Approximation Flows in Shape Manifolds

Martin Aigner and Bert Jüttler

Abstract. We consider manifolds of curves and surfaces which are controlled by certain systems of shape parameters. These systems may be given by the control points of a spline curve, the coefficients of an implicit equation, or other parameters controlling the shape. Each system of shape parameters corresponds to a chart of the manifold. In order to fit a curve or surface from such a manifold to given unorganized point data, we define an evolution process which takes an initial solution and modifies it in order to adapt it to the data. We show that this evolution defines a flow on the shape manifold. Consequently, the result of the evolution is independent of the particular choice of the shape parameters / of the chart.

§1. Introduction

In order to fit curves and surfaces defined by parametric representations to given point cloud data, a non–linear optimization problem has to be solved, due to the influence of the parameterization. Several methods for addressing this influence are available (e.g., [3, 8, 10, 11, 12, 13, 16, 17]), ranging from simple techniques, such as 'parameter correction', to the use of geometrically motivated quasi–Newton methods.

Due to the iterative nature of the techniques for non–linear optimization, it is tempting to view the intermediate results as a time–dependent curve or surface which tries to adapt itself to the given points [11, 17]. This is related to the idea of 'active curves' used for image segmentation in Computer Vision [9, 5, 14].

In this paper, we consider the curve and surface fitting problem for a very general class of objects, forming a shape manifold, as described in Section 2. In the third section we present an evolution–based technique for approximating points and discuss its relation to existing methods. Section 4 analyzes the dependence of the evolution result on the choice of the chart, i.e., of the shape parameters describing the object. Finally we show several examples and conclude this paper.

Conference Title
Editors pp. 1–6.

ISBN 0-0-9728482-x-x

§2. Shape Manifolds

In order to keep the notations simple, all results are presented for planar curves. However, they can be generalized to the surface case.

2.1. Manifolds of planar curves

We consider a family of planar curves of the form $\mathbf{c_s}(u)$, where the parameter u varies within the parameter domain $I = [a, b]$. The shape of the curves is controlled by the vector $\mathbf{s}$ which collects the so–called *shape parameters* $\mathbf{s} = (s^1, \ldots, s^n) \in \Omega$ which are allowed to vary within a certain open subset $\Omega \subseteq \mathbb{R}^n$. Except for finitely many parameter values $U^0 = \{u_1^0, \ldots, u_k^0\}$, which are called the vertex parameters, the curve $\mathbf{c_s}(u)$ is assumed to be differentiable and singularity–free, $\mathbf{c}'_{\mathbf{s}}(u) \neq \mathbf{0}$ for $\mathbf{s} \in \Omega$ and $u \in I \setminus U^0$, where the prime indicates the differentiation with respect to u. Moreover, for any constant value of u, the mapping $\mathbf{s} \to \mathbf{c_s}(u)$ should be C^1. In the case of open curves, the end points are treated like vertices.

Clearly, it is always possible to find another system of shape parameters $\boldsymbol{\sigma} = (\sigma^1, \ldots \sigma^n) \in \Omega^*$, which describes the same family of curves,

$$\mathbf{c_s}(.) = \mathbf{c}^*_{\boldsymbol{\sigma}}(.), \quad \text{where} \quad \boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{s}). \tag{1}$$

Consequently, the family of shapes can be equipped with a structure of a *differentiable manifold*, which will be called the *shape manifold* M. The different choices of shape parameters correspond to different charts of the shape manifold. As we shall see later, it may happen that more than one chart is needed in order to cover the entire manifold.

If (1) holds for the shape parameters $\mathbf{s} \in \Omega$ and $\boldsymbol{\sigma}(\mathbf{s}) \in \Omega^*$, then there exist open neighborhoods $N \subseteq \Omega$ and $N^* \subseteq \Omega^*$ of $\mathbf{s}$ and $\boldsymbol{\sigma}$, respectively, such that the transformation between the charts

$$N \to N^* : \quad \mathbf{s} \mapsto \boldsymbol{\sigma}(\mathbf{s}) \tag{2}$$

is bijective, differentiable and regular, i.e., the Jacobian $(\partial\sigma^i/\partial s^j)_{i,j=1,\ldots,n}$ has full rank.

Example 1. We consider all circles in the plane,

$$\mathbf{x_s}(u) = (x_c + r\cos(u), y_c + r\sin(u))^\top, \quad u \in [0, 2\pi] \tag{3}$$

with shape parameters $\mathbf{s} = (x_c, y_c, r) \in \mathbb{R}^2 \times \mathbb{R}^+$. Instead of using the parametric representation (3), they may be described in implicit form $f(x, y) = 0$ with

$$f(x, y) = a(x^2 + y^2) + x + by + c. \tag{4}$$

where the shape parameters $\boldsymbol{\sigma} = (a, b, c)$ satisfy $a > 0$ and $1 + b^2 > 4ac$.

Example 2. In order to fit the bridge–shaped data set shown in Fig. 4, we consider a curve defined by collecting six line segments and two circular arcs, which are pieced together continuously. The set of shape parameters consists of the centers and the radii of the circles parameterized in polar coordinates and the end points of the lines. Furthermore, we need 4 angles in order to describe the positions of the vertices on the circular arcs. Summing up, we obtain 18 shape parameters.

Example 3. We consider all rational cubic curves in the plane which intersect the interior of the unit square. For any two points on the four boundary lines, we choose the standardized (i.e., the weights of the boundary control points are equal to 1) Bernstein–Bézier representation of the curve. Counting all possibilities, this defines $32 = 2 \cdot 4^2$ different charts on this 8–dimensional manifold of shapes. For reasons of numerical stability, one should always choose the Bernstein–Bézier representation of the shortest possible arc (i.e., containing all intersection with the interior of the unit square). Another chart can be defined by using the implicit representation, which has 10 coefficients, and specifying a singular point (three equations, two degrees of freedom). After taking the homogeneity into account, we arrive again at an 8–dimensional manifold.

Remark 1. Note that the parameterization of the curve does not need to be known explicitly, since the definition of the shape manifold requires only the existence of such a parameterization. For instance, closed ovals of implicitly defined curves, where a uniform speed parameterization can be guaranteed to exist, fit into the framework also.

2.2. Vector fields and flows

Recall that a *contravariant vector field* on a manifold obeys the transformation rule

$$v^i = \sum_{j=1}^{n} v_*^j \frac{\partial s^i}{\partial \sigma^j} \tag{5}$$

where $v^i = v^i(s^1, \ldots, s^n)$ and $v_*^j = v_*^j(\sigma^1, \ldots, \sigma^n)$ are the components with respect to the charts defined by the shape parameters $\mathbf{s}$ and $\boldsymbol{\sigma}$, respectively. In the remainder of this paper we shall use Einstein's summation convention, omitting summation signs as in (5). Indices not affected by it will be chosen as Greek characters.

A contravariant vector field $v^i = v^i(s^1, \ldots, s^n)$ defines a *flow*

$$\phi_t : \mathbb{R} \times M \to M : \quad \phi_t : s_0 \mapsto \phi_t(s_0) \tag{6}$$

where $\phi_t(s_0)$ is obtained by solving the differential equation $\dot{s}_i = v^i$ with the initial condition $s(0) = s_0$. Eq. (6) describes the flow in the chart

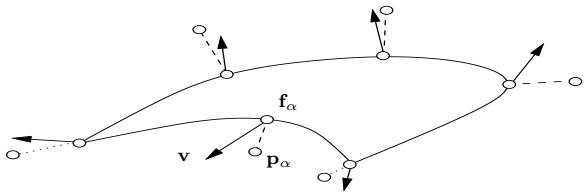

Fig. 1. In order to fit the data, the closest points are attracted by the given points.

determined by the shape parameters $\mathbf{s}$, and the description in any other chart is equivalent to this one. See [6] for a more detailed introduction to vector fields and flows on manifolds.

§3. Fitting by evolution

We assume that an unorganized point cloud $\{\mathbf{p}_\alpha\}_{\alpha=1..N}$ is given, which is to be approximated by a curve from a given shape manifold. More precisely, we whish to find the shape parameters which solve

$$\sum_{\alpha=1}^{N} \min_{u_\alpha \in I} \|\mathbf{p}_\alpha - \mathbf{c}_\mathbf{s}(u_\alpha)\|^2 \to \min_{\mathbf{s}\in\Omega}. \tag{7}$$

In this section we recall the framework of [2] for solving this problem and adapt it to the case of a shape manifold.

In order to identify the best shape parameters, we consider them as smooth functions of an evolution parameter t. Starting from suitable initial values, the shape parameters $\mathbf{s} = \mathbf{s}(t)$ are modified such that the curve moves closer to the data points. In order to achieve this, the data points $\mathbf{p}_\alpha$ attract the associated closest points $\mathbf{f}_\alpha = \mathbf{c}_\mathbf{s}(u_\alpha)$ in a least–squares sense.

During the evolution each point of the curve travels with the velocity

$$\mathbf{v}(u) = \frac{\partial \mathbf{c}_\mathbf{s}(u)}{\partial s^i}\, \dot{s}^i(t). \tag{8}$$

We relate this velocity to the difference vectors $\mathbf{d}_\alpha := \mathbf{p}_\alpha - \mathbf{f}_\alpha$ by minimizing the objective function

$$\sum_{\substack{\alpha=1,\dots,N \\ u_\alpha \notin U^0}} ((\mathbf{v}(u_\alpha) - \mathbf{d}_\alpha)^\top \vec{\mathbf{n}}_\alpha)^2 + \sum_{\substack{\alpha=1,\dots,N \\ u_\alpha \in U^0}} \|\mathbf{v}(u_\alpha) - \mathbf{d}_\alpha\|^2 \to \min_{\dot{\mathbf{s}}\in\mathbb{R}^n}, \tag{9}$$

where $\vec{\mathbf{n}}_\alpha$ denotes the unit normal of the curve in the point $\mathbf{c}_\mathbf{s}(u_\alpha)$, see Fig. 1.

In order to simplify the presentation we assume that none of the u_α is a vertex parameter, i.e., $u_\alpha \notin U^0$, $\alpha = 1, \ldots, N$. For instance, this is clearly the case if $U^0 = \emptyset$. Consequently, the second sum in (8) is not present.

Due to (8), the velocities $\mathbf{v}(u_\alpha)$ depend linearly on the time derivatives of the shape parameters. Consequently, (9) defines a quadratic optimization problem. As a necessary condition for a minimum, the first derivatives of the objective function with respect to the $\dot{s}^i$ vanish. This leads to the linear system

$$\sum_{\alpha=1}^{N} \frac{\partial \mathbf{c_s}(u_\alpha)^\top}{\partial s^i} \vec{\mathbf{n}}_\alpha \vec{\mathbf{n}}_\alpha{}^\top \frac{\partial \mathbf{c_s}(u_\alpha)}{\partial s^k} \dot{s}^i = \sum_{\alpha=1}^{N} \mathbf{d}_\alpha^\top \frac{\partial \mathbf{c_s}(u_\alpha)}{\partial s^k}, \quad k = 1, \ldots, n. \tag{10}$$

In matrix notation, this can be written as

$$\mathbf{M}(\mathbf{s})^\top \, \mathbf{M}(\mathbf{s}) \, \dot{\mathbf{s}} = \mathbf{r}(\mathbf{s}) \quad \text{with} \quad M_{\alpha,k} = \vec{\mathbf{n}}_\alpha{}^\top \frac{\partial \mathbf{c_s}(u_\alpha)}{\partial s_k}. \tag{11}$$

If the matrix $\mathbf{M}^\top \mathbf{M}$ is regular (see [1] for sufficient conditions), then (11) is a differential equation for the shape parameters $\mathbf{s} = \mathbf{s}(t)$. It defines an *evolution process* for the curve. Starting with some initial values, we may apply the evolution in order to drive the curve closer to the given points.

In general, the differential equation (11) cannot be solved exactly. In order to solve it numerically, one may use explicit Euler steps, i.e. $s^i \to s^i + h\dot{s}^i$ with a suitable step size h. As observed in [2], solving this differential equation by Euler steps with some step size h is equivalent to Gauss–Newton iterations with the same step size h for solving the least–squares problem (7). In the case of polynomial spline curves and surfaces, this has been studied as 'normal distance minimization' [4] or, more recently, 'TDM' [17].

The evolution viewpoint allows to control the step size in a geometric way: after computing the time derivatives $\dot{\mathbf{s}}$, one may choose the step size h such that the displacement of each closest point, which is approximately equal to $h\mathbf{v}(u_\alpha)$, does not exceed a certain threshold (e.g., a user–defined feature size). Moreover, it makes it possible to analyze the dependency of the entire process on the choice of the shape parameters, i.e., the chart of the shape manifold M. This will be discussed in the next section.

Remark 2. If the curve is still far from the data points then the computation of the closest points may be too costly. In this situation one may use a precomputed (e.g., using Graphics hardware [7]) unsigned distance field and uniformly distributed sample points on the curve in order to derive the evolution.

§4. Approximation flows

The system (10) can be used to define a contravariant vector field.

Lemma 1. *If the matrix* $\mathbf{M}(\mathbf{s})^\top \mathbf{M}(\mathbf{s})$ *is regular for all* $\mathbf{s} \in \Omega$*, then the solution* $(a^1, \dots a^n)$ *of*

$$\sum_\alpha \frac{\partial \mathbf{c_s}(u_\alpha)^\top}{\partial s^i} \vec{\mathbf{n}}_\alpha \vec{\mathbf{n}}_\alpha{}^\top \frac{\partial \mathbf{c_s}(u_\alpha)}{\partial s^k} a^i = \sum_\alpha \mathbf{d}_\alpha^\top \frac{\partial \mathbf{c_s}(u_\alpha)}{\partial s^k}, \quad k = 1, \dots, n. \quad (12)$$

defines a contravariant vector field on the shape manifold M*.*

Proof: We change the shape parameters from $\mathbf{s}$ to $\boldsymbol{\sigma}$. In the new chart, the coordinates $(a_*^1, \dots a_*^n)$ of the contravariant vector field are obtained as the solution of

$$\sum_\alpha \frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^q}^\top \vec{\mathbf{n}}_\alpha \vec{\mathbf{n}}_\alpha{}^\top \frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^p} a_*^q = \sum_\alpha \mathbf{d}_\alpha^\top \frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^p}, \quad p = 1, \dots, n,$$

which implies

$$\sum_\alpha \left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^q} \right)^\top \vec{\mathbf{n}}_\alpha \vec{\mathbf{n}}_\alpha{}^\top \left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^p} \frac{\partial \sigma^p}{\partial s^k} \right) a_*^q = \sum_\alpha \mathbf{d}_\alpha^\top \left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^p} \frac{\partial \sigma^p}{\partial s^k} \right).$$

Using the identity

$$\left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^q} \right)^\top = \left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^l} \frac{\partial \sigma^l}{\partial s^i} \right)^\top \frac{\partial s^i}{\partial \sigma^q}$$

we obtain

$$\sum_\alpha \left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^l} \frac{\partial \sigma^l}{\partial s^i} \right)^\top \vec{\mathbf{n}}_\alpha \vec{\mathbf{n}}_\alpha{}^\top \left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^p} \frac{\partial \sigma^p}{\partial s_k} \right) \frac{\partial s^i}{\partial \sigma^q} a_*^q = \sum_\alpha \mathbf{d}_\alpha^\top \left(\frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^l} \frac{\partial \sigma^l}{\partial s^k} \right).$$

With the help of the transformation rule (5) and

$$\frac{\partial \mathbf{c_s}(u_\alpha)}{\partial s^i} = \frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial s^i} = \frac{\partial \mathbf{c}^*_{\boldsymbol{\sigma}}(u_\alpha)}{\partial \sigma^l} \frac{\partial \sigma^l}{\partial s^i},$$

this can now be shown to be equivalent to (12). □

Corollary 1. *The result of the evolution process defined by the differential equation* (11) *is independent of the choice of the shape parameters.*

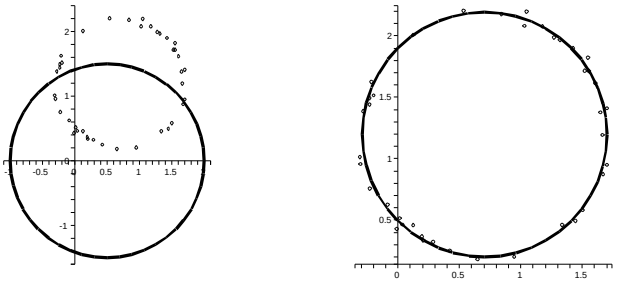

Fig. 2. Example 1: Initial position (left) and after 20 steps (right).

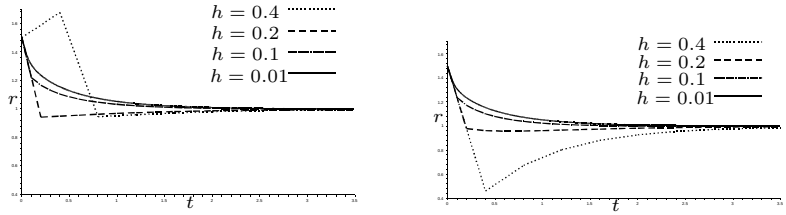

Fig. 3. Evolution of the radius for different step sizes. Left: implicit representation, Eq. (3), right: parametric representation, Eq. (4).

Proof: This follows from the fact that the trajectories of the flow defined by the contravariant vector field a^i are independent of the choice of the chart of the shape manifold. □

Remark 3. Two evolving curves $\mathbf{c}_\mathbf{s}$ and $\mathbf{c}^*_{\boldsymbol{\sigma}}$ with shape parameters $\mathbf{s}(t)$ and $\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}(\mathbf{s}(t))$ are identical and they therefore share the same normal velocities. Consequently, the solutions to the optimization problem (9), which defines the evolution, are identical in both charts.

§5. Examples

We continue the three examples of Section 2. The first example illustrates the result of Corollary 1.

Example 1: We initialize the center of the circle with $(0.5, 0)$ and the radius is set to 1.5. The target of the evolution is a circle–shaped point cloud, consisting of 40 data points with random errors (amplitude 0.05). Figure 2 (left) shows the initial configuration. We applied the evolution to both representations. After 20 Euler steps, both reach an average

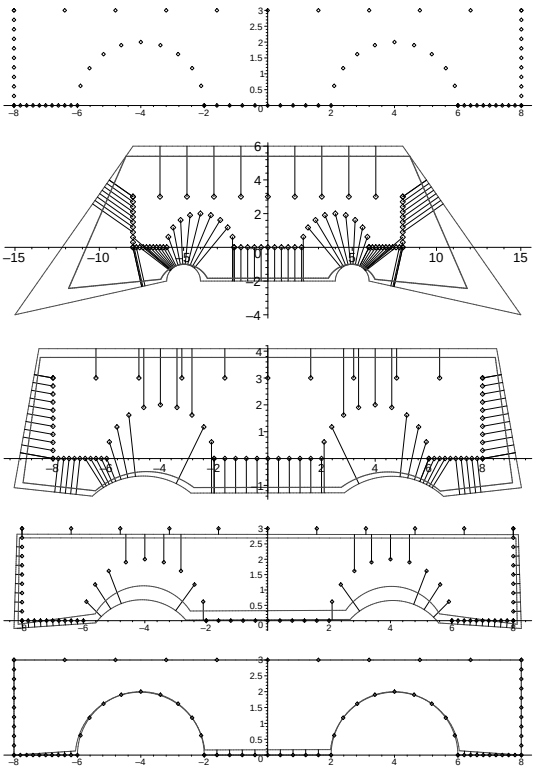

Fig. 4. Example 2: The data (top row) and the position obtained by the evolution after 1, 5, 12 and 22 steps. Note the different scaling!

error of 0.024, see Fig 2 right. Though the implicit and the parametric representations were initialized with the same values and both reached the same solution, the paths leading to it were different, which seems to be a contradiction to Corollary 1. However, the difference is caused by the fact that the Euler steps act in different charts, i.e., in different linearizations of the shape manifold. If the accuracy of the numerical method is increased, then the two evolution paths become more and more similar to each other.

In Fig. 3 we compare the evolutions of the circle radius in the two representations. If a large step size is h chosen, then they are different, but they become increasingly similar for smaller values of h.

Example 2: We approximate the data shown in Fig. 4, where the shape parameters are chosen as described in Section 2. The evolution is shown in Fig. 4. In each of the four figures, we show the position before the iteration step (thin) along with the difference vectors $\mathbf{d}_\alpha$ and the new

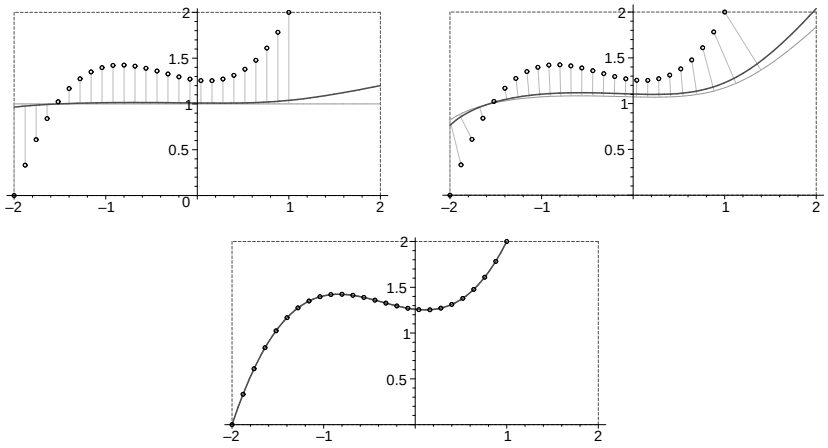

Fig. 5. Example 3: Evolution of a rational cubic curve.

position (thick curve). After 25 iterations, the average error is below 1e-8.

Example 3: Finally we consider again the system of rational cubic curves, but this time within the box $[-2, 2] \times [0, 2]$. In order to fit the data shown in Fig. 5, we apply the evolution process, where the initial value is a horizontal line with a uniform parameterization. In the step 6 of the iteration (top right), we have to change the chart, since the second intersection point with the boundaries leaves the right boundary and continues on the top boundary of the box.

§6. Conclusion

The continuous version of the Gauss–Newton method for fitting a curve or surface from a shape manifold to given point data defines an evolution process which is equivalent to a flow on the shape manifold. Consequently, the result is invariant with respect to the choice of the system of parameters (the chart) controlling the shape of the curve or surface.

Acknowledgment. The authors were supported by the Austrian Science Fund through the Joint Research Programme S92 "Industrial Geometry", subproject 2.

References

1. M. Aigner and B. Jüttler, Hybrid curve fitting, Computing, to appear. Available as FSP report no. 2 at www.ig.jku.at.

2. M. Aigner, Z. Šír and B. Jüttler, Least–squares approximation by Pythagorean hodograph spline curves via an evolution process, in: M.-S. Kim, K. Shimada, eds., *GMP 2006*, Springer LNCS vol. 4077, 45–58.
3. M. Alhanaty and M. Bercovier, Curve and surface fitting and design by optimal control methods, Computer-Aided Design, **33** (2001), 167–182.
4. A. Blake and M. Isard, *Active contours*. Springer, New York, 1998.
5. D. Cremers et al., Diffusion snakes: Introducing statistical shape knowledge into the Mumford–Shah functional. Int. J. Comp. Vision, **50** (2002), 295–313.
6. T. Frankel, *The Geometry of Physics*, Cambridge Univ. Press, 1999.
7. Kenneth E. Hoff et al., Fast computation of generalized Voronoi diagrams using graphics hardware. In *Proc. SIGGRAPH '99*, 277–286, ACM Press, New York, 1999.
8. J. Hoschek and D. Lasser, *Fundamentals of computer aided geometric design*. A K Peters, Wellesley, MA, 1993.
9. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: active contour models. Int. J. Comp. Vision, **1** (1987), 321–331.
10. H. Pottmann and S. Leopoldseder, A concept for parametric surface fitting which avoids the parametrization problem, Comput. Aided Geom. Design, **20** (2003), 343–362.
11. H. Pottmann, S. Leopoldseder, and M. Hofer, Approximation with active B-spline curves and surfaces. In Pacific Graphics 2002, 8–25. IEEE.
12. H. Pottmann et al., Industrial geometry: recent advances and applications in CAD. Computer-Aided Design, **37** (2005), 751–766.
13. D. Rogers and N. Fog. Constrained B-spline curve and surface fitting. Computer-Aided Design, **21** (1989), 641–648.
14. M. Rousson and N. Paragios, Shape Priors for Level Set Representations. ECCV (2), (2002), 78–92.
15. E. Saux and M. Daniel, An improved Hoschek intrinsic parametrization, Comput. Aided Geom. Design, **20** (2003), 513–521.
16. T. Speer, M. Kuppe, J. Hoschek. Global reparametrization for curve approximation. Comput. Aided Geom. Design, **15** (1998), 869–877.
17. W. Wang, H. Pottmann, and Y. Liu, Fitting B-spline curves to point clouds by squared distance minimization. ACM Trans. on Graphics, **25.2** (2006).

Martin Aigner and Bert Jüttler
Institute of Applied Geometry, Johannes Kepler University
Altenberger Sr. 69, 4040 Linz, AUSTRIA
{martin.aigner,bert.juettler}@jku.at
http://www.ag.jku.at