Vision and Image Processing, geodesics have been considered not only on surfaces in 3-space, but also on general Riemannian manifolds. The metric then depends on the application, e.g. on the texture in an image. Recent algorithms for the fast computation of the distance function from a point **p** in a manifold, a tringulation or even a point cloud [30, 31, 32, 37], are used to compute the shortest path between **p** and any other point **q** in the manifold; one just follows the gradient flow of the distance function. Clearly, this requires the computation of a bivariate function (at least in some neighborhood of the expected geodesic). Conceptually, this is similar to a shooting approach.

Within our framework, we can proceed as follows. Given two points **p** and **q** on a parametric surface, we take an initial shape $\mathbf{c}(t)$ of an active contour, e.g., a B-spline curve representing the straight line segment between **p** and **q**, or a B-spline curve which approximates a surface curve from **p** to **q**. The curve $\mathbf{c}(t)$ is evaluated at a sequence of parameters t_k and we obtain points

$$\mathbf{s}_k = \mathbf{c}(t_k).$$

Now, nonnegative quadratic approximants F_d^k of the squared distance function of Φ are computed at the points $\mathbf{s}_k$. Displacement vectors $\mathbf{c}_i$ for the control points $\mathbf{d}_i$ are computed as minimizers of

$$F = \sum_{k=1}^{N} F_d^k(L_k(\mathbf{d}_1 + \mathbf{c}_1, \ldots, \mathbf{d}_n + \mathbf{c}_n)) + \lambda F_s. \quad (13)$$

As before, $L_k(\cdot)$ expresses the linear dependence of $\mathbf{s}_k$ on the control points $\mathbf{d}_i$. F_s could be taken as the arc length

$$F_s = \int \|\dot{\mathbf{c}}(t)\| dt. \quad (14)$$

This would not yield a quadratic function in the unknown displacement vectors $\mathbf{c}_i$ of the control points. The Cauchy-Schwartz inequality shows easily that minimization of

$$F_s = \int \|\dot{\mathbf{c}}(t)\|^2 dt \quad (15)$$

yields precisely the geodesics, but parametrized with constant velocity (cf. [7], p. 307, or [39], p. 70). This functional now is quadratic in the unknowns $\mathbf{c}_i$ and thus our concept is fully applicable to the approximate computation of geodesics.

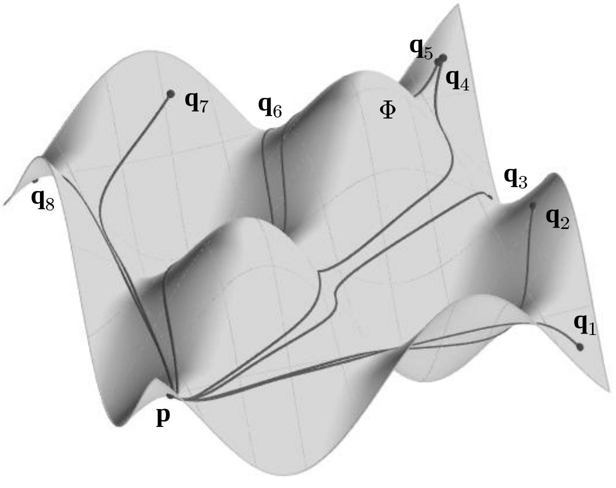

Figure 13. Geodesic curves from $\mathbf{p}$ to several points $\mathbf{q}_i$ on a given surface Φ.

An example of our algorithm is given in Fig. 13. Several geodesic curves emanating from a surface point **p** to surface points $\mathbf{q}_i$ have been computed. For each of the geodesics the initial position of the active curve has been chosen as the straight line connecting $\mathbf{p}$ and $\mathbf{q}_i$. The function (13) has been minimized with F_s from (15).

It is important to note that our method of minimizing (15) under the constraint that $\mathbf{c}$ lies within a given tolerance to the given surface Φ is a penalty function approach. We usually obtain a good approximation of a geodesic, but we can improve it as follows: We replace $\mathbf{c}$ by a polygonal approximation. This polygon is now considered active and moves such that its vertices remain on Φ or close to Φ and also a discrete version of (14), (15), or any equivalent functional is minimized. One particular example is the following: We consider the polygon as a rope and exert equal forces at both ends. The forces in each segment will be of the same magnitude, and unless the vector sum of the two forces acting in a vertex is orthogonal to the surface, this vertex will be moved tangentially to the surface. A simple iterative procedure will produce a discrete geodesic polygon, which fulfills a discrete version of the differential equation of geodesics.

5.2. Spline curves on surfaces

Spline curves in $\mathbb{R}^d$ which interpolate or approximate a sequence of points are often computed as solutions of a variational problem. The most famous example is that of natural cubic splines which minimize the L^2 norm of the second derivative,

$$F_s = \int \|\ddot{\mathbf{c}}(t)\|^2 dt, \tag{16}$$

subject to the given interpolation conditions $\mathbf{p}_j = \mathbf{c}(t_j),\ j = 1, \dots, M$. Other well-known examples are cubic smoothing splines, splines in tension and ν-splines [23].

We now discuss interpolating or approximating spline curves $\mathbf{c}$ on a given surface Φ. Thus, also the given points $\mathbf{p}_j$ lie on Φ. Unlike in the classical case, the solution cannot be described anymore in an explicit way (see e.g. [57]).

The solution concept is simple: we will work with an active curve from a space of curves all of which satisfy the given interpolation constraints. With curves of this space, we then minimize a function as in (13), however with F_s from (16) or with an F_s which appears in another variational spline formulation (smoothing spline, splines in tension, ...).

We still have to describe the choice of an appropriate space of splines. One may simply compute a cubic spline interpolating the points $\mathbf{p}_j$ at (estimated or given) parameter values t_j. This spline $\mathbf{c}^0$ is completely determined by this input and there would be no flexibility anymore to move it towards the given surface Φ. There are several simple ways to achieve more flexibility:

1. We may formally raise the degree of the segments. For example, we can raise the degree to five and express them in Hermite form. The unknowns are then the first and second order derivative vectors at the given points $\mathbf{p}_j$. In the original position they are in such a special position that the spline curve is a cubic C^2 spline, but this will change during the optimization algorithm.

2. We may split each segment of $\mathbf{c}^0$ between consecutive points $\mathbf{p}_j, \mathbf{p}_{j+1}$ into several segments (knot insertion). If we stay with cubics, it is convenient to use a B-spline representation. For such a B-spline curve between two input points, the end points $\mathbf{d}_0^j, \mathbf{d}_m^j$ are fixed (the given input points $\mathbf{p}_j, \mathbf{p}_{j+1}$). The control points next to the end points, $\mathbf{d}_1^j, \mathbf{d}_{m-1}^j$ depend linearly on the (unknown) first derivative vectors in the endpoints, and the remaining B-spline control points can move freely (are unknowns). For C^2 join at $\mathbf{p}_j$, also the control points $\mathbf{d}_2^j$ and $\mathbf{d}_{m-2}^j$ of the B-spline depend linearly on the unknown first and second derivative vectors at $\mathbf{p}_j, \mathbf{p}_{j+1}$.

3. We can combine splitting and degree elevation.

For an example of the construction of a spline curve on a surface Φ see Fig. 14. The initial curve $\mathbf{c}^0$ is a natural cubic spline through the surface points $\mathbf{p}_j$ and does not lie on the given surface. By knot insertion each cubic segment between $\mathbf{p}_j$ and $\mathbf{p}_{j+1}$ has been splitted into three cubic segments. The resulting free parameters have been used in the active curve flow of the cubic C^2 spline curve $\mathbf{c}$ towards the surface Φ.

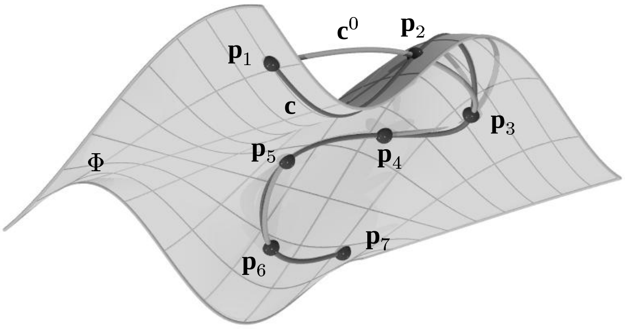

Figure 14. Initial position $\mathbf{c}^0$ and final position $\mathbf{c}$ of a cubic spline curve which lies very close to the surface Φ and interpolates given surface points $\mathbf{p}_j$.

It is possible that the initial cubic spline is far away from the given surface. In that case it is advisable to start with another curve. A simple choice is the restriction to a spline which is tangent to the given surface at the given points $\mathbf{p}_j$. Using a basis $\{\mathbf{q}_j, \mathbf{r}_j\}$ of the tangent plane of $\mathbf{p}_j$, the first derivative vector $\dot{\mathbf{c}}_j$ is of the form

$$\dot{\mathbf{c}}_j = \lambda_j \mathbf{q}_j + \mu_j \mathbf{r}_j.$$

The dependence of $\dot{\mathbf{c}}$ on the unknowns λ_j, μ_j is linear. We then minimize F_s within the space of quintic C^2 splines with these first derivative vectors and unknown second derivative vectors at $\mathbf{p}_j$. This is again the minimization of a quadratic function and yields an appropriate starting shape for active contour propagation towards a spline curve on the given surface Φ.

A fine tuning of the shape analogous to the computation of geodesics is possible, but may be unnecessary. Since a functional such as (16) is just an approximation of the bending energy, we minimize a simplified version anyway. However, the minimization of a geometric functional with a method as in Greiner's work [19] can also be included in the present method. Then, fine tuning of the shape might be more interesting. We will report on such algorithms in a subsequent paper.

5.3. Motion design

The following problem appears in computer animation and robot motion planning: Given a set of positions Σ_i of a moving rigid body Σ, compute a smooth motion of the body which assumes the given positions Σ_i at given time instances t_i. This problem has received a lot of attention in various scientific communities (Computer Graphics, CAGD, Robotics, Computational Geometry). We do not review the literature, but just point to a survey paper [48] and to a recent paper [21], where a new concept is presented which is based on the following idea.

It is simple to solve the problem with an affine motion, i.e., to at first admit affine distortions of the moving body. This is so since we may choose any *linear* curve interpolation or approximation scheme. Moreover, we choose a set of four independent points (called feature points $\mathbf{f}^k$ henceforth) on the body Σ and compute their locations $\mathbf{f}^k_i$ at time instances t_i. This results in 4 sequences of homologous points to which we apply the curve scheme. Thereby, we obtain four curves $\mathbf{f}^1(t), \ldots, \mathbf{f}^4(t)$. For each t, the four points $\mathbf{f}^1(t), \ldots, \mathbf{f}^4(t)$ may be considered as image points of $\mathbf{f}^1, \ldots, \mathbf{f}^4$ under an affine map $\alpha(t)$. Applying $\alpha(t)$ to Σ we obtain an affinely distorted copy $\Sigma(t)$ of the body Σ and thus an affine motion which interpolates or approximates the given positions Σ_i. By the linearity of the chosen curve scheme, it does not matter at all which four feature points we select. Another choice would lead to the same affine motion.

What we eventually want, however, is a rigid body motion. In [21] it is suggested to proceed as follows: For each t, approximate the affine map $\alpha(t)$ by a Euclidean motion (rigid body transformation) $\beta(t)$. As a quality measure for the fit, the chosen feature points (or more points on the body) are considered. The sum of squared distances between the images of these points under α and β is minimized. This is a special case of a well-known registration problem in Computer Vision [22], and has been recently investigated in depth by J. Wallner [58]. Thus, one knows exactly how to explicitly compute the best approximation of $\alpha(t)$ by a Euclidean motion $\beta(t)$ and under which circumstances the solution is unique.

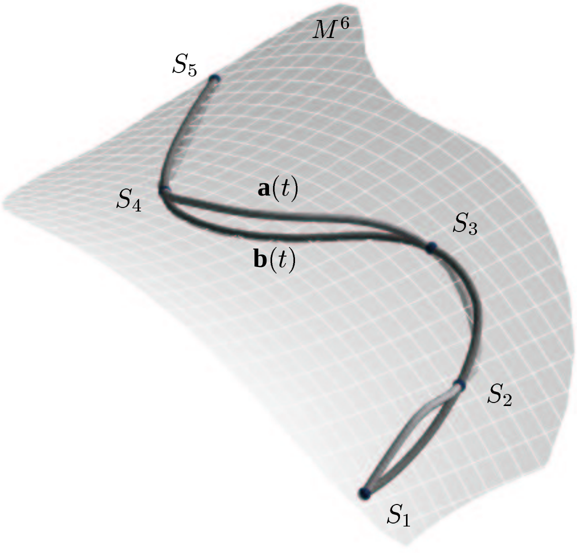

Figure 15. The affine motion corresponds to a curve $\mathbf{a}(t)$ in A^{12} and the designed motion $\mathbf{b}(t)$ is the orthogonal projection of $\mathbf{a}(t)$ onto the 6-dimensional manifold M^6 of Euclidean motions.

Let us view the approach to motion design which has been just outlined in a more geometric way (Fig. 15). To each affine map α in 3-space we may associate a point in 12-dimensional affine space A^{12}. For example, we may just collect the coordinates of the image points of the four chosen feature points under α. The image points of Euclidean motions form a 6-dimensional submanifold $M^6 \subset A^{12}$. The input positions Σ_i correspond to points S_i on M^6. The affine motion $\alpha(t)$ corresponds to a curve $\mathbf{a}(t)$ which interpolates the points S_i, but does not lie on M^6. The measurement of distances between two affine maps with help of the sum of squared distances between the images of selected feature points is equivalent to the introduction of a

Euclidean metric in A^{12}. Thus, we also have an orthogonality in A^{12}. The way in which we compute a rigid body motion $\beta(t)$ from the affine motion $\alpha(t)$ outlined above corresponds to the orthogonal projection $\mathbf{b}(t)$ of the curve $\mathbf{a}(t)$ onto M^6. However, if the original affine motion has been obtained with a variational formulation, the new solution $\mathbf{b}(t)$ is in general not the minimizer of that functional under the constraint of lying in M^6.

This view shows immediately how to apply the concepts of the present paper to motion design. We have an active curve in A^{12} moving towards the manifold M^6 in a way such that a given functional F_s is minimized. For a solution, one needs local quadratic approximants of the squared distance function to M^6, e.g. squared distances to tangent spaces at normal footpoints. The remaining algorithm is pretty much the same as in the previous subsection.

As an example, Fig. 16 shows a motion computed via minimization of the cubic spline functional (16).

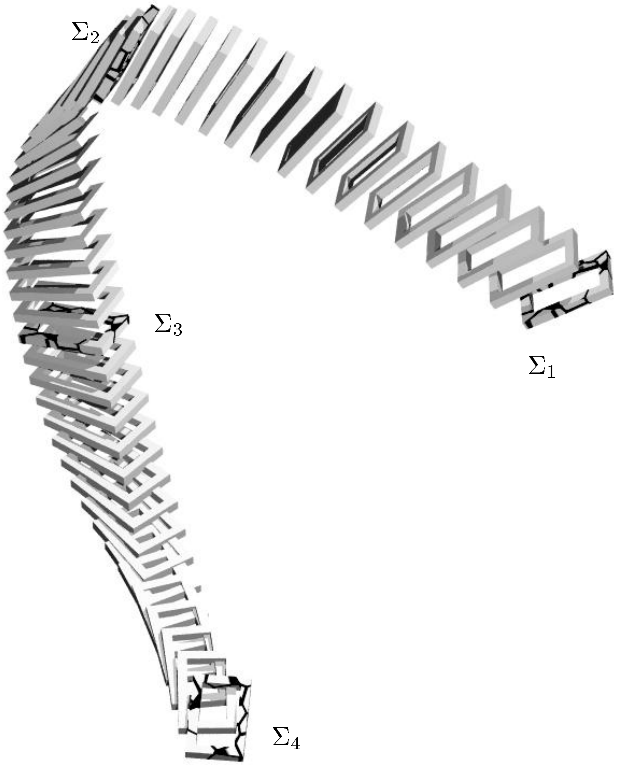

Figure 16. Cubic spline motion interpolating given positions Σ_i.

6. Conclusion and future research

We have presented an active contour model for curve and surface approximation which avoids the parametrization problem. According to the nonlinearity of the approximation problem, this model yields an iterative method: In each iteration step we solve a linear system of equations, which arises from the minimization of a quadratic function. The new idea is that in this quadratic function we use local quadratic approximants of the squared distance function to the model shape which shall be approximated. In the present paper, we just outlined the concept and applied it to a few typical problems such as offset approximation, degree reduction and conversion of arbitrary representations into B-spline form. Moreover, we showed how the computation of geodesics, of spline curves on surfaces and of interpolating or approximating smooth rigid body motions can be handled. There are many possibilities for extensions and future research:

- The convergence behaviour of the present method, conditions on the choice of the initial shape and — if possible — conditions for reaching a global optimum should be investigated.

- We need more research on quadratic approximants of d^2. In particular, an appropriate space partitioning structure with cells carrying the approximants needs to be developed. Also, a hierarchical representation of the squared distance function with such a spatial data structure would be important.

- The concept is applicable to *approximation with subdivision surfaces*. One can use an initial shape as in [34], but other choices and an appropriate handling of details require a lot of future research.

- An interesting extension concerns the incorporation of *shape constraints* such as convexity. For example, we can use the sufficient linear convexity conditions which have been derived by B. Jüttler for surface fitting with convex tensor product splines [24, 25]. In our framework, we would then have to solve a quadratic programming problem in each iteration step.

- Special interpolation and approximation problems which appear in geometric modeling, for example the design of skinning surfaces, may be addressed with the new concept.

- The new approach to motion design needs to be studied in more depth. Moreover, we want to investigate motions constrained by a gliding surface pair and we would like to compute motions which avoid given obstacles. This can then be used for motion planning in connection with 5-axis NC machining.

- The present active contour model and its deformability form the lower levels of a so-called *artifical life model* [55]. The evaluation points $\mathbf{s}_k$ can be seen as 'sensors', and the control structure is the layer above that which allows us to coordinate the action of the sensors. Including phyiscal modeling and on top of that behavioral, perceptual and cognitive modeling yields an artificial life model. Those models play an increasingly important role in Computer Graphics [55] and Medical Imaging [20]. We expect that this is also a promising research direction for the creation of 'intelligent' modeling and reverse engineering systems, which include shape understanding abilities.

Acknowledgements

This research has been carried out as part of the project P13938-MAT supported by the Austrian Science Fund (FWF). We would like to thank Johannes Wallner for valuable comments during our work.

References

[1] Bajaj, C., Bernardini, F., Xu, G., Automatic reconstruction of surfaces and scalar fields from 3D scans. SIGGRAPH '95 Proceedings (1995), 193–198.

[2] Besl, P. J., McKay, N. D., A method for registration of 3D shapes, IEEE Trans. Pattern Anal. and Machine Intell. **14** (1992), 239–256.

[3] Blake, A., Isard, M., *Active Contours*, Springer, 1998.

[4] Boissonnat, J.D., Cazals, F., Smooth shape reconstruction via natural neighbor interpolation of distance functions, In: Proc. 16th Ann. ACM Sympos. Comput. Geom. (2000), 223–232.

[5] Bloomenthal, J., ed., *Introduction to Implicit Surfaces*, Morgan Kaufmann Publ., San Francisco, 1997.

[6] Brunnet, G., Hagen, H., Santarelli, P., Variational design of curves and surfaces, Surveys on Mathematics for Industry **3**, 1–27.

[7] do Carmo, M.P., *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[8] Caselles, V., Kimmel, R., Sapiro, G., Geodesic active contours, Intl. J. Computer Vision **22** (1997), 61–79.

[9] Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces: A geometric three-dimensional segmentation approach, Numerische Mathematik **77** (1997), 423–451.

[10] Choi, H.I, Choi, S.W., Moon, H.P.: Mathematical theory of medial axis transform. Pacific J. Math. **181** (1997), 57–88.

[11] Choi, H.I., Choi, S.W., Moon, H.P.: New algorithm for medial axis transform of plane domain. Graphical Models and Image Processing **59** (1997), 463–483.

[12] Degen, W.L.F., Best approximations of parametric curves by splines, In: T. Lyche and L. L. Schumaker, eds., Mathematical Methods in Computer Aided Geometric Design II, Boston, Academic Pres, 1992, pp. 171–184.

[13] Delingette, H., Montagnat, J., Shape and topology constraints on parametric active contours, Computer Vision and Image Understanding **83** (2001), 140–171.

[14] Desbrun, M., Cani-Gascuel, M.-P., Active implicit surface for animation, Graphics Interface (1998), 143–150.

[15] Eck, M., Degree reduction of Bézier curves, Computer Aided Geometric Design **10** (1993), 237–251.

[16] Eck, M., Approximative Degree Reduction in Bernstein-Bézier form, Preprint Nr. 1540, Fachbereich Mathematik, Technische Hochschule Darmstadt, 1993.

[17] Faugeras, O., Gomes, J., Dynamic shapes of arbitrary dimension: the vector distance functions, In: R. Martin, ed., The Mathematics of Surfaces IX, Springer, 2000, pp. 227–262.

[18] Gomes, J., Faugeras, O., Reconciling distance functions and level sets, J. Visual Communication and Image Representation **11** (2000), 209–223.

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

[20] Hamarneh, G., McInerny, T., Terzopoulos, D., Deformable organisms for automatic medical image analysis, Proc. Medical Image Computing and Computer–Assisted Intervention — MICCAI 2001, Springer, pp. 66–76.

[21] Hofer, M., Pottmann, H., Ravani, B., From curve design algorithms to motion design, Technical Report No. 95, Insitute of Geometry, Vienna Univ. of Technology, July 2002.

[22] Horn, B.K.P., Closed form solution of absolute orientation using unit quaternions, Journal of the Optical Society A **4** (1987), 629–642.

[23] Hoschek, J. and Lasser, D., *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, MA., 1993.

[24] Hoschek, J., Jüttler, B., Techniques for fair and shape preserving surface fitting with tensor-product B-splines, In: Shape Preserving Representations in Computer-Aided Geometric Design (J.M. Peña, ed.), Nova Science Publ., Commak, New York, 1999.

[25] Jüttler, B., Surface fitting using convex tensor-product splines, J. Comput. Appl. Math. **84** (1997), 23–44.

[26] Jüttler, B., Felis, A., Least-squares fitting of algebraic spline surfaces, Advances in Computational Mathematics **17** (2002), 135–152.

[27] Kass, M., Witkin, A., Terzopoulos, D., Snakes: Active contour models, Intern. J. Computer Vision **1** (1988), 321–332.

[28] Kelley, C. T., Iterative Methods for Optimization, SIAM, Philadelphia, 1999.

[29] Kimmel, R., Shaked, D., Kiryati, N., Bruckstein, A., Skeletonization via distance maps and level sets, Computer Vision and Image Understanding **62** (1995), 382–391.

[30] Kimmel, R., Kiryati, N., Bruckstein, A., Distance maps and weighted distance transforms, Journal of Mathematical Imaging and Vision **6** (1996), 223–233.

[31] Kimmel, R., Kiryati, N., Finding shortest paths on surfaces by fast global approximation and precise local refinement, Int. Journal of Pattern Recognition and Artificial Intelligence **10** (1996), 643–656.

[32] Kimmel, R., Sethian, J. A., Computing Geodesic Paths on Manifolds, Proceedings of National Academy of Sciences **95** (1998), 8431–8435.

[33] Ma, W., Kruth, J. P., Parametrization of randomly measured points for the least squares fitting of B-spline curves and surfaces, Computer Aided Design **27** (1995), 663–675.

[34] Ma, W., Zhao, N., Catmull-Clark surface fitting for reverse engineering applications, in: Proc. Geometric Modeling and Processing, Hong Kong, 2000, pp. 274–283.

[35] Maekawa, T., An overview of offset curves and surfaces, Comp. Aided Design **31** (1999), 165–173.

[36] Malladi, R., Sethian, J. A., Vemuri, B. C., Shape modeling with front propagation: A level set approach, IEEE Trans. Pattern Anal. and Machine Intell. **17** (1995), 158–175.

[37] Memoli, F., Sapiro, G., Fast computation of weighted distance functions and geodesics on implicit hypersurfaces, Journal of Computational Physics **173**(2) (2001), 730–764.

[38] Mick, S., Röschel, O., Interpolation of helical patches by kinematic rational Bézier patches, Computers and Graphics **14** (1990), 275–280.

[39] Milnor, J., *Morse Theory*, Annals of Mathematics Studies, Vol. 51, Princton Univ. Press, 1963.

[40] Nikolaidis, N., Pitas, I., *3-D Image Processing Algorithms*, Wiley, 2001.

[41] Osher, S., Fedkiw, R., Level set methods: an overview and some recent results, *J. Comp. Physics* **169** (2001), 463–502.

[42] Osher, S. J., Sethian, J. A., Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation, Journal of Computational Physics **79** (1988), 12–49.

[43] Patrikalakis, N. M., Maekawa, T., *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer, Berlin, 2002.

[44] Piegl, L, Tiller, W., *The NURBS book*, Springer Verlag, New York, 1995.

[45] Pottmann, H., Hofer, M., Geometry of the squared distance function to curves and surfaces, Technical report 90, Institute of Geometry, Vienna Univ. of Technology, January 2002. To appear in: Proceedings of 'Mathematics and Visualization 2002', Springer.

[46] Pottmann, H., Leopoldseder, S., A concept for parametric surface fitting which avoids the parameterization problem, Technical Report No. 94, Insitute of Geometry, Vienna Univ. of Technology, June 2002.

[47] Pottmann, H., Wallner, J., *Computational Line Geometry*, Springer, 2001.

[48] Röschel, O., Rational motion design — a survey, Computer Aided Design **30** (1998), 169–178.

[49] Sapiro, G., *Geometric Partial Differential Equations and Image Analysis*, Cambridge Univ. Press, Cambridge, 2001.

[50] Seemann, G., Approximating a helix segment with a rational Bézier curve, Computer Aided Geometric Design **14** (1997), 475–490.

[51] Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[52] Serra, J., Soille, J., eds., *Mathematical Morphology and its Applications to Image Processing*, Kluwer, Dordrecht 1994.

[53] Sethian, J. A., *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.

[54] Siddiqi, K., Tannenbaum, A., Zucker, S.W., A Hamiltonian approach to the eikonal equation, Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 1999, pp. 1–13.

[55] Terzopoulos, D., Artifical life for Computer Graphics, Communications of the ACM **42** (1999), 33–42.

[56] Toriwaki, J., Yokoi, S., Distance transformations and skeletons of digitized pictures with applications, *Progress in Pattern Recognition*, L.N. Kanal and A. Rosenfeld, eds., North Hollang, 1981, pp. 187–264.

[57] Wallner, J., Pottmann, H., Variational spline interpolation, Technical Report No. 84, Institute of Geometry, Vienna Univ. of Technology, 2001.

[58] Wallner, J., L^2 approximation by Euclidean motions, Technical Report No. 93, Institute of Geometry, Vienna Univ. of Technology, 2002.

[59] Warren, J., Weimer, H, *Subdivision Methods for Geometric Design: A Constructive Approach*, Morgan Kaufmann Series in Computer Graphics, San Francisco, 2001.

[60] Zhao, H.K., Chan, T., Merriman, B., Osher, S., A variational level set approach to multiphase motion, J. Comp. Physics **127** (1996), 179–195.

[61] Zhao, H.K., Osher, S., Merriman, B., Kang, M., Implicit, non-parametric shape reconstruction from unorganized data using variational level set method, Computer Vision and Image Understanding **80** (2000), 295–314.

[62] Zhao, H.K., Osher, S., Fedkiw, R., Fast surface reconstruction and deformation using the level set method, Proc. IEEE Workshop on Variational and Level Set Methods in Computer Vision, Vancouver, 2001.

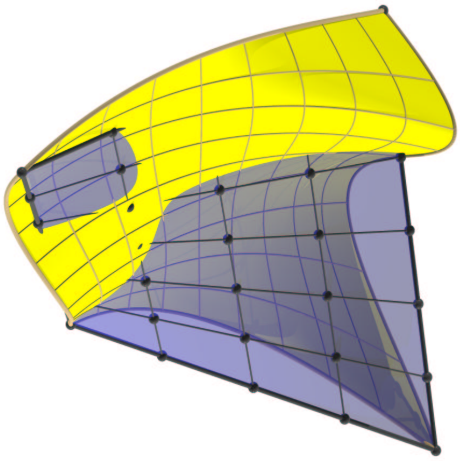

Figure 11. Model shape (yellow) and initial position of approximating B-spline surface (blue).

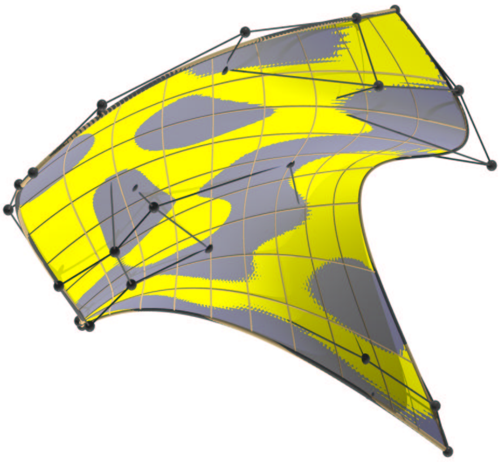

Figure 12. Model shape (yellow) and final position of approximating B-spline surface (blue) with boundary curve approximation.

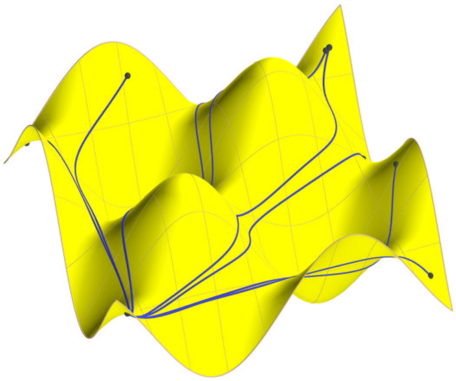

Figure 13. Geodesics on a given surface.

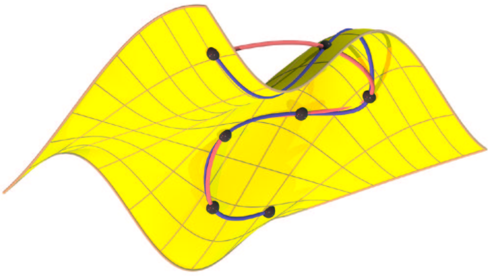

Figure 14. Initial (red) and final (blue) position of an interpolating cubic spline curve which lies very close to the given surface.

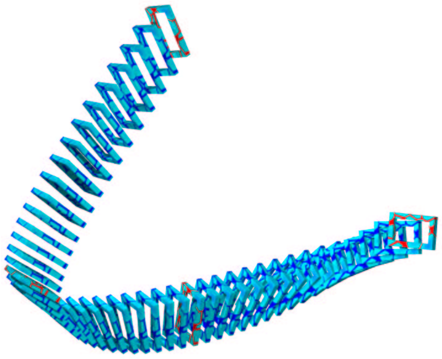

Figure 16. Cubic spline motion interpolating given positions (marked in red).