

# Constrained Curve Fitting on Manifolds

Simon Flöry and Michael Hofer \*

*Geometric Modeling and Industrial Geometry Research Group, Vienna University of Technology, Wiedner Hauptstraße 8-10, A-1040 Wien, Austria*

---

## Abstract

When designing curves on surfaces the need arises to approximate a given noisy target shape by a smooth fitting shape. We discuss the problem of fitting a B-spline curve to a point cloud by squared distance minimization in the case that both, the point cloud and the fitting curve, are constrained to lie on a smooth manifold. The on-manifold constraint is included by using the first fundamental form of the surface for squared distance computations between the point cloud and the fitting curve. For the solution we employ a constrained optimization algorithm that allows us to include further constraints such as one-sided fitting or surface regions that have to be avoided by the fitting curve. We illustrate the effectiveness of our algorithm at hand of several examples showing different applications.

*Key words:* B-spline curve, curve fitting, constrained optimization, squared distance minimization, geometric constraints, damped Gauss-Newton method, shape approximation, free-form curves, splines on manifolds, constrained curve design.

---

## 1. Introduction

The approximation of a given set of scattered data points (the *target shape*) by a smooth polynomial curve (the *fitting shape*) is a frequently encountered problem in CAD/CAM, computer graphics, computer vision, image processing and many other fields. Due to their popularity and widespread use often B-spline curves are chosen as the fitting shapes. The target shapes we are working with are noisy *unordered* point sets, also called *point clouds* henceforth. Such point clouds arise in various ways, e.g. they result from digitization, they appear in intermediate steps of surface reconstruction in reverse engineering, or they are a design input.

Most of the previous work concerns curve fitting in Euclidean spaces. However, practical applications often impose additional constraints on the curve fit-

ting problem. These constraints can be of various nature and include spatial restrictions that constrain the curve to lie on a certain surface, within some tolerance zone, or outside of forbidden regions that must not be entered by the fitting curve. We introduce the novel aspect of approximating *unordered* point clouds that are lying on smooth parametric surfaces with constraints that prevent the fitting shapes from entering forbidden areas (e.g. trimmed areas of B-spline surfaces) or that allow one-sided fitting of the target shapes. Furthermore, for design purposes it is advantageous to first sketch a curve on a surface which is then automatically smoothed, possibly with the additional constraint that the fitting curve shall follow a certain tangential vector field of the surface. Note that the type of geometric constraints we employ is novel and different from previously studied constraints for curve fitting such as a fitting circle whose radius is constrained to a certain value (cf. Benkő et al. (2002)).

From the optimization point of view the curve approximation problem is often dealt with as a non-

---

\* Corresponding author.

*Email addresses:* floery@geometrie.tuwien.ac.at (Simon Flöry), hofer@geometrie.tuwien.ac.at (Michael Hofer).

linear least squares problem. Since the distance from the target shape is measured orthogonally to the fitting shape, some researchers speak of orthogonal distance regression (ODR). The present paper uses the optimization framework discussed in detail in Wang et al. (2006) and extends the constrained curve fitting problem studied by Flöry (2006) from the Euclidean space to manifolds.

The main contributions of our paper are:

- An optimization algorithm for fitting B-spline curves to unorganized point clouds lying on a smooth manifold.
- The incorporation of additional side conditions into the optimization problem such as obstacles (e.g. trimmed regions) that have to be avoided, and tangential vector fields that guide the fitting curve.

The remaining parts of this paper are organized as follows. First, we review some recent literature and continue our introduction into the topic. Then, we outline an algorithm to solve the curve fitting problem on manifolds and discuss its steps, including foot point computation, description of current fitting error and optimization, in detail. Additionally, we consider three more constraints: two kinds of obstacles (the point cloud itself and forbidden regions) and a design constraint (tangential vector field). Finally, we illustrate applications of our algorithm in several examples.

### 1.1. Related Work

The amount of literature on parametric curve fitting is huge and thus we only cite closely related work. In Wang et al. (2006), the authors undertake a detailed review of current curve fitting methods from the point of optimization and we adopt their terminology in the following. Chronologically, the first method and its variants studied e.g. by Plass and Stone (1983); Hoschek (1988); Bercovier and Jacobi (1994); Goshtasby (2000); Lee (2000); Saux and Daniel (2003) employs *Point Distance Minimization (PDM)* and usually converges very slowly. The chronologically next method employs error terms that measure the orthogonal distance to the curve tangent. It was proposed by Blake and Isard (1998) and coined *Tangent Distance Minimization (TDM)* by Wang et al. (2006). This method is a Gauss-Newton iteration (see also Atieg and Watson (2003); Liu et al. (2005)) and achieves good results if we add an appropriate regularization technique.

A third method (see Wang et al. (2006); Liu et al. (2006); Pottmann et al. (2005); Yang et al. (2004)) employs *Squared Distance Minimization (SDM)* and is based on a quadratic approximation of the squared distance function (Ambrosio and Montegazza (1998), Pottmann and Hofer (2003)) to the fitting shape. The superior performance (by means of convergence speed and stability) of SDM is due to the fact that also second order information of the fitting curve (curvature) is included. However, this error term requires curvature computations which are sometimes too expensive to be carried out. Thus in the present paper we employ a modified regularized TDM method and we achieve very satisfactory results.

The curve (and surface) fitting problem is also known in the literature as orthogonal distance regression (Ahn (2004); Atieg and Watson (2003)). An alternative optimization approach for parametric curve (and surface fitting) using a trust region algorithm was studied by Helfrich and Zwick (1996).

Computing curves with spatial constraints is an important active research area in Computer-Aided Design. One class of algorithms deals with spatial constraints that have to be met by the designed curves. Hildebrandt et al. (2005) proposed an algorithm for smoothing 3d curves with an  $\varepsilon$ -constraint that keeps the curve inside a pipe surface of radius  $\varepsilon$  around the original curve. Recently, Peters (in Peters and Wu (2004)) introduced SLEVEs for planar spline curves to solve the so-called *channel problem* (Myles and Peters (2005)): the task here is to compute a spline curve with a limited amount of pieces that traverses a narrow channel bounded by polygonal obstacles. An older contribution by Opfer and Oberle (1988) dealt with cubic splines and obstacles and the constraint interpolation with rational cubics was studied in Meek et al. (2003). The paper of Renner and Weiß (2004) includes an algorithm for fitting a B-spline curve to an ordered sequence of sample points lying on a B-spline surface. From the optimization point of view they are using a direction-weighted least-squares fitting that employs PDM without parameter correction.

Recently Machado (2006) and Machado and Leite (2006) studied the fitting of an *ordered* sequence of points by smooth paths on Riemannian manifolds. They present solutions for fitting geodesics on the unit sphere  $S^n$  and on the rotation group  $SO(n)$ , and for fitting smoothing splines to ordered point clouds on general Riemannian manifolds. Their work follows the pioneering work of Noakes et al. (1989)

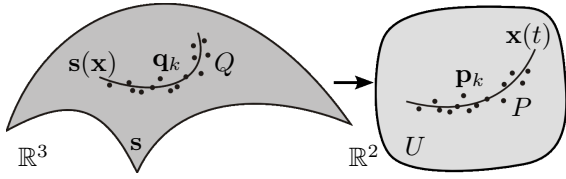


Fig. 1. Curve fitting on a parametric surface  $\mathbf{s}$  is considered in the parameter space  $U \subseteq \mathbb{R}^2$  while taking the inner geometry of  $\mathbf{s}$  into account at the same time.

where cubic splines on Riemannian manifolds are seen as solutions of a variational problem. Related is also Bohl's PhD thesis (Bohl (1999)) that computes energy-minimizing splines on trimmed two-dimensional parametric surfaces. The algorithm presented by Hofer and Pottmann (2004) was also used for smoothing noisy input curves on manifolds by means of energy-minimizing spline curves on manifolds that interpolate a certain subset of the input points. Thus the present paper also improves this previous work in the sense that now all data points (and not only some arbitrarily picked ones) are used to find a smooth solution curve that faithfully approximates the noisy input data.

## 2. Curve Fitting on Manifolds

Let  $\Phi$  be a two dimensional manifold in  $\mathbb{R}^3$ , e.g. a parameterized surface  $\mathbf{s} : U \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}^3$  and  $Q = \{\mathbf{q}_k : k = 0, \dots, n\} \subseteq \Phi$  a point cloud thereon. Our aim is to approximate  $Q$  with a parametric curve  $\mathbf{s}(\mathbf{x})$  on  $\Phi$ . Considering that curve fitting in  $\mathbb{R}^2$  is a well surveyed topic, we choose to carry out the curve fitting in the parameter space  $U$  of  $\mathbf{s}$  while taking the inner geometry of the manifold into account at the same time. Accordingly, we may state our problem as follows: given points  $P = \{\mathbf{p}_k : k = 0, \dots, n\} \subseteq U$  such that  $\mathbf{q}_k = \mathbf{s}(\mathbf{p}_k)$  we want to find a curve  $\mathbf{x}$  approximating the point cloud  $P$  in a least squares sense (see Fig. 1). A general formula for this type of problem is of the form

$$h = \sum_{k=0}^n \tilde{d}^2(\mathbf{x}(t), \mathbf{p}_k) + wr, \quad (1)$$

where  $\tilde{d}^2$  denotes an approximation of the squared distance function and  $r$  a regularization term weighted by  $w$ , all detailed below. If the data points  $\mathbf{q}_k$  are given directly on the manifold, then we first compute the corresponding points  $\mathbf{p}_k$  in the parameter space, cf. Hu and Wallner (2005).

In the following, we describe a general curve fitting algorithm for point clouds in  $\mathbb{R}^2$  and modify it

in certain steps to acknowledge that we perform a curve fitting on manifolds. The input to the algorithm are the point cloud  $P$  and the initial position of the fitting curve  $\mathbf{x}$ .

**Algorithm 1** A general curve fitting algorithm involves the following steps:

- (i) Compute the foot points  $\mathbf{f}_k$  of  $P$  on  $\mathbf{x}$ .
- (ii) Describe the current fitting error in these foot points.
- (iii) Get an updated position  $\mathbf{x}_c$  of  $\mathbf{x}$  by minimizing the fitting error.

Steps (i) to (iii) are repeated until the fitting is of satisfactory quality and a least squares approximation of the point cloud is returned. Algorithm 1 turns the fitting problem into an iterative optimization problem that we stop once a predefined criterion is met (for example the fitting error falls below a pre-defined threshold or a maximum number of iteration steps is reached).

If we have a closer look at Algorithm 1, the problem of fitting a curve to a point cloud in the parameter space of a two dimensional manifold requires modifications to two steps: the foot point computation in step (i) and the description of the current fitting error term in step (ii).

### 2.1. Foot Point Computation

In  $\mathbb{R}^2$ , the foot point of the shortest distance from a point  $\mathbf{p}_k$  to a parameterized curve  $\mathbf{x}(t)$  is given by minimizing the squared distance from the curve to  $\mathbf{p}_k$ ,

$$g(t) = \|\mathbf{x}(t) - \mathbf{p}_k\|^2, \quad (2)$$

whereas  $\|\mathbf{v}\| = \sqrt{\mathbf{v}^T \cdot \mathbf{v}}$  denotes the common Euclidean norm. Geometrically, the minimum of Equ. (2) is taken for that parameter value  $t_k$  for which the curve tangent in  $\mathbf{x}(t_k)$  is orthogonal to the connecting line from  $\mathbf{x}(t_k)$  to  $\mathbf{p}_k$ . On a parameterized surface  $\mathbf{s}$  the shortest distance between a curve  $\mathbf{s}(\mathbf{x})$  and a point  $\mathbf{q}_k \in \mathbf{s}$  is measured along a geodesic. In general, geodesics are not just straight lines and can not be computed explicitly. To simplify our computations, we work with a local first order approximation of the surface, namely with the tangent space. The first fundamental form  $G_k$  defines an inner product on the tangent space  $T\mathbf{q}_k$  in  $\mathbf{q}_k$  (see do Carmo (1976)). In  $T\mathbf{q}_k$ ,  $\|\mathbf{v}\|_{G_k} = \sqrt{\mathbf{v}^T \cdot G_k \cdot \mathbf{v}}$  gives a norm with respect to the inner geometry of the manifold. Thus, we compute to each point  $\mathbf{q}_k$  the foot point  $\mathbf{f}_k$  on  $\mathbf{s}(\mathbf{x}(t))$  in the tangent space  $T\mathbf{q}_k$  (see Fig. 2) by minimizing

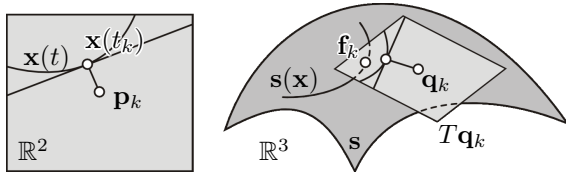


Fig. 2. Foot point computation in  $\mathbb{R}^2$  and on a parameterized surface  $\mathbf{s}$ . On the surface  $\mathbf{s}$ , the computation is performed in the tangent space  $T\mathbf{q}_k$  with the inner product induced by the first fundamental form  $G_k$  in  $\mathbf{q}_k$ .

$$g(t) = \|\mathbf{x}(t) - \mathbf{p}_k\|_{G_k}^2, \quad (3)$$

for example with a Newton iteration. This first order approximation is sufficient for our purposes. In the course of the iteration, the fitting shape approximates the data points better and better and thus the foot point computation becomes more accurate.

## 2.2. Fitting Error Term

There are various ways to describe the current fitting error in a foot point  $\mathbf{f}_k = \mathbf{x}(t_k)$ . Most approaches are based on an approximation of the squared distance from  $\mathbf{f}_k$  to  $\mathbf{p}_k$ . As stated in the introduction we base our succeeding considerations on a regularized *Tangent Distance Minimization* method.

Let  $d$  be the signed distance function to a curve  $\mathbf{x}$ , defined as the solution of the Eikonal equation  $\|\nabla d\| = 1$  with  $d(\mathbf{x}) = 0$ . Second order Taylor approximation of the squared distance function  $d^2$  in a point  $\mathbf{f}_k = \mathbf{x}(t_k)$  of  $\mathbf{x}$  gives

$$d^2(\mathbf{u}) \approx [(\mathbf{u} - \mathbf{f}_k)^T \cdot \nabla d(\mathbf{f}_k)]^2. \quad (4)$$

In Euclidean space, the gradient of  $d$  in  $\mathbf{f}_k$  can be identified with the unit normal vector  $\mathbf{n}_k$  in  $\mathbf{f}_k$ . On manifolds, the relation  $\|\nabla d\| = 1$  has a different meaning. If we recall that we use the norm induced by the first fundamental form  $G_k$  to measure lengths in the tangent space and that the gradient of a function on a manifold is given by  $\nabla_{G_k} d = G_k^{-1} \cdot \nabla d$ , the requirement  $\|\nabla d\| = 1$  turns into

$$\|\nabla_{G_k} d\|_{G_k}^2 = \nabla d^T \cdot G_k^{-1} \cdot \nabla d = 1. \quad (5)$$

Thus,  $\nabla d(\mathbf{f}_k)$  of Equ. (4) will be a vector pointing in the direction of  $\mathbf{n}_k$  again, however, we norm it according to Equ. (5).

This approximation of the squared distance function is used to measure the current fitting error  $\varepsilon_k$ . For a curve fitting on manifolds,

$$\varepsilon_k = [(\mathbf{p}_k - \mathbf{x}(t_k))^T \cdot \mathbf{n}_k]^2 \quad (6)$$

with  $\mathbf{n}_k^T \cdot G_k^{-1} \cdot \mathbf{n}_k = 1$  describes the current fitting error term in a foot point  $\mathbf{f}_k = \mathbf{x}(t_k)$ .

## 2.3. Optimization

We choose B-spline curves as approximating entities and describe the updated position  $\mathbf{x}_c(t) = \sum_{i=0}^m N_i(t)(\mathbf{d}_i + \mathbf{c}_i)$  by displacements  $\mathbf{c}_i$  of the control points  $\mathbf{d}_i$ . Subscript  $\mathbf{c}$  in  $\mathbf{x}_c(t)$  reflects the vector of unknown displacements  $\mathbf{c} = (\mathbf{c}_0, \dots, \mathbf{c}_m)$ . The objective function of the curve fitting problem comprises several terms. First, we include the current fitting error, given by summing up all error terms  $\varepsilon_k$  of Equ. (6)

$$h(\mathbf{c}) = \sum_{k=0}^n [(\mathbf{p}_k - \mathbf{x}_c(t_k))^T \cdot \mathbf{n}_k]^2, \quad (7)$$

with  $\mathbf{n}_k^T \cdot G_k^{-1} \cdot \mathbf{n}_k = 1$  for  $k = 0, \dots, n$ . In order to ensure that the new position  $\mathbf{x}_c$  is not only optimal in a mathematical sense but also visually appealing, we further add two regularization terms:

$$r_0(\mathbf{c}) = \int \|\dot{\mathbf{x}}_c(t)\|_{G(\mathbf{x}(t))}^2 dt, \quad (8)$$

where  $G(\mathbf{x}(t))$  denotes the first fundamental form in  $\mathbf{x}(t)$ , and

$$r_1(\mathbf{c}) = \int \|\ddot{\mathbf{x}}_c(t)\|^2 dt. \quad (9)$$

Thus we get the objective function

$$\sum_{k=0}^n [(\mathbf{p}_k - \mathbf{x}_c(t_k))^T \cdot \mathbf{n}_k]^2 + w_0 r_0(\mathbf{c}) + w_1 r_1(\mathbf{c}). \quad (10)$$

The smoothing terms  $r_0$  and  $r_1$  are chosen such that the shape of the fitting curve is as simple as possible. By neglecting any regularization, the optimization would fail to converge or give unpleasant results, for example strong oscillations.  $r_0$  keeps the length of  $\mathbf{x}_c$  small: as the fitting is constrained to happen on a manifold, we approximate the length of the fitting curve by integrating the squared norm (as induced by the first fundamental form) of the curve's first derivative over the entire curve.

The second term is included to smooth the fitting and we apply a simplified measure for the curve's bending energy. Both  $r_0$  and  $r_1$  enter the objective function weighted with a factor. In general, we initialize these weights such that the current approximation error and the regularization terms influence the optimization equally (e.g. by setting  $w_0 = w_1 = \sum_k \varepsilon_k / (r_0 + r_1)$ ) and halve the factors at each iteration step. We are going to give more details on how we choose  $w_0$  and  $w_1$  in our applications in Sec. 4.

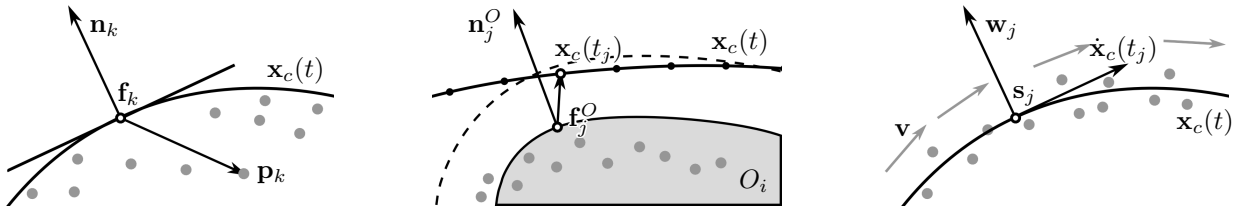


Fig. 3. We further constrain curve fitting on manifolds by regarding the point cloud  $\{\mathbf{p}_k\}$  itself as an obstacle (left), by avoiding forbidden regions  $O_i$  (middle) and by requiring  $\mathbf{x}_c$  to be tangential to a vector field  $\mathbf{v}$  (right). For better visualization, all three figures illustrate the setting after updating the curve’s position but before new foot points have been computed.

The objective function in Equ. (10) is a positive semi-definite quadratic form (with the displacements in  $\mathbf{c}$  as unknowns) and the minimum is found by solving a system of linear equations. TDM in general and our modified version as well face instability issues if applied without further adjustment. As shown in Wang et al. (2006) TDM is a variant of a Gauss-Newton method. It is well known that the convergence behavior of Gauss-Newton methods can be improved by regularization strategies such as the *damped Gauss-Newton* or the *Levenberg-Marquardt* method (see e.g. Kelley (1999)).

We employ a damped Gauss-Newton method and control the step-width at each iteration. Let  $h(\mathbf{0})$  be the current approximation error and  $h(\mathbf{c}^0)$  the fitting error for the displacement  $\mathbf{c}^0$  obtained by minimizing  $h(\mathbf{c})$ . As long as  $h(\mathbf{c}^i) > h(\mathbf{0})$  holds and  $i$  is below a certain upper bound (usually 10), the position update is decreased by the rule  $\mathbf{c}^{i+1} = 0.5\mathbf{c}^i$ .

### 3. Additional Constraints

The problem of fitting a curve to a point cloud is an optimization problem, as shown in the previous section. So far, the solution space was not restricted by any side conditions and the unknown displacements  $\mathbf{c}$  were allowed to be elements of whole  $\mathbb{R}^{2m+2}$ . In the following, we examine in how far a curve fitting on manifolds may benefit from adding constraints to the optimization process.

Our main interest is to fit a curve to a point cloud on a manifold and to avoid *obstacles* at the same time. In Flöry (2006), two types of constraints are discussed for curve and surface fitting in Euclidean space. Here, we want to briefly review this work and show how it can be extended to curve fitting on manifolds. Then, constraints more specific to curve fitting on manifolds are discussed when we describe approximations tangential to some vector field.

#### 3.1. Point Cloud as Obstacle

At first, we want to regard the elements of the point cloud  $P = \{\mathbf{p}_k : k = 0, \dots, n\}$  itself as obstacle. In this way we will be able to reconstruct any apparent boundaries of  $P$ . More precisely speaking, if we define some orientation by orienting the normals  $\mathbf{n}_k$  of the approximating curve in a unique way, the data points will be located on a single side of the final fitting shape.

Let the normals  $\mathbf{n}_k$  point outside of the approximating curve. As before,  $\mathbf{x}_c(t_k)$  denotes the foot point of a point cloud element  $\mathbf{p}_k$  on  $\mathbf{x}_c$ . Then, the linear constraints

$$(\mathbf{p}_k - \mathbf{x}_c(t_k))^T \cdot \mathbf{n}_k \leq 0 \quad k = 0, \dots, n \quad (11)$$

make the fitting curve approximate the outer boundary of the point cloud and  $P$  will lie inside (see Fig. 3, left). The converse scenario — a reconstruction of the inner boundary with  $P$  outside of  $\mathbf{x}_c$  — can be achieved by either changing the orientation of the normals or by replacing the less or equal sign in Equ. (11) with a greater or equal sign.

#### 3.2. General Obstacles

A different kind of obstacles are arbitrary regions the final fitting must avoid. In curve fitting on manifolds, such obstacles are of special interest. As the approximation is done in the parameter space  $U$  of the manifold, we can perform a curve fitting on a trimmed surface by adding those subsets of  $U$ , that define surface parts trimmed away, as obstacles.

Let  $O_i \subseteq U$  be such a subset in the parameter space. For a sample point  $\mathbf{x}_c(t_j)$  on the fitting curve,  $\mathbf{f}_j^O$  denotes the foot point of the shortest distance from  $\mathbf{x}_c(t_j)$  to the boundary of the obstacle  $O_i$  and  $\mathbf{n}_j^O$  the outward oriented normal vector in  $\mathbf{f}_j^O$ . Then, by adapting the idea of Sec. 3.1,

$$(\mathbf{f}_j^O - \mathbf{x}_c(t_j))^T \cdot \mathbf{n}_j^O \leq 0 \quad \forall j \in J, \quad (12)$$

constrains the fitting to avoid the obstacle  $O_i$  (see Fig. 3, middle). This side condition isn't active for all sample points on  $\mathbf{x}_c$  but only for those either inside an obstacle or within a certain distance to an obstacle's boundary (see the dashed line in Fig. 3, middle). The latter criteria ensures that the fitting curve doesn't penetrate the obstacle again to fit the data points once it left the forbidden region.

### 3.3. Tangential Vector Fields

Let  $\mathbf{v}(\mathbf{q}) : \mathbf{s} \rightarrow \mathbb{R}^3$  be a tangential vector field defined for each point  $\mathbf{q}$  of a parameterized surface  $\mathbf{s}$ . We would like  $\mathbf{v}$  to control the shape of the approximating curve in certain points. Therefore, we choose samples  $\mathbf{s}_j = \mathbf{s}(\mathbf{x}_c(t_j))$ ,  $j \in J$ , on the surface curve  $\mathbf{s}(\mathbf{x}_c)$  and require the curve in  $\mathbf{s}_j$  to be tangent to  $\mathbf{v}$ . To simplify the formulation of this constraint we introduce an auxiliary vector field  $\mathbf{w}$  that results from  $\mathbf{v}$  by a rotation of 90 degrees in the respective tangent planes. Then, we require

$$\mathbf{w}(\mathbf{s}_j)^T \cdot \frac{d}{dt} \mathbf{s}(\mathbf{x}_c(t))|_{t=t_j} = 0 \quad \forall j \in J.$$

In order to find an equivalent constraint in parameter space, we use the coordinates  $\mathbf{w}_j$  of  $\mathbf{w}(\mathbf{s}_j)$  with respect to the basis of the tangent plane  $T\mathbf{s}_j$  spanned by the partial derivatives of  $\mathbf{s}$  along the parameter directions. Then,

$$\mathbf{w}_j^T \cdot \dot{\mathbf{x}}_c(t_j) = 0 \quad \forall j \in J, \quad (13)$$

makes the fitting curve perpendicular to  $\mathbf{w}$  (and thus tangential to  $\mathbf{v}$ ) in the sample points  $\mathbf{s}_j$  (see Fig. 3, right). If the regularization term  $r_0$  is applied, the first derivative of  $\mathbf{x}_c$  is available anyway and can be used here as well. However, if derivation is too time consuming, we may compute some numerical derivative instead, e.g.  $\mathbf{w}_j^T \cdot (\mathbf{x}_c(t_j + \delta) - \mathbf{x}_c(t_j - \delta)) = 0$  for small  $\delta$ . If  $\mathbf{v}$  is not a tangential but a more general vector field on  $\mathbf{s}$ , we first project  $\mathbf{v}$  onto the tangent space.

For all constraints described here, it turns out to be advantageous to first ignore the side conditions at early iterations of the algorithm (usually 5 to 10 iterations are sufficient for our examples) to increase the stability and global convergence of the algorithm.

### 3.4. Constrained Optimization

After describing several possible constraints for a curve fitting on manifolds, we need to address the question how these side conditions are added to the

unconstrained minimization problem of Equ. (10) and how to solve the emerging constrained optimization problem.

The point cloud constraint, the general obstacle constraint and the vector field constraint have one important characteristic in common. They are all linear in the unknown displacement vector  $\mathbf{c}$ . If we recall the definition of the fitting problem's objective function in Equ. (10) and that it's quadratic in  $\mathbf{c}$ , we get in total a quadratic minimization problem with linear constraints by writing

$$\text{minimize} \quad \sum_{k=0}^n \varepsilon_k(\mathbf{c}) + w_0 r_0(\mathbf{c}) + w_1 r_1(\mathbf{c}) \quad (14)$$

subject to one or more of (11), (12), (13).

There are basically two families of algorithms to solve linear constrained quadratic optimization problems (see e.g. Nocedal and Wright (1999)). As our problem is rather small scaled ( $\mathbf{c} \in \mathbb{R}^{2m+2}$ ) we favor an *Active Set Method* over an *Interior Point Method*. Active set methods deal with the constraints by estimating and continuously updating a set of active side conditions in order to not leave the feasible solution space.

In Sec. 2.3 we already emphasized the need for a step-width control which is still important for the constrained optimization problem. However, the damping method described above may fail as it does not allow the value of the objective function to grow from one iteration step to another. Exactly this might be necessary to fulfill the given constraints and to get a feasible solution at all. By allowing the objective function to increase by a factor of 1.1 (thus requiring  $h(\mathbf{c}_i) > 1.1h(0)$ ) we achieve a good trade off between computing a valid solution and enjoying stable convergence.

## 4. Experiments and Results

We want to illustrate our theoretical considerations with a handful of examples. For the following results, we choose parametric surfaces (cubic tensor product B-spline surfaces) as manifolds. The approximating curves themselves are either closed or open cubic B-spline curves. The point clouds have been created in different ways. For most examples, an auxiliary B-spline curve was evaluated at uniformly distributed parameters and distorted in normal direction by some Gaussian noise. In one example, the points were placed manually in parameter space while in another, a black and white im-

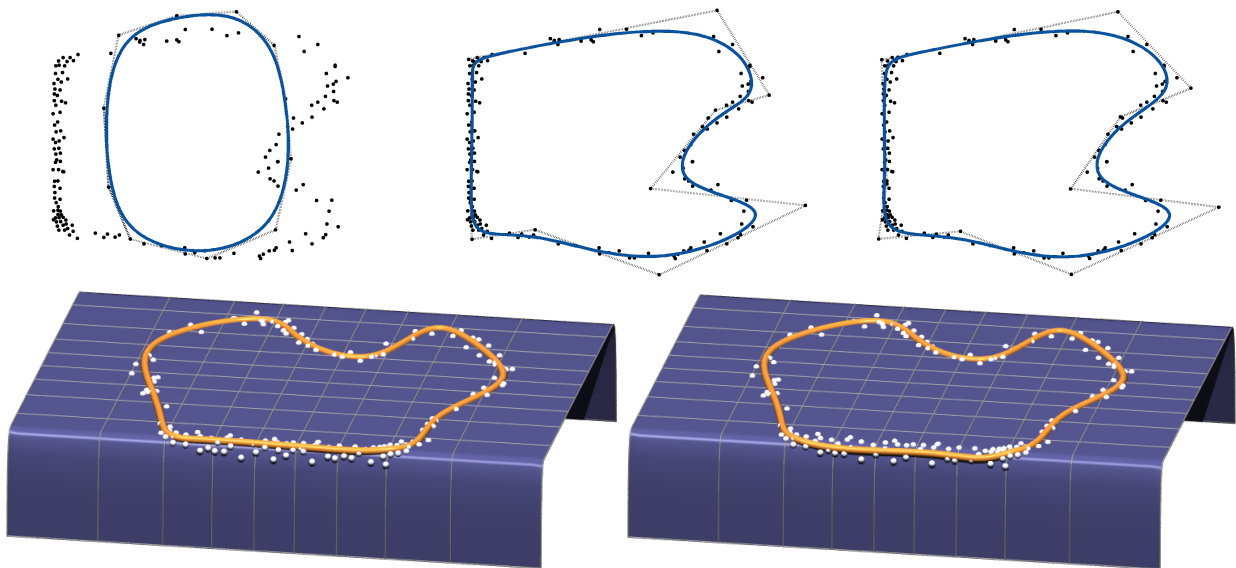


Fig. 4. Example 1. We compare the TDM method based on the Euclidean metric (top center, bottom left) to our modified TDM algorithm taking the surface metric into account (top and bottom, right) on a simple surface. The initial position for both approximations is shown (top left). The fitting curves are shown in parameter space (top row) and on the surface (bottom row). We see that our method (bottom right) approximates the points along the surface bend much better.

age was randomly sampled to get the set of data points. For points interactively entered on a two-dimensional surface we first solve the inversion problem, cf. Hu and Wallner (2005), to map them to parameter space. The algorithm was implemented in an Matlab environment and ran on a *AMD Sempron Processor 3100+*. Runtime information and application specific values for all examples are summarized in Table 1 at the end of this section.

*Example 1.* In a first simple example (see Fig. 4) the point cloud is defined on an u-bend surface. The parameter distortion on the surface is for the vertical parts much higher than for the horizontal one (as indicated by the parameter lines). We performed two fittings, one with the standard TDM method (using the Euclidean metric) and one with our modified TDM algorithm incorporating a surface specific metric. As expected, the fittings are very similar in the horizontal part of the surface, where the control points' positions are nearly the same. However, the error term based on the first fundamental form refines the fitting along the edge and leads to a more natural fitting on the surface. While the standard TDM method approximates the center of the points along the edge, our algorithm shifts the curve left to balance the parameter distortion.

The average fitting errors

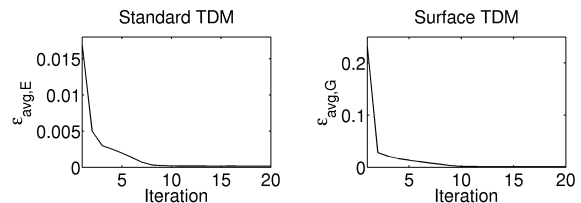


Fig. 5. The average fitting error decreases for our surface TDM algorithm as fast as for the ordinary, standard TDM method. Here shown for Example 1.

$$\varepsilon_{avg,E} := \frac{1}{n+1} \sum_{k=0}^n \|\mathbf{x}(t_k) - \mathbf{p}_k\| \quad (15)$$

and

$$\varepsilon_{avg,G} := \frac{1}{n+1} \sum_{k=0}^n \|\mathbf{x}(t_k) - \mathbf{p}_k\|_{G_k} \quad (16)$$

decrease similarly for both methods and reach a minimum after about 10 iterations (see Fig. 5). Please note that we can not directly compare the final values of  $\varepsilon_{avg,E}$  and  $\varepsilon_{avg,G}$  as they are based on different metrics.

This and some of the following examples feature closed approximating curves. It is well known that a closed smooth cubic B-spline curve is obtained by repeating the first three points of the control polygon. However, this requires us to add the additional constraints



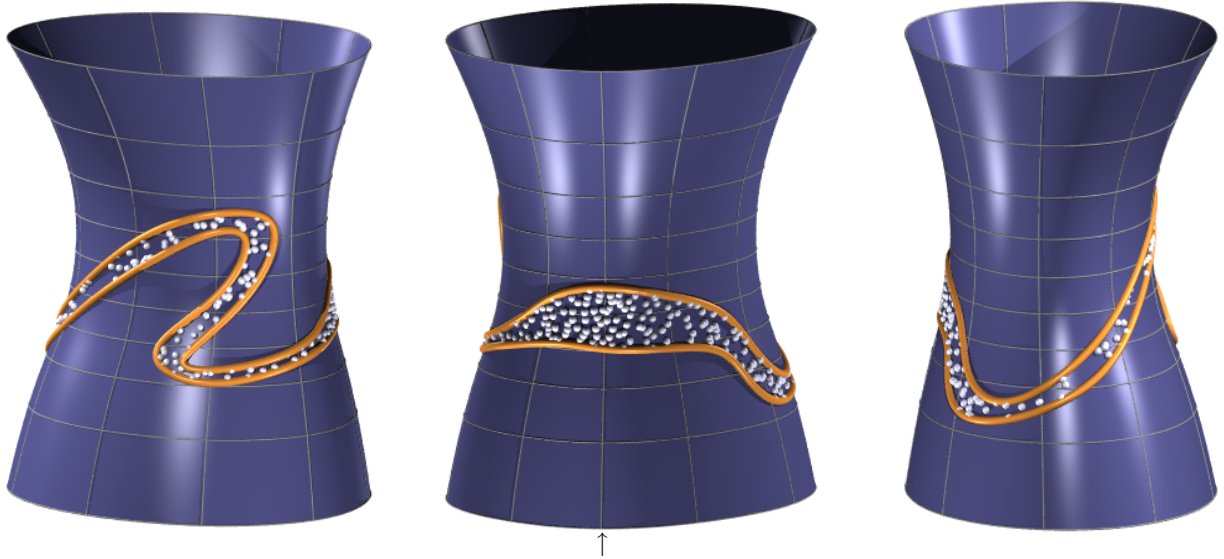


Fig. 6. Example 2. Three views, each rotated by  $120^\circ$ , of a reconstruction of a point cloud's boundaries on a parametric surface closed in one parameter direction. The arrow in the middle marks the parameter line along which the parameter space was glued together.

$$\mathbf{c}_i = \mathbf{c}_{m-2+i} \quad i = 0, 1, 2 \quad (17)$$

to Equ. (14) to maintain a smooth, closed fitting curve.

*Example 2.* In the second example, the point cloud itself is seen as an obstacle. Thus we add the linear constraints of Equ. (11) to the objective function of Equ. (10). In the following, superscript indices denote coordinate entries, e.g.  $u^0$  and  $u^1$  are the first and second parameter directions, respectively. The two parametric B-spline surface of this example is closed in one parameter direction, w.l.o.g. we assume that the surface is closed with respect to parameter  $u^0$  which ranges from 0 to 1. In order to get a smooth closed curve on this kind of surface, the control points of the approximating B-spline curve need to be repeated cyclically

$$d_{m-2+i}^0 = d_i^0 + 1 \quad i = 0, 1, 2 \quad (18)$$

whereas  $d_1^0 = 0$  and  $d_1^0 - d_0^0 = d_2^0 - d_1^0$  holds. Additional constraints

$$c_1^0 = 0 \quad \text{and} \quad c_2^0 + c_0^0 = 0 \quad (19)$$

in Equ. (10) ensure the smoothness throughout the iterations.

Fig. 6 shows three views (each rotated by  $120^\circ$ ) of the composed final results of two approximations of the point cloud's boundaries. The higher curvature parts in the left view, the varying thickness of the point set in the middle view, and the increasing

spacing between the data points in the right view give evidence of the stability and robustness of the algorithm.

The most centered vertical parameter line (see arrow) in the middle plot marks the line along which the parameter space was glued together. The approximating curves pass this line smoothly.

*Example 3.* The next example shows results for the second family of obstacles discussed in Sec. 3. We examine general obstacles in the context of trimmed surfaces. The holes of the surface are defined by regions  $O_i$  in the parameter domain with point sampled boundaries. From these boundaries we compute an approximated distance field (see Tsai (2002)) and propagate foot point and normal information over a grid at the same time. The trimmed regions  $O_i$  are then included in the optimization procedure via the linear constraints described in Sec. 3.2. The final result of a curve fitting on a trimmed surface is visualized in Fig. 7 (top), where the B-spline curve was sampled in 200 points for building the constraints. Please note, that the boundaries of the holes might not be approximated properly by a cubic polynomial curve and thus the curve will not completely follow the said boundary.

As mentioned above, the constraints discussed in Sec. 3 can be combined (see Fig. 7, bottom). In this example we combine the objective function (10) with the linear constraints (11) and (12). We computed



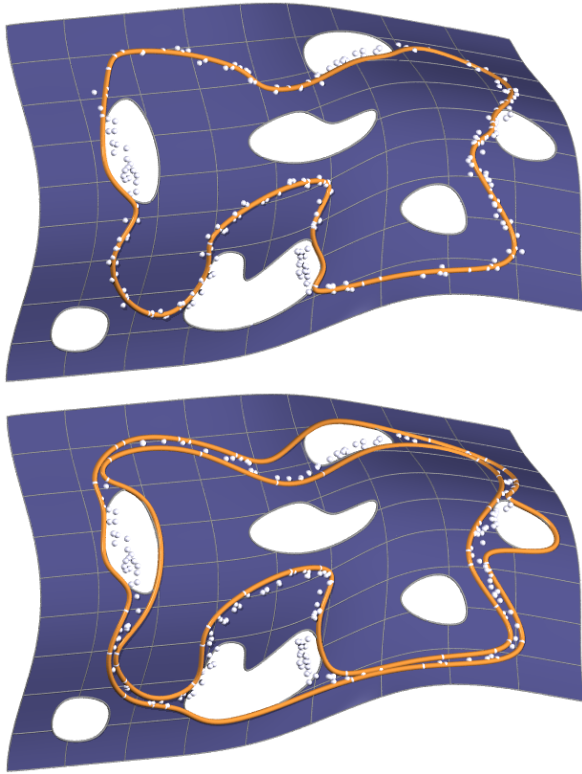


Fig. 7. Example 3. Fitting a curve to a point cloud on a trimmed surface (top). Note that the trimmed regions are avoided by the fitting curve. By combining the constraints of Equ. (11) and (12) the fitting curve approximates the boundaries of the point set and lies outside of the trimmed regions (bottom).

two such curves that enclose the given data points from either side and additionally avoid the trimmed regions. This example also illustrates a limitation of our method: we need to choose two different initial positions to get the two one-sided fits shown in Fig. 7 (bottom). This is due to the fact that the fitting curve is not allowed to change the side of an obstacle once the constraints are active. For a sample point  $\mathbf{x}_c(t_j)$ , the side conditions of Equ. (11) separate the parameter space  $U$  in two half spaces from which the one containing the other side of the obstacle is marked forbidden. Sometimes, one can ease this limitation by including only side conditions for sample points close to an obstacle's boundary. However, these constraints were still too strong in this example.

The approximation of the outer boundary reveals another important aspect. Left and right of the biggest hole of the surface the point cloud and the final fitting curve are nearly perpendicular. If all those points would be included in the computa-

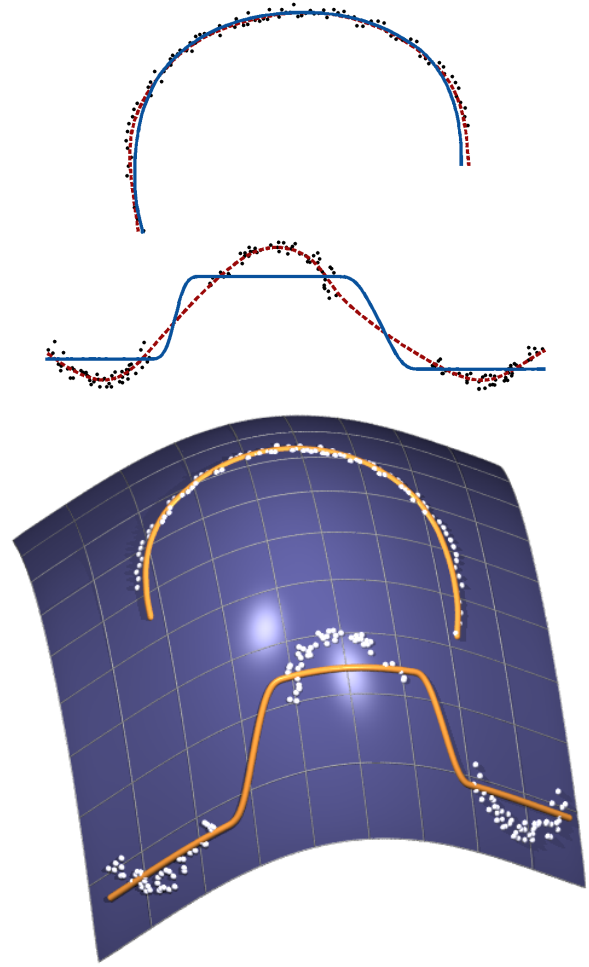


Fig. 8. Example 4. Curve fitting tangential to vector fields on surfaces, for parameter lines (bottom parts resp.) and principal curvature lines (top parts resp.). The top figure illustrates the fitting in parameter space, the dashed lines denote the unconstrained fittings.

tion of the fitting term they would become strong outliers and make the fitting unstable. For this reason, we compute the fitting error term and the constraints only for data points within a certain distance from the B-spline curve.

*Example 4.* The design constraint proposed in Sec. 3 aims for a curve fitting tangential to a given vector field. First, we define a point cloud comprising three small arc-like point clusters (see Fig. 8, bottom parts). We fit this point set and require the fitting to be tangential to one parameter direction at the same time by employing the constraints of Equ. (13). While the unconstrained fitting (shown in the parameter space as a dashed line) follows the

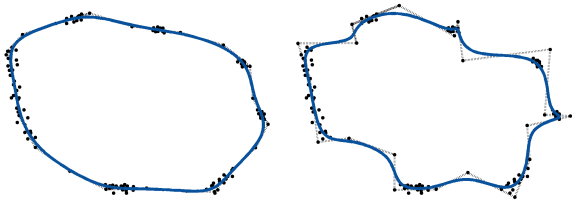


Fig. 9. Example 5. Standard (left) and length minimizing (right) curve fitting on manifolds, visualized in parameter space.

shape of the point cloud closely, the constrained fitting consists of three connected horizontal segments. All optimization parameters ( $w_0$ ,  $w_1$ , etc.) were the same for both approximations.

In general, vector field constraints are of a very strong nature. A single constraint influences the location of the control points much more than the previous side conditions. It turned out that the fitting tends to get unstable if too many constraints are added. Thus we increased the influence of the smoothing terms to smooth out any gaps and included the constraints for only every 5th foot point.

The optimization gets more challenging for the second fitting example where we used one principal curvature direction as the tangential vector field constraint (Fig. 8, top parts). While the parameter directions are straight lines in parameter space and the control points just need to line up, the principal curvature lines are of a more general nature. These constraints act very similar to smoothing forces. As for the optimization a good fit of the data points is most important, the fitting curve aligns loosely along the principal curvature directions. This approximation was ruled by 25 tangential constraints.

In difference to previous examples, we choose open B-splines as approximating curves. When approximating with open B-spline curves, one has to take care of the start and end points to avoid a shrinking of the fitting shape. Therefore, we fixed the first and last control point, either by

$$c_0^0 = c_m^0 = 0 \quad (20)$$

for the parameter direction example or by

$$c_0^1 = c_m^1 = 0 \quad (21)$$

for the principal curvature direction example. This simple strategy was sufficient for our needs.

*Example 5.* In another example we want to further discuss the influence of the smoothing term  $r_0$  of Equ. (8), based on the metric induced by the first fundamental form and minimizing the length of the

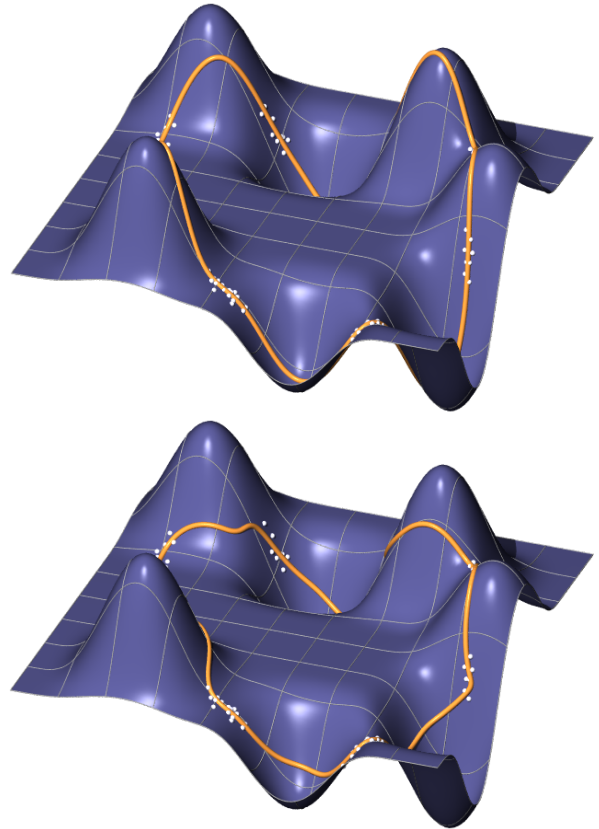


Fig. 10. Example 5. Standard (top) and length minimizing (bottom) curve fitting on manifolds.

approximating curve. We choose a parametric surface with distinct peaks and valleys and place data points between these extrema. If we set  $w_0 = 0$ , thus neglecting this smoothing term at all, the final solution is of small length in the parameter space (Fig. 9, left) but passes the peaks and valleys of the surface (Fig. 10, top). Instead, if  $r_0$  is applied by setting  $w_0 > 0$ , the curve's length gets smaller on the manifold (Fig. 10, bottom). As seen in the parameter domain (Fig. 9, right), the curve fitting is clearly influenced by the surface's geometry.

Please note that the peak in the background is rather a pyramid than a cone due to the tensor product nature of the surface. Thus the sharp bend in the curve there originates from a distinct feature of the manifold.

*Example 6.* Fig. 11 shows a point cloud with more than 5000 elements. We approximate the outer boundary of the data set with three open B-spline curves, each counting approximately 20 control points. Each curve starts at the end of a previous

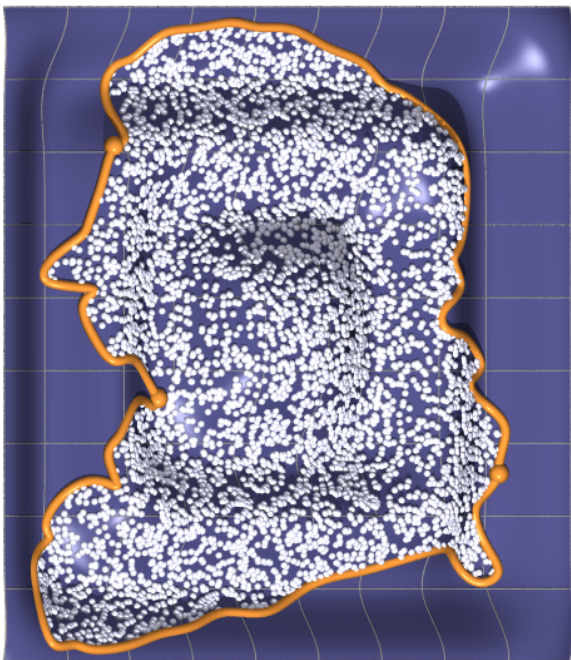


Fig. 11. Example 6. We sampled a black and white image on a surface to generate 5000 data points. Then we approximate the boundary of this point set with three open B-spline curves on the given surface (the start and end points of the three curves are marked with balls).

curve. We did not impose any further constraints on the connections (marked by bigger balls in the figure) between the segments thus getting only  $C^0$  smoothness in these points. The approximation would work as well for a single closed curve. In order to make the fitting more stable, we included only those data points within a certain distance from the fitting curve.

*Example 7.* Our last example shows another application of constrained curve fitting on manifolds. Assume that the rough shape of a curve was drawn on a surface by a designer (see Fig. 12, top) and we want to reconstruct a smooth curve from this input. As an improvement of the approach presented in Hofer and Pottmann (2004) (Figure 7), where some input points were picked and then interpolated by a spline on the manifold, we regard the corners of the input polygon as a point cloud and approximate it on the surface (Fig. 12, bottom) using the algorithm proposed in the present paper. Thus, the smoothed curve now closer resembles the shape intended by the designer than by the previous approach.

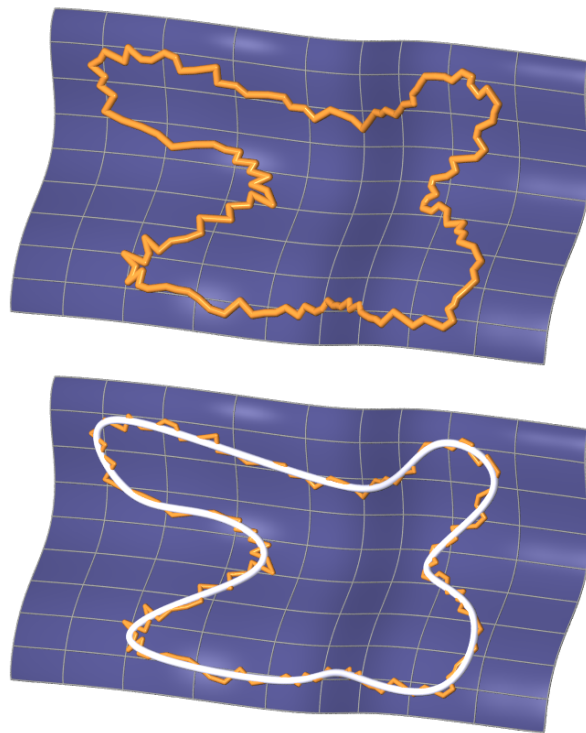


Fig. 12. Example 7. The noisy input shape on a parametric surface (top) is smoothed by fitting a B-spline curve to it (bottom).

Example	1	2	3	4	5	6	7
nr. points	150	300	250	150/100	100	5000	150
iterations	20	25	19/30/30	20/30	30	30	20
time (s)	3.8/2.5	9.8	30/37/34	25/25	15/10	13	4
$w_0$	0	0.1	0.1/1/1.2	1/1	0.01/0	1.5	0

Table 1

*Runtime and parameter details.* In Table 1 we summarize information concerning our examples. We list the number of data points, the required iterations, the computation time in seconds, and the values of the weight  $w_0$  for the regularization. The weight  $w_1$  was always chosen automatically as described in Sec. 2.3. For examples with more than one approximating curve, the values are given in the order the fitting curves are described in the text and illustrated in the figures. Thus a time of 3.8/2.5 for Example 1 means that the curve shown in Fig. 4 (bottom, left) took 3.8s to compute and the curve shown in Fig. 4 (bottom, right) took 2.5s to compute.

## 5. Conclusions and Future Research

In the present paper we have shown how to approximate unordered noisy point clouds on two-dimensional parametric surfaces by B-spline curves. The proposed iterative optimization algorithm takes into account the surface metric via the first fundamental form, which is used to compute foot points and squared distances that enter the objective function. The framework allows the inclusion of additional constraints such as obstacles that have to be avoided. For the solution of the quadratic minimization problem with linear constraints we employ an active set method. In a set of examples we illustrated applications of constrained curve fitting on manifolds and also discussed some limitations of the presented method.

Our method can be extended in several ways. Currently the number of control points of the fitting B-spline curve is set by the user. To achieve a good fit with a reasonable number of control points one sometimes has to manually increase/decrease the number of control points. Thus, an automatic adjustment of the number of control points as proposed by Yang et al. (2004) for a related fitting problem would be useful, but is beyond the scope of the current paper. Another topic is to make the obstacles *fuzzy*. Currently the fitting curve is completely pushed out of all obstacles. For some applications one would like to allow the fitting curve to penetrate the obstacle up to a certain degree. This could be achieved with a penalty type optimization strategy and is again a topic of future research. If one needs the control points of a spatial B-spline curve approximately lying on the given manifold, then one can take the curves computed in the present paper as input and use e.g. the algorithm of Renner and Weiß (2004) to compute the spatial B-spline curve control polygon.

While the current algorithm has possible applications to trajectory planning for CNC machining of two-dimensional freeform surfaces, we are also interested in a transfer from fitting on two-dimensional surfaces to the higher-dimensional case. One problem of immediate interest is the approximation of a noisy rigid body motion by a smooth one, and furthermore, the rigid body motion shall avoid given obstacles. However, it is non-trivial to generalize the present algorithm to the motion design case and we hope to address this in future research.

## Acknowledgements

We would like to thank the Austrian Science Fund (FWF) for supporting our work under grants P16002 and P18865. We further acknowledge support by Vienna University of Technology via the Innovative Project “Geometric Model Building with CNC Technology”.

## References

- Ahn, S. J., 2004. Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space. Vol. 3151 of Lecture Notes in Computer Science. Springer.
- Ambrosio, L., Montegazza, C., 1998. Curvature and distance function from a manifold. *J. Geom. Analysis* 8, 723–748.
- Atieg, A., Watson, G., 2003. A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances. *J. of Comp. Appl. Math.* 158, 277–296.
- Benkő, P., Kós, G., Várady, T., Andor, L., Martin, R., 2002. Constrained fitting in reverse engineering. *Comput. Aided Geom. Design* 19, 173–205.
- Bercovier, M., Jacobi, A., 1994. Minimization, constraints and composite Bézier curves. *Comput. Aided Geom. Design* 11 (5), 533–563.
- Blake, A., Isard, M., 1998. *Active Contours*. Springer-Verlag, NJ.
- Bohl, H., 1999. *Kurven minimaler Energie auf getrimmten Flächen*. Ph.D. thesis, Univ. Stuttgart.
- do Carmo, M., 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.
- Flöry, S., 2006. Fitting curves and surfaces to point clouds in the presence of obstacles. Tech. Rep. 160, Geometry Preprint Series, Vienna Univ. of Technology.
- Goshtasby, A. A., 2000. Grouping and parameterizing irregularly spaced points for curve fitting. *ACM Trans. Graph.* 19 (3), 185–203.
- Helrich, H.-P., Zwick, D., 1996. A trust region algorithm for parametric curve and surface fitting. *J. of Comp. Appl. Math.* 73, 119–134.
- Hildebrandt, K., Polthier, K., Preuss, E., 2005. Evolution of 3d curves under strict spatial constraints. *CAD-CG'05* 0, 40–45.
- Hofer, M., Pottmann, H., 2004. Energy-minimizing splines in manifolds. *ACM Trans. on Graph.* 23 (3), 284–293.
- Hoschek, J., 1988. Intrinsic parametrization for ap-

- proximation. *Comput. Aided Geom. Design* 5 (1), 27–31.
- Hu, S.-M., Wallner, J., 2005. A second order algorithm for orthogonal projection onto curves and surfaces. *Comput. Aided Geom. Design* 22 (3), 251–260.
- Kelley, C. T., 1999. *Iterative Methods for Optimization*. No. 18 in *Frontiers in Applied Mathematics*. SIAM, Philadelphia.
- Lee, I.-K., 2000. Curve reconstruction from unorganized points. *Comput. Aided Geom. Design* 17 (2), 161–177.
- Liu, Y., Pottmann, H., Wang, W., 2006. Constrained 3D shape reconstruction using a combination of surface fitting and registration. *Computer-Aided Design* 38 (6), 572–583.
- Liu, Y., Yang, H., Wang, W., 2005. Reconstructing B-spline curves from point clouds – a tangential flow approach using least squares minimization. In: Spagnuolo, M., Pasko, A., Belyaev, A. (Eds.), *Proc. SMA. IEEE Computer Society*, pp. 4–12.
- Machado, L., May 2006. Least square problems on Riemannian manifolds. Phd thesis, University of Coimbra.
- Machado, L., Leite, F. S., 2006. Fitting smooth paths on Riemannian manifolds. *Int. J. Appl. Math. & Stat.* 4 (J06), 25–53.
- Meek, D. S., Ong, B. H., Walton, D. J., 2003. Constrained interpolation with rational cubics. *Computer-Aided Design* 20, 253–275.
- Myles, A., Peters, J., 2005. Threading splines through 3d channels. *Computer-Aided Design* 37, 139–148.
- Noakes, L., Heinzinger, G., Paden, B., 1989. Cubic splines on curved spaces. *IMA J. Math. Cont. & Inf.* 6, 465–473.
- Nocedal, J., Wright, S. J., 1999. *Numerical Optimization*. Springer.
- Opfer, G., Oberle, H. J., 1988. The derivation of cubic splines with obstacles by methods of optimization and optimal control. *Numer. Math.* 52, 17–31.
- Peters, J., Wu, X., 2004. SLEVEs for planar spline curves. *Comput. Aided Geom. Design* 21, 615–635.
- Plass, M., Stone, M., 1983. Curve-fitting with piecewise parametric cubics. In: *Proc. SIGGRAPH '83*. ACM Press, New York, NY, USA, pp. 229–239.
- Pottmann, H., Hofer, M., 2003. Geometry of the squared distance function to curves and surfaces. In: Hege, H.-C., Polthier, K. (Eds.), *Visualization and Mathematics III*. Springer, pp. 223–244.
- Pottmann, H., Leopoldseder, S., Hofer, M., Steiner, T., Wang, W., 2005. Industrial geometry: recent advances and applications in CAD. *Computer-Aided Design* 37 (7), 751–766.
- Renner, G., Weiß, V., 2004. Exact and approximate computation of B-spline curves on surfaces. *Computer-Aided Design* 36 (4), 351–362.
- Saux, E., Daniel, M., 2003. An improved Hoschek intrinsic parametrization. *Comput. Aided Geom. Design* 20 (8-9), 513–521.
- Tsai, Y. R., 2002. Rapid and accurate computation of the distance function using grids. *J. Comput. Phys.* 178 (1), 175–195.
- Wang, W., Pottmann, H., Liu, Y., 2006. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graphics* 25 (2), 214–238.
- Yang, H., Wang, W., Sun, J., 2004. Control point adjustment for B-spline curve approximation. *Computer-Aided Design* 36 (7), 639–652.

**Simon Flöry** received his M.Sc. in Mathematics in 2005 at Vienna University of Technology where he is currently pursuing his PhD. He is a member of the Geometric Modeling and Industrial Geometry research group in Vienna. His research interests include computer aided geometric design, geometric modeling, and optimization.

**Michael Hofer** received his Ph.D. in Mathematics from Vienna University of Technology in 2005 and then worked as a Postdoc at the University of Minnesota. He is currently a Postdoc at the Geometric Modeling and Industrial Geometry research group in Vienna. His research focuses on application oriented fundamental research in geometry, geometric modeling and processing, and imaging sciences. A list of publications is available online at <http://www.geometrie.tuwien.ac.at/hofer>.