Approximate Implicitization of Space Curves

M. Aigner, B. Jüttler, Adrien Poteaux

Approximate Implicitization of Space Curves

Martin Aigner, Bert Jüttler and Adrien Poteaux

Abstract

The process of implicitization generates an implicit representation of a curve or surface from a given parametric one. This process is potentially interesting for applications in Computer Aided Design, where the robustness and efficiency of intersection algorithm can be improved by simultaneously considering implicit and parametric representations. This paper gives an brief survey of the existing techniques for approximate implicitization of hyper surfaces. In addition it describes a framework for the approximate implicitization of space curves.

Keywords: Approximate implicitization, space curves.

1 Introduction

There exist two main representations of curves and surfaces in Computer Aided Geometric Design: the implicit and the parametric form. In both cases, the functions which describe the curve or surface are almost always chosen as polynomial or rational functions or, more generally, as polynomial or rational spline functions [15]. Consequently, one deals with segments and patches of algebraic curves and surfaces.

Each of the two different representation is particularly well suited for certain applications. Parametric representations are well suited to generate points, e.g., for displaying curves and surfaces, and to apply the results of the classical differential geometry of curves and surfaces, e.g., for shape interrogation. Implicit representations encompass a larger class of shapes and are more powerful for certain geometric queries. Moreover, the class of algebraic curves and surfaces is closed under certain geometric operations, such as offsetting, while the class of rational parametric curves and surfaces is not.

Consequently, it is often desirable to change from one representation to the other one. For instance, the implicitization of a planar curve reduces the computation of the intersection of two curves given in the parametric form to find the roots of a single polynomial [23].

The exact conversion procedures, implicitization and parameterization, have been studied in classical algebraic geometry and in symbolic computation. Their practical application in Computer Aided Design is rather limited, due to the feasibility reasons outlined below. As an alternative, approximate techniques have emerged recently. These alternatives contribute to the use of symbolic-numerical techniques in Computer Aided Geometric Design.

The remainder of this paper consists of four parts. First we introduce the notation. Section 3 then presents a survey of related techniques for the approximate implicitization of hypersurfaces. The following section describes a new framework for the approximate implicitizaton of space curves. Finally we conclude this paper.

2 Preliminaries

We start by introducing a few notations. A parametric representation of a curve segment or a surface patch is a mapping

$$\mathbf{p}: \Omega \to \mathbb{R}^d: \mathbf{t} \mapsto \mathbf{p}(\mathbf{t}) \tag{1}$$

where $\Omega \subset \mathbb{R}^k$ is the parameter domain (typically a closed interval in $\mathbb{R}$ or a box in $\mathbb{R}^2$). A curve or surface is described for $k = 1$ and $k = 2$, respectively. In many applications, e.g. in Computer-Aided Design, the mapping $\mathbf{p}$ is represented by piecewise rational functions (rational spline functions), see [15].

An implicitly defined hypersurface $\mathscr{F}$ in $\mathbb{R}^d$ is the zero-set of a function $f_\mathbf{s}: \mathbb{R}^d \to \mathbb{R}$,

$$\mathscr{F} = \{\mathbf{x} \in \mathbb{R}^d : f_\mathbf{s}(\mathbf{x}) = 0\}. \tag{2}$$

If $d = 3$ or $d = 2$, then it is called an implicitly defined *surface* or *planar curve*, respectively.

The subscript represents a vector $\mathbf{s} \in \mathbb{R}^N$ which collects the parameters which characterize the function $f_\mathbf{s}(\mathbf{x})$. They are called the *shape parameters*, since they control the shape of the curve or surface. For instance, if $f_\mathbf{s}$ is a polynomial of some finite degree,

$$f_\mathbf{s}(\mathbf{x}) = \sum_{i=1}^{N} s_i \, \phi_i(\mathbf{x}), \tag{3}$$

then $\mathbf{s} = (s_1, \ldots, s_N)$ contains the coefficients with respect to a suitable basis $(\phi_i)_{i=0}^N$ of the space of polynomials.

An implicitly defined space curve

$$\mathscr{C} = \{\mathbf{x} \in \mathbb{R}^3 : f_\mathbf{s}(\mathbf{x}) = 0 \wedge g_\mathbf{s}(\mathbf{x}) = 0\}. \tag{4}$$

is defined by two intersecting implicitly defined surfaces $\mathscr{F}$ and $\mathscr{G}$, see Fig. 1. Clearly, $f_\mathbf{s}$ and $g_\mathbf{s}$ are not unique. This space curve is said to be *regular* at point $\mathbf{x} \in \mathscr{F} \cap \mathscr{G}$, if there exists a representation (4) such that the two gradient vectors $\nabla_\mathbf{x} f_\mathbf{s}(\mathbf{x})$ and $\nabla_\mathbf{x} g_\mathbf{s}(\mathbf{x})$ with $\nabla_\mathbf{x} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ are linearly independent.

Typically, the two functions defining $\mathscr{F}$ and $\mathscr{G}$ are characterized by two independent sets of shape parameters, say $\mathbf{s}_f$ and $\mathbf{s}_g$. In order to simplify the notation, we shall use the convention that both functions depend on the union of these two sets, hence on $\mathbf{s} = \mathbf{s}_f \cup \mathbf{s}_g$. If the two functions $f_\mathbf{s}(\mathbf{x})$ and $g_\mathbf{s}(\mathbf{x})$ are polynomials, then $\mathscr{C}$ is said to be an *algebraic space curve*.

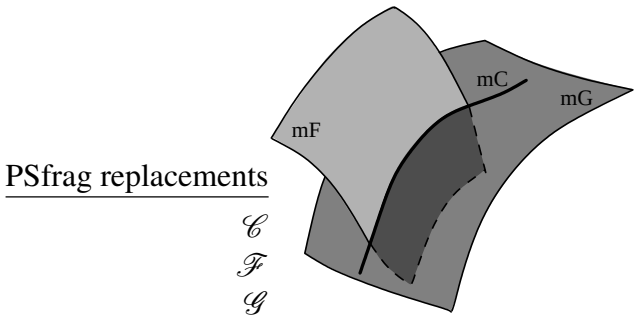


Figure 1: An implicitly defined space curve

3 Approximate Implicitization

Exact techniques for the implicitization of curves and surfaces have been studied for a long time. In 1862, Salmon [20] noted that the surface implicitization can be performed by eliminating the parameters. This was improved by Dixon in 1908 [8], who published a more compact resultant for eliminating two variables from three polynomials. In 1983, Sederberg [21] considered the implicitization of surface patches for Computer Aided Geometric Design.

From a theoretical point of view, the problem of the implicitization of a given rational curve or surface is always solvable. However, there remains a number of challenging computational difficulties. As described in [15, chapter 12], while the 2D case can be handled satisfactorily by building the Bezout resultant, the 3D case is more complicated: for instance, a tensor product surface of degree (m,n) leads to an implicit formula of degree $2mn$. Then, in the simple case $m = n = 3$, we already have an algebraic representation of degree 18. After expanding this polynomial in monomial basis this would lead to 1330 terms.

Practical problems associated with the exact implicitization of curves and surfaces are addressed in [22] and [5]. Gröbner bases can also be used [7]. For more details on resultant based methods, the reader may also consult [6].

To conclude, as shown in [22, 15], exact implicitization has many associated difficulties, in particular in the case of surfaces. Moreover, the computed implicit form of a curve or surface can be difficult to use, since the degree of the polynomial is often too high. On the other hand, CAD (Computer-Aided Design) systems are based on floating point computations, and so all quantities are represented with a rounding error. Therefore, if we apply any of the exact implicitization method in this context, the result is not exact.

The existing techniques for approximate implicitization can be classified as direct ones, where the result is found in a single step, and evolution-based techniques, where an iterative process is needed to find the result.

3.1 Direct techniques

We describe three approaches to approximate implicitization. The first two approaches are due to Dokken, who also coined the notion of AI. The third approach comprises various fitting-based

techniques.

Dokken's method. In order to adapt implicitization to the need for approximate computation in CAD, and to achieve more practical algorithms, Dokken introduced the approximate implicitization of a curve or surface [9, 10]. In the sequel we recall Dokken's method to compute the approximate implicitization of a curve or surface. See also [12] for a survey of these and related techniques.

Given a parametric curve or surface $\mathbf{p}(\mathbf{t})$, $\mathbf{t} \in \Omega$, we want to find a polynomial $f_{\mathbf{s}}(\mathbf{x}))$ such that

$$f_{\mathbf{s}}(\mathbf{p}(\mathbf{t}) + \eta(\mathbf{t})\mathbf{g}(\mathbf{t})) = 0, \tag{5}$$

where $\mathbf{g}(\mathbf{t})$ is a continuous direction function satisfying $\|\mathbf{g}(\mathbf{t})\|_2 = 1$ and $\eta(\mathbf{t})$ a continuous error function with $|\eta(\mathbf{t})| \leq s$ (see [9, Definition 35]). We denote by n the degree of the parametrization $\mathbf{p}$ and by m the degree of $f_{\mathbf{s}}$.

Dokken observes that the composition $f_{\mathbf{s}} \circ \mathbf{p}$ can be factorized as

$$f_{\mathbf{s}}(\mathbf{p}(\mathbf{t})) = (\mathbf{D}\mathbf{s})^T \alpha(\mathbf{t}), \tag{6}$$

where $\mathbf{D}$ is a matrix build from certain products of the coordinate functions of $\mathbf{p}(\mathbf{t})$, $\mathbf{s}$ is the vector of parameters that characterize the function $f_{\mathbf{s}}(\mathbf{x})$, and $\alpha(\mathbf{t}) = (\alpha_1, \ldots, \alpha_N(t))^T$, where N is the dimension of polynomial space, is the basis of the space of polynomials of degree mn, which is used to describe $f_{\mathbf{s}}(\mathbf{p}(\mathbf{t}))$.

This basis is assumed to form partition of unity,

$$\sum_{i=1}^{N} \alpha_i = 1$$

and in addition, the basis $\alpha(\mathbf{t})$ is assumed to be nonnegative for $\mathbf{t} \in \Omega$:

$$\alpha_i \geq 0, \quad \forall i, \forall t \in \Omega.$$

For instance, one may use the Bernstein-Bézier basis with respect to the interval Ω or with respect to a triangle which contains Ω in the case of curves and surfaces, respectively.

Consequently we obtain that

$$|f_{\mathbf{s}}(\mathbf{p}(\mathbf{t}))| = |(\mathbf{D}\mathbf{s})^T \alpha(\mathbf{t})| \leq \|\mathbf{D}\mathbf{s}\|_2 \|\alpha(\mathbf{t})\|_2 \leq \|\mathbf{D}\mathbf{s}\|_2, \tag{7}$$

hence we are led to find a vector $\mathbf{s}$ which makes $\|\mathbf{D}\mathbf{s}\|_2$ small. Using the Singular Value Decomposition (SVD) of the matrix $\mathbf{D}$, one can show that $\|f_{\mathbf{s}_1}(\mathbf{p}(\mathbf{t}))\|_\infty \leq \sqrt{\sigma_1}$, where σ_1 is the smallest singular value, and $\mathbf{s}_1$ is the corresponding singular vector. This strategy enables the use of Linear Algebra tools to solve the problem of approximate implicitization. Moreover, this approach provides high convergence rates, see [12, Table 1 and 2].

Dokken's weak method. Dokken's original method has several limitations: for instance, it is relatively costly to build the matrix $\mathbf{D}$. Moreover, it is impossible to use spline functions for describing $f_{\mathbf{s}}$, since no suitable basis for the composition $f_{\mathbf{s}} \circ \mathbf{p}$ can be found.

This problem can be avoided by using the *weak form* of approximate implicitization which was introduced in [11], see also [12, section 10]. For a given curve or surface $\mathbf{p}$ with parameter domain Ω, we now find the approximate implicitization by minimizing

$$\int_{\Omega} (f_{\mathbf{s}}(\mathbf{p}(\mathbf{t})))^2 \mathrm{d}\mathbf{t} = \mathbf{s}^T \mathbf{A} s \tag{8}$$

where

$$\mathbf{A} = \mathbf{D}^T \left(\int_{\Omega} \alpha(\mathbf{t}) \alpha(\mathbf{t})^T \mathrm{d}\mathbf{t} \right) \mathbf{D}. \tag{9}$$

The matrix $\mathbf{A}$ can be analyzed by eigenvalue decomposition, similar to the original approach, where the matrix $\mathbf{D}$ was analyzed with singular value decomposition. Note that one can apply this strategy even if no explicit expression is available: one only needs to be able to evaluate points on the curve or surface. The integrals can then be approximately evaluated by numerical integration.

Choosing the eigenvector which is associated with the smallest eigenvalue of the matrix $\mathbf{A}$ is equivalent to minimizing the objective function defined in (8) subject to the constraint $\|\mathbf{s}\| = 1$. This can be seen as a special case of fitting, see next section.

Algebraic curve and surface fitting. Given a number of points $(\mathbf{p}_i)_{i=1}^N$, which have been sampled from a given curve or surface, one may fit a curve or surface by minimizing the sum of the squared residuals (also called algebraic distances),

$$\sum_{i=1}^{N} (f_{\mathbf{s}}(\mathbf{p}_i))^2. \tag{10}$$

This objective function can be obtained by applying a simple numerical integration to (8).

If the algebraic curve or surface is given as in (3), then this objective function has the trivial minimum $\mathbf{s} = \mathbf{0}$. In order to obtain a meaningful result by minimizing (10), several additional constraints have been introduced.

Pratt [19] picks one of the coefficients and restricts it to 1, e.g.

$$s_1 = 1. \tag{11}$$

For instance, if $f_{\mathbf{s}}$ is a polynomial which is represented with respect to the usual power basis, then one may consider the absolute term. This constraint is clearly not geometrically invariant, since the curve and surface cannot pass through the origin of the system of coordinates.

Geometrically invariant constraints can be obtained by considering quadratic functions of the unknown coefficients $\mathbf{s}$. An interesting normalization has been suggested by Taubin [24], who

proposed to use the norm of the squared gradient vectors at the given data,

$$\sum_{i=1}^{N} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_i)\|^2 = 1. \tag{12}$$

Adding this constraint leads to a generalized eigenvalue problem. Taubin's method gives results which are independent of the choice of the coordinate system.

Finally, Dokken's weak method – when combined with numerical integration for evaluating the objective function (8) – uses the constraint

$$\|\mathbf{s}\|^2 = \sum_{i=1}^{N} s_i^2 = 1. \tag{13}$$

These three approaches are able to provide meaningful solutions which minimize the squared algebraic distances (10). However, they may still lead to fairly unexpected results. Additional branches and isolated singular points may be present, even for data which are sampled from regular curves or surfaces.

If a method for approximate implicitization is to reproduce the exact results for sufficiently high degrees, then this unpleasant phenomenon is always present. For instance, consider a cubic planar curve with a double point. Even if we take sample points only from one of the two branches which pass through the singular point, any of the above-mentioned methods will generate the cubic curve with the double point, provided that the degree of $f_{\mathbf{s}}$ is at least 3.

These difficulties can be avoided by using additional normal (or gradient) information. More precisely, a nontrivial solution of the minimization problem can be found by considering a convex combination of the two objective functions (8) and

$$\sum_{i=1}^{N} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_i) - \mathbf{n}_i\|^2, \tag{14}$$

where the vectors $(\mathbf{n}_i)_{i=1}^N$ represent additional normal vector information at the given points.

This gives a quadratic function of the unknown coefficients $\mathbf{s}$, hence the minimum is found by solving a system of linear equations. This approach has been introduced in [16], and it has later been extended in [17, 27, 26]. Among other topics, these papers also consider the case of curves which contain singular points, where a globally consistent propagation of the normals is needed.

3.2 Iterative (evolution-based) techniques

Iterative (evolution-based) methods have been considered for several reasons. First, they lead to a uniform framework for handling various representations of curves and surfaces, which can handle implicitly defined curves and surfaces as well as parametric ones [1, 13]. Second, they make it possible to include various conditions, such as constraints on the gradient field, volume constraints or range constraints [28, 14, 29]. Finally, the sequence of curves or surfaces generated

by an iterative method can be seen as discrete instances of a continuous evolution process, which links this approach to the level set method and to active curves and surfaces in Computer Vision [18, 4].

We recall the evolution-based framework for fitting point data $(\mathbf{p}_j)_{j=1,\dots,M}$ with implicitly defined hypersurfaces, which was described in [1]. In this framework, the approximate solutions which are generated by an iterative algorithm are seen as discrete instances of a continuous movement of an initial curve or surface towards the target points (the given point data).

More precisely, we assume that the shape parameters $\mathbf{s}$ depend on a time-like parameter t, and consider the evolution of the hypersurface described by the parameters $\mathbf{s}(t)$ for $t \to \infty$. Each data point $\mathbf{p}_j$ attracts a certain point $\mathbf{f}_j$ on the hypersurface $\mathscr{F}$ which is associated with it. Usually $\mathbf{f}_j$ is chosen to be the closest point on $\mathscr{F}$, i.e.

$$\mathbf{f}_j = \arg\min_{\mathbf{p}\in\mathscr{F}} \|\mathbf{p}-\mathbf{p}_j\|. \tag{15}$$

These attracting forces push the time-dependent hypersurface towards the data. This is realized by assigning certain velocities to the points on the hypersurface. For a point lying on a time-dependent implicitly defined curve or surface, which is described by a function $f_{\mathbf{s}}$, the normal velocity is given by

$$\mathbf{v} = -\frac{\partial f_{\mathbf{s}}}{\partial t}\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}^{\top}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|^2} = -\nabla_{\mathbf{s}} f_{\mathbf{s}}\,\dot{\mathbf{s}}\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}^{\top}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|^2}, \tag{16}$$

where the dot indicates the derivative with respect to t and the gradient operator

$$\nabla_{\mathbf{s}} = \left(\frac{\partial}{\partial s_1},\dots,\frac{\partial}{\partial s_N}\right) \tag{17}$$

gives the row vector of the first partial derivatives. Note that we omitted the time dependency of $\mathbf{s}$ in (16), in order to simplify the notation.

The first term $-\nabla_{\mathbf{s}} f_{\mathbf{s}}\,\dot{\mathbf{s}}$ in (16) specifies the absolute value of the normal velocity. The second term is the unit normal vector of the curve, which identifies the direction of the velocity vector.

As the number of data points exceeds in general the degrees of freedom of the hypersurface, the velocities are found as the least squares solution of

$$\sum_{j=1}^{M} ((\mathbf{v}_j - \mathbf{d}_j)^{\top}\mathbf{n}_j)^2 \to \min_{\dot{\mathbf{s}}}, \tag{18}$$

where $\mathbf{d}_j = \mathbf{f}_j - \mathbf{p}_j$ is the residual vector from a data point to its associated point on the hypersurface, $\mathbf{n}_j = \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|}$ is the unit normal in this point and $\mathbf{v}_j$ is the velocity computed via (16) at $\mathbf{f}_j$. More precisely, this leads to the minimization problem

$$\sum_{j=1}^{M}\left((\frac{(\nabla_{\mathbf{s}} f_{\mathbf{s}})(\mathbf{p}_j)\,\dot{\mathbf{s}}\,(\nabla_{\mathbf{x}} f_{\mathbf{s}})(\mathbf{p}_j)}{\|(\nabla_{\mathbf{x}} f_{\mathbf{s}})(\mathbf{p}_j)\|^2} - (\mathbf{f}_j-\mathbf{p}_j)^{\top})\frac{(\nabla_{\mathbf{x}} f_{\mathbf{s}})(\mathbf{p}_j)^{\top}}{\|(\nabla_{\mathbf{x}} f_{\mathbf{s}})(\mathbf{p}_j)\|}\right)^2 \to \min_{\dot{\mathbf{s}}}. \tag{19}$$

We use Tikhonov regularization in order to obtain a unique solution. In addition, we apply a distance field constraint, in order to avoid the trivial solution, cf. [28].

The geometric interpretation of this approach is as follows: The bigger the distance to the associated data point, the greater is the velocity that causes the movement of the hypersurface at the corresponding point. Note that (18) takes only the normal component of the velocity into account, as a tangential motion does not change the distance to the data.

The objective function in Eq. (19) depends on $\mathbf{s}$ as well as on $\dot{\mathbf{s}}$. For a given value of $\mathbf{s}$, we can find $\dot{\mathbf{s}}$ by solving a system of linear equations. Consequently, (19) leads to an ordinary differential equation for the vector of shape parameters. We can solve it by using Euler steps with a suitable stepsize control, see [1] for details.

The solution converges to a stationary point, which defines the solution of the fitting problem. It can be shown that this evolution-based approach is equivalent to a Gauss-Newton method for the implicit fitting problem, and the stationary point of the ODE is a (generally only) local minimum of the objective function

$$\sum_{j=1}^{M} ||\mathbf{p}_j - \mathbf{f}_j||^2, \tag{20}$$

where $\mathbf{f}_j$ has been defined in (15), see [2].

The evolution viewpoint has several advantages. It provides a geometric interpretation of the initial solution, which is now seen as the starting point of an evolution that drives the hypersurface towards the data. It also provides a geometrically motivated stepsize control, which is based on the velocity of the points during the evolution (see [1]). Finally, the framework makes it possible to introduce various other constraints on the shape of the hypersurface, see [13, 14].

In the remainder of this paper we will apply the evolution framework to the approximate implicitization of space curves. In this situation we need to generate two surfaces which intersect in the given space curve. Moreover, these two surfaces should intersect transversely, in order to obtain a robustly defined intersection curve.

4 Approximate implicitization of space curves

Now we consider a point cloud $(\mathbf{p}_j)_{j=1,\dots,M}$ which has been sampled from a space curve. Recall that a point $\mathbf{p}_j$ lies on an implicitly defined space curve $\mathscr{C}$ if it is contained in both surfaces defining the curve. Consequently we fit the spatial data with two surfaces $\mathscr{F}$ and $\mathscr{G}$. The desired solution $\mathscr{C}$ is then contained in the intersection of $\mathscr{F}$ and $\mathscr{G}$. We need to couple the fitting of the two surfaces, in order to obtain a well-defined intersection curve.

4.1 Fitting two implicitly defined surfaces

Following the idea in [2] we use an approximation of the exact geometric distance from a data point to a space curve. More precisely, we use the Sampson distance which was originally introduced for the case of hypersurfaces [25]. The oriented distance from a point $\mathbf{p}_j$ to a curve or surface which is defined implicitly as the zero set of some function $f_{\mathbf{s}}$ can be approximated by

$$\frac{f_{\mathbf{s}}(\mathbf{p}_j)}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_j)\|}. \tag{21}$$

Geometrically speaking, the equation of the surface is linearized in the point $\mathbf{p}_j$ and the distance from this point to the zero-set of the linearization is taken as an approximation of the exact distance. Consequently, this measure is exact for planes, as they coincide with their linearization. The Sampson distance is not defined at points with vanishing gradients, which have to be excluded.

A natural extension of this distance to two surfaces defining a space curve is

$$d_j = \sqrt{\frac{f_{\mathbf{s}}(\mathbf{p}_j)^2}{\|\nabla_{\mathbf{x}} \mathbf{s}(\mathbf{p}_j)\|^2} + \frac{g_{\mathbf{s}}(\mathbf{p}_j)^2}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}(\mathbf{p}_j)\|^2}}. \tag{22}$$

If both surfaces intersect each other orthogonally, then this expression approximates the distance to the implicitly defined space curve.

In order to approximate a set of points which has been sampled from a space curve, we minimize the sum of the squared distances, which leads to the objective function

$$\sum_{j=1}^{M} d_j^2 = \sum_{j=1}^{M} \frac{f_{\mathbf{s}}(\mathbf{p}_j)^2}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_j)\|^2} + \frac{g_{\mathbf{s}}(\mathbf{p}_j)^2}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}(\mathbf{p}_j)\|^2} \to \min_{\mathbf{s}}. \tag{23}$$

Note that both functions $f_{\mathbf{s}}$ and $g_{\mathbf{s}}$ depend formally on the same vector $\mathbf{s}$ of shape parameters. Typically, each shape parameter s_i is uniquely associated with either $f_{\mathbf{s}}$ or $g_{\mathbf{s}}$. Consequently, (23) minimizes the Sampson distances from a point $\mathbf{p}_j$ to each of the surfaces $\mathscr{F}$ and $\mathscr{G}$ independently.

We adapt the evolution based-framework [2] in order to deal with the objective function (23). We consider the combination of the two evolutions for $\mathscr{F}$ and $\mathscr{G}$ which is defined by the minimization problem $E \to \min_{\dot{\mathbf{s}}}$, where

$$E(f,g) = \sum \left(\frac{f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} + \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \dot{\mathbf{s}} \right)^2 + \left(\frac{g_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} + \frac{\nabla_{\mathbf{s}} g_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \dot{\mathbf{s}} \right)^2. \tag{24}$$

In order to simplify the notation, we omit the argument $\mathbf{p}_j$ from now on and omit the range of the sum, which is taken over all sampled points $(\mathbf{p}_j)_{j=1,\dots,M}$. This sum can also be seen as simple numerical integration along the given space curve.

The geometric meaning of this objective function is as follows: The normal velocity (cf. (16)) of the level set of $f_{\mathbf{s}}$ (and analogously for $g_{\mathbf{s}}$) which passes through the given point $\mathbf{p}_j$ is to be equal to the estimated oriented distance, see (21), to the surface. Later we will provide another interpretation of this evolution as a Gauss-Newton-type method.

Similar to Eq. (19), the objective function in Eq. (24) depends on $\mathbf{s}$ and on $\dot{\mathbf{s}}$. For a given value of $\mathbf{s}$, we find $\dot{\mathbf{s}}$ by solving a system of linear equations. Consequently, (24) leads to an ordinary differential equation for the vector of shape parameters. We can again solve it simply by using Euler steps with a suitable stepsize control.

As a necessary condition for a minimum of (24), the first derivatives with respect to the vector $\dot{\mathbf{s}}$ have to vanish. This yields the linear system

$$\sum \left[\frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} + \frac{\nabla_{\mathbf{s}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{s}} g_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \right] \dot{\mathbf{s}} = -\sum \frac{f_{\mathbf{s}} \nabla_{\mathbf{s}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|^2} + \frac{g_{\mathbf{s}} \nabla_{\mathbf{s}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|^2}. \tag{25}$$

If there exists a zero-residual solution, then the right hand side vanishes, as $f_{\mathbf{s}}(\mathbf{p}_j) = g_{\mathbf{s}}(\mathbf{p}_j) = 0$ for all j. Hence $\dot{\mathbf{s}} = 0$ is a solution for the problem and we have reached a stationary point of the evolution. However, the solution may not be unique.

First, the trivial (and unwanted) functions $f_{\mathbf{s}} \equiv 0$ and $g_{\mathbf{s}} \equiv 0$ solve always the minimum problem (23) for all data sets $(\mathbf{p}_j)_{j=1\ldots M}$. Of course these solutions have to be avoided.

Second, the evolution defined via (24) pushes both surfaces *independently* towards the data points $\mathbf{p}_j$. This may lead to the unsatisfying result $f_{\mathbf{s}} \equiv g_{\mathbf{s}}$ (where the two functions are identical up to a factor λ). Consequently, we need to introduce additional terms which guarantee that $f_{\mathbf{s}}$ and $g_{\mathbf{s}}$ do not vanish and that they intersect orthogonally along the data points.

4.2 Regularization

So far, the implicitization problem is not well–posed. If $f_{\mathbf{s}}$ is a solution to the problem, then $\lambda f_{\mathbf{s}}$ is a solution as well. In this section we discuss several strategies that shall prevent the functions $f_{\mathbf{s}}$ and $g_{\mathbf{s}}$ from vanishing and that shall guarantee a unique solution to the individual fitting problems for the two defining surfaces $\mathscr{F}$ and $\mathscr{G}$. Additionally, we propose a coupling term that ensures a well-defined intersection curve of the surfaces $\mathscr{F}$ and $\mathscr{G}$.

Distance field constraint. In order to avoid the unwanted solutions $f_{\mathbf{s}} \equiv 0$ and $g_{\mathbf{s}} \equiv 0$ we use the distance field constraint which was described in [28]. Consider the term

$$D(f) = \left(\frac{d}{dt} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x})\| + \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x})\| - 1 \right)^2 . \tag{26}$$

It pushes the function $f_{\mathbf{s}}$ in a point $\mathbf{x}$ closer to a unit distance field, hence

$$\|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x})\| = 1 \tag{27}$$

If the length of the gradient in (26) equals 1, it is expected to remain unchanged. Consequently, its derivative shall be 0. Otherwise (26) modifies $f_{\mathbf{s}}$ such that the norm of its gradient gets closer to 1.

We apply this penalty term to both functions $f_{\mathbf{s}}$ and $g_{\mathbf{s}}$.

This side condition has also an important influence on the robustness of the implicit representation of the two surfaces $\mathscr{F}$ and $\mathscr{G}$, cf. [3]. Roughly speaking, the closer the defining functions $f_{\mathbf{s}}$ and $g_{\mathbf{s}}$ are to a unit gradient field, the less sensible is the representation to potential errors in its coefficients.

Theoretically, this condition can be integrated over the entire domain of interest. In order to obtain a robust representation of the implicit space curve, the robustness of the two generating surfaces is mainly required along their intersection, i.e. near the data points. This leads to the idea of imposing the distance field constraint only in the data points $\mathbf{p}_j$.

We note two more observations. First, the term is quadratic in the unknowns $\dot{\mathbf{s}}$ which follows directly from expanding the derivative in (26),

$$\frac{d}{dt} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x}_j)\| = \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \nabla_{\mathbf{s}} \nabla_{\mathbf{x}} f_{\mathbf{s}}\, \dot{\mathbf{s}} \tag{28}$$

Consequently, the objective function with the distance field constrained is still quadratic in the unknowns, and we can compute the derivative vector $\dot{\mathbf{s}}$ of the shape parameters by solving a system of linear equations.

Second, the constrained problem does in general not reproduce exact solutions which would be available without any constraints. For instance, if the data were sampled from a low degree algebraic space curve, then the approximation technique would not provide an exact equation of this curve. Only if that solution possesses a unit gradient field along the data, then it can be recovered. In the next section we introduce another regularization term which makes it possible to reproduce the exact solution.

Averaged gradient constraint. This technique is related to a method that was introduced by Taubin [25]. The core idea is to restrict the sum of the norms of the gradients. Hence, not all the gradient lengths are expected to be uniform, but the average gradient length

$$\frac{1}{M}\sum \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_j)\| = 1. \tag{29}$$

This can be dealt with by adding the term

$$A(f) = \left(\sum \frac{d}{dt}\|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_j)\| + \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_j)\| - 1\right)^2 \tag{30}$$

to our framework.

Although (27) and (30) look quite similar, their effects on the solution are rather different. Note that Eq. (29) is only one constraint, whereas (27) is a set of constraints, which depends on the number of points.

Consequently, the condition on the average norm of the gradient can only handle the singularity that is due to the scalability of implicit representations. If the ambiguity of the solution arises from an incorrectly chosen degree of the polynomial, then Taubin's method and the term (30) do not provide a unique solution.

For instance, when fitting a straight line with two quadratic surfaces, the obtained linear system is singular as the number of unknowns exceeds the number of linearly independent equations provided by the data points. On the other hand, if we use the distance field constraint (26), then we will obtain a unique solution.

Orthogonality constraint. The distance field constraint leads to a robust representation of each of the two surfaces which define the curve. Now we introduce an additional term which provides a robust representation of the curve itself.

Ideally, the two surfaces would intersect orthogonally along the space curve $\mathscr{C}$, i.e.

$$(\nabla_{\mathbf{x}} f_{\mathbf{s}} \nabla_{\mathbf{x}} g_{\mathbf{s}}^{\top})\Big|_{\mathscr{C}} = 0. \tag{31}$$

In this case, small displacements in the two surfaces cause only small errors in the curve. Moreover, the term (22) then approximates the distance to the space curve very well. On the other

hand, if the two surfaces intersect tangentially, even small perturbations may cause big changes of the curve.

In order to obtain two surfaces that intersect each other approximately orthogonally, we add the term

$$O(f,g) = \sum \left(\frac{d}{dt} \left(\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{x}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \right) + \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{x}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \right)^2 \tag{32}$$

to the objective function. This term penalizes deviations from the optimal case $\nabla_{\mathbf{x}} f_{\mathbf{s}} \nabla_{\mathbf{x}} g_{\mathbf{s}}^\top = 0$. More precisely, if the gradients of the surfaces are not orthogonal in a point where (32) is applied to, then the time derivative of the product of the unit gradients forces the surfaces to restore this property. Theoretically, this term should be imposed along the intersection of the surfaces $\mathscr{F}$ and $\mathscr{G}$. As the exact intersection curve is not known, we apply (32) to the data points $\mathbf{p}_j$.

We analyze the structure of this term in more detail. The time derivative of the first product in (32) gives

$$\begin{aligned} \frac{d}{dt} \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{x}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} =& \frac{\nabla_{\mathbf{x}} \dot{f}_{\mathbf{s}} \nabla_{\mathbf{x}} g_{\mathbf{s}}^\top + \nabla_{\mathbf{x}} f_{\mathbf{s}} \nabla_{\mathbf{x}} \dot{g}_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\| \|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \\ & - \nabla_{\mathbf{x}} f_{\mathbf{s}} \nabla_{\mathbf{x}} g_{\mathbf{s}}^\top \left(\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}} \nabla_{\mathbf{x}} \dot{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|^3 \|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} + \frac{\nabla_{\mathbf{x}} g_{\mathbf{s}} \nabla_{\mathbf{x}} \dot{g}_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\| \|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|^3} \right) \end{aligned} \tag{33}$$

Since $\nabla_{\mathbf{x}} \dot{f}_{\mathbf{s}} = \nabla_{\mathbf{x}} \nabla_{\mathbf{s}} f_{\mathbf{s}} \dot{\mathbf{s}}$ and $\nabla_{\mathbf{x}} \dot{g}_{\mathbf{s}} = \nabla_{\mathbf{x}} \nabla_{\mathbf{s}} g_{\mathbf{s}} \dot{\mathbf{s}}$, the term (32) is quadratic in $\dot{\mathbf{s}}$.

4.3 Putting things together

Summing up, we obtain the minimization problem

$$F(\dot{\mathbf{s}}, \mathbf{s}) \to \min_{\dot{\mathbf{s}}} \tag{34}$$

where

$$F = E(f,g) + \omega_1 (D(f) + D(g)) + \omega_2 O(f,g) + \omega_3 (A(f) + A(g)) + \omega_4 \dot{\mathbf{s}}^2 \tag{35}$$

The non-negative weights ω_1, ω_2, ω_3 and ω_4 control the influence of the distance field constraint, the orthogonality constraint, the averaged gradient constraint and the Tikhonov regularization, respectively. Due to the special structure Eq. (35) is quadratic in the vector $\dot{\mathbf{s}}$. Hence, for a given vector $\mathbf{s}$ of shape parameters, we can find $\dot{\mathbf{s}}$ by solving a system of linear equations. The evolution of the implicit representation of the space curve can then be traced using explicit Euler steps with a suitable stepsize control (cf. [1]).

We conclude this section by discussing the coupled evolution from the optimization viewpoint. We show that the constrained optimization is in fact a Gauss-Newton method for a particular fitting problem.

Consider the optimization problem

$$
\begin{aligned}
C = \sum \left(\frac{f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \right)^2 + \left(\frac{g_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \right)^2 + \omega_1 \left((\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\| - 1)^2 + (\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\| - 1)^2 \right) \\
+ \omega_2 \left(\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{x}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \right)^2 \\
+ \omega_3 (\left(\sum \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{p}_j)\| - 1 \right)^2 + \left(\sum \|\nabla_{\mathbf{x}} g_{\mathbf{s}}(\mathbf{p}_j)\| - 1 \right)^2) \rightarrow \min_{\mathbf{s}}.
\end{aligned} \tag{36}
$$

Obviously, a solution of (36) minimizes simultaneously the Sampson distances from the data points to the space curve (term 1 and 2) the distance field constraint (term 3), the orthogonality constraint (term 4) and the averaged gradient constraint (term 5 and 6). Hence a zero residual solution of (36) interpolates all data points, the defining surfaces have slope one in the data points and furthermore, the surfaces intersect orthogonally.

Since (36) is non-linear in the vector of unknowns $\mathbf{s}$, we consider an iterative solution technique. A Gauss-Newton approach for (36) solves iteratively the linearized version of (36),

$$
C^* \rightarrow \min_{\Delta \mathbf{s}} \tag{37}
$$

where

$$
\begin{aligned}
C^* = \sum \left(\frac{f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} + \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \Delta \mathbf{s} \right)^2 + \left(\frac{g_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} + \frac{\nabla_{\mathbf{s}} g_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} \Delta \mathbf{s} \right)^2 \\
+ \omega_1 \left[(\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\| - 1 + \nabla_{\mathbf{s}} (\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\| - 1) \Delta \mathbf{s})^2 + (\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\| - 1 + \nabla_{\mathbf{s}} (\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\| - 1) \Delta \mathbf{s})^2 \right] \\
+ \omega_2 \left(\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{x}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|} + \nabla_{\mathbf{s}} (\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{x}} g_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} g_{\mathbf{s}}\|}) \Delta \mathbf{s} \right)^2 \\
+ \omega_3 (\left(\sum \|\nabla_{\mathbf{x}} f_{\mathbf{s}}\| - 1 + \nabla_{\mathbf{s}} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}\| \Delta \mathbf{s} \right)^2 + \left(\sum \|\nabla_{\mathbf{x}} g_{\mathbf{s}}\| - 1 + \nabla_{\mathbf{s}} \|\nabla_{\mathbf{x}} g_{\mathbf{s}}\| \Delta \mathbf{s} \right)^2)
\end{aligned} \tag{38}
$$

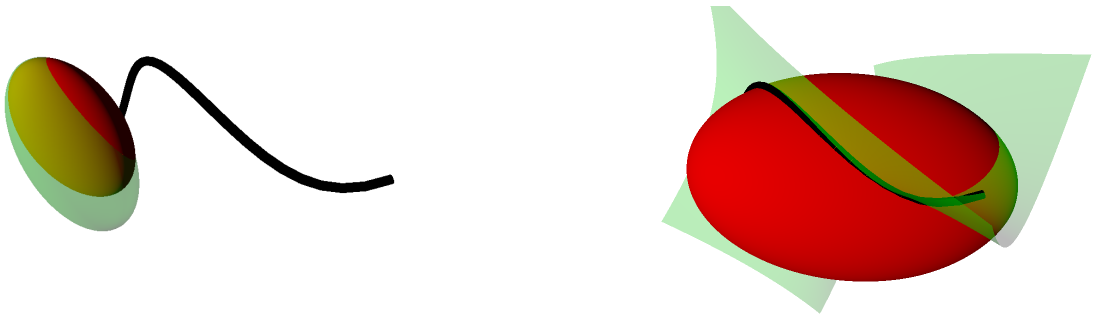
and computes an update of the previous solution via $\mathbf{s}^+ = \mathbf{s} + \Delta \mathbf{s}$. By comparing (35) and (38) we arrive at the following observation.

> *An explicit Euler step for the evolution equation* (35) *with stepsize* 1 *is equivalent to the Gauss-Newton update* (38) *for the optimization problem* (36).

Indeed, if we use that for any function $h(\mathbf{s}(t))$,

$$
\frac{d}{dt} h(\mathbf{s}(t)) = \nabla_{\mathbf{s}} h(\mathbf{s}(t)) \dot{\mathbf{s}}, \tag{39}
$$

then we can replace the time derivatives in (35). Substituting $\dot{\mathbf{s}}$ for $\Delta \mathbf{s}$ then gives the desired result.



(a) (b)

Figure 2: Implicitization of a space curve represented by data points sampled from a parametric curve. Left: Initial surfaces, right: Final result.

4.4 Examples

Finally we present some examples.

Example 1. We sampled 50 points from a parametric space curve of degree 6. The two implicit patches that represent the implicit space curve are of degree 2. As initial configuration we have chosen two surfaces deviating from each other slightly, see Figure 2(a).

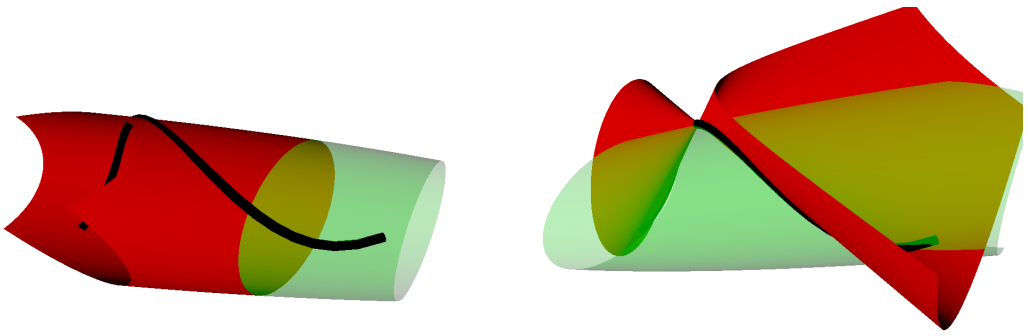
The obtained result after 15 iterations is shown in Figure 2(b). In order to demonstrate the robustness of the representation we note that the norm of the gradients of the two surfaces in the data points varies between 0.94 and 1.94. The maximal deviation of the gradients from orthogonality at the data points is 0.49 degrees.

Example 2. We choose again the same data set, but modify the various weights in order to demonstrate their influence. First we omit the orthogonality constraint. That is, the evolution is not coupled, and both surfaces move independently towards the data. The result is obvious, both surfaces converge towards the same result, as the initial values are quite similar, cf. Fig. 3(a). Alternatively, we omit the distance field constraint. The results can be seen in Fig. 3(b).

As one can verify, the two surfaces match still the data. However, one of the surfaces has a singularity. This is due to the fact that the averaged gradient constraint allows also vanishing gradients. For the distance field constraint this is not true, as the norm of the gradients in the data points is forced to be close to one, hence singular points are unlikely to appear.

Example 3. For this example we added a random error of maximal magnitude 0.05 % of the diameter of the bounding box to the data points from the previous example. The fitted space curve is represented in Fig. 4.

Example 4. In a fourth example we consider a parametric curve of degree 8. The two surfaces were chosen to have degree 3. This example shall illustrate again the good convergence behavior, as the two initial surfaces are far away from the final result.



(a) (b)

Figure 3: Result with omitted orthogonality constraint(left) and omitted distance field constraint(right).

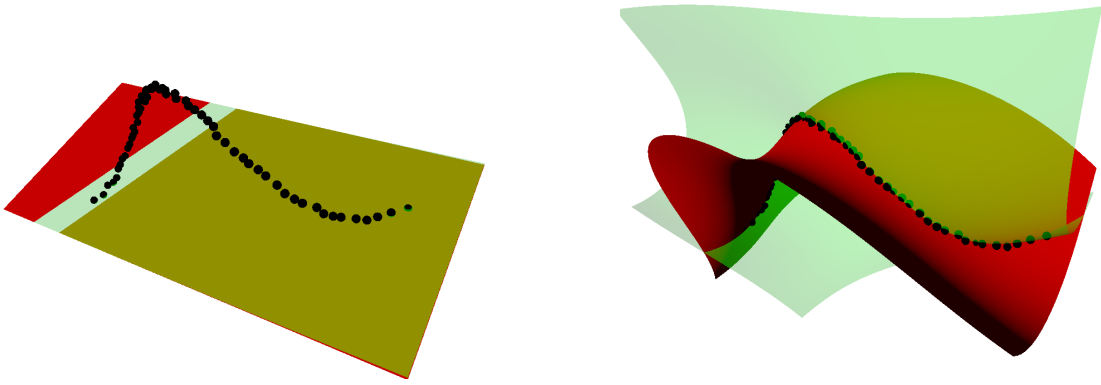


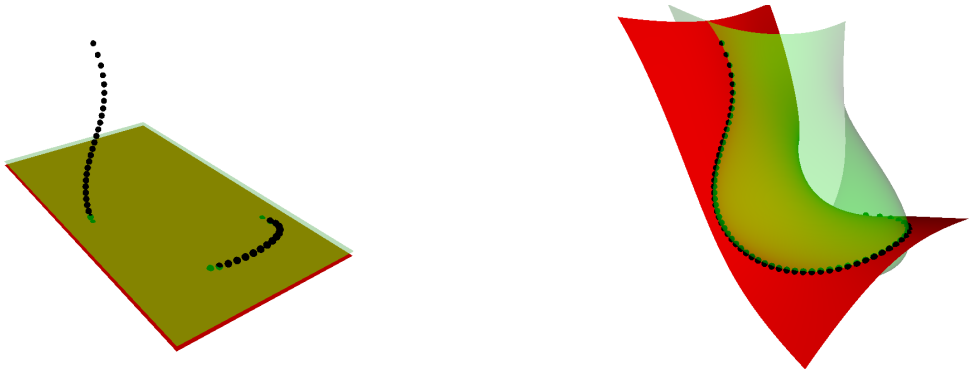
Figure 4: Implicit description of a curve represented by perturbed data. Left: Initial surfaces, right: Final result.

5 Conclusion

In the first part of the paper we reviewed some of the existing techniques for approximate implicitization of hypersurfaces. Starting with Dokken's approach, which relies on the use of singular value decomposition, we observed that the weak version of Dokken's method can be seen as a special instance of a fitting method. Finally we described a general framework for evolution based fitting techniques.

The second part of the paper extended the existing evolution framework to the implicitization of space curves, by coupling the evolution of two implicitly defined surfaces. As the implicit representation of a curve or surface is not unique, additional regularization terms have to be added in order to achieve the uniqueness of the solution. We discussed two possibilities.

The first, called the distance field constraint, tries to achieve a unit gradient field along the intersecting surfaces. Hence a unique solution to the fitting problem is always guaranteed. Furthermore, it can even cope with an incorrectly chosen degree, that is when the degrees of the



(a) (b)

Figure 5: Implicit representation of a curve described by exact point data. Left: Initial surfaces, right: Final result.

defining polynomials have been chosen too high. However, this approach prevents the evolution from finding the exact solution.

The second proposed regularization eliminates only the redundancy which is caused by the scalability of the underlying functions. As an advantage, it allows to find the exact solution, provided that the degrees of the implicitly defined surfaces are sufficiently high.

In order to obtain also a robust representation of the intersection curve we introduced another constraint which is to guarantee that the defining surfaces intersect as orthogonal as possible. Consequently, small perturbations of the coefficients of the defining functions lead only to small deviations of the intersection points of the two surfaces.

References

[1] M. Aigner and B. Jüttler. Hybrid curve fitting. *Computing*, 79:237–247, 2007.

[2] M. Aigner and B. Jüttler. Robust fitting of implicitly defined surfaces using Gauss–Newton–type techniques. *The Visual Computer*, 25:731–741, 2009.

[3] M. Aigner, B. Jüttler, and M.-S. Kim. Analyzing and enhancing the robustness of implicit representations. In *Geometric modelling and Processing*, pages 131–142. IEEE Press, 2004.

[4] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer, Secaucus, 1998.

[5] D. Cox, R. Goldman, and M. Zhang. On the validity of implicitization by moving quadrics for rational surfaces with no base points. *J. Symb. Comput.*, 29(3):419–440, 2000.

[6] D. A. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[7] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[8] A. L. Dixon. The eliminant of three quantics in two independents variables. *Proc. London Math. Soc.*, 6:49–69, 1908.

[9] T. Dokken. *Aspects of Intersection Algorithms and Approximation*. PhD thesis, University of Oslo, 1997.

[10] T. Dokken. Approximate implicitization. In *Mathematical methods for curves and surfaces*, pages 81–102. Vanderbilt Univ. Press, Nashville, TN, 2001.

[11] T. Dokken, H. K. Kellerman, and C. Tegnander. An approach to weak approximate implicitization. In *Mathematical Methods for Curves and Surfaces: Oslo 2000*, pages 103–112. Vanderbilt University, Nashville, TN, USA, 2001.

[12] T. Dokken and J. Thomassen. Overview of approximate implicitization. In *Topics in algebraic geometry and geometric modeling*, volume 334, pages 169–184. Amer. Math. Soc., Providence, RI, 2003.

[13] R. Feichtinger, M. Fuchs, B. Jüttler, O. Scherzer, and Huaiping Yang. Dual evolution of planar parametric spline curves and T–spline level sets. *Computer-Aided Design*, 40:13–24, 2008.

[14] R. Feichtinger, B. Jüttler, and H. Yang. Particle-based T-spline level set evolution for 3D object reconstruction with range and volume constraints. In S. Cunningham and V. Skala, editors, *Proc. WSCG*, pages 49–56. University of Plzen, Union Press, 2008.

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

[16] B. Jüttler and A. Felis. Least–squares fitting of algebraic spline surfaces. *Advances in Computational Mathematics*, 17:135–152, 2002.

[17] B. Jüttler and E. Wurm. Approximate implicitization via curve fitting. In L. Kobbelt, P. Schröder, and H. Hoppe, editors, *Symposium on Geometry Processing*, pages 240–247, New York, 2003. Eurographics/ACM Press.

[18] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, New York, 2003.

[19] V. Pratt. Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph.*, 21(4):145–152, 1987.

[20] G. Salmon. *A treatise on the analytic geometry of three dimensions*. Hodges, Figgis and Co., 4th ed. edition, 1882.

[21] T. W. Sederberg. *Implicit and parametric curves and surfaces for computer aided geometric design*. PhD thesis, Purdue University, West Lafayette, IN, USA, 1983.

[22] T.W. Sederberg and F. Chen. Implicitization using moving curves and surfaces. *Computer Graphics*, 29(Annual Conference Series):301–308, 1995.

[23] T.W. Sederberg and S.R. Parry. Comparison of three curve intersection algorithms. *Comput. Aided Des.*, 18(1):58–64, 1986.

[24] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(11):1115–1138, 1991.

[25] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(11):1115–1138, 1991.

[26] E. Wurm. *Approximate Techniques for the Implicitisation and Parameterisation of Surfaces*. PhD thesis, Johannes Kepler University, Linz, Austria, 2005.

[27] E. Wurm, J.B. Thomassen, B. Jüttler, and T. Dokken. Comparative benchmarking of methods for approximate implicitization. In M. Neamtu and M. Lucian, editors, *Geometric Modeling and Computing: Seattle 2003*, pages 537–548. Nashboro Press, Brentwood, 2004.

[28] H. Yang, M. Fuchs, B. Jüttler, and O. Scherzer. Evolution of T-spline level sets with distance field constraints for geometry reconstruction and image segmentation. In *Shape Modeling International*, pages 247–252. IEEE Press, 2006.

[29] H. Yang and B. J üttler. Evolution of T-spline level sets for meshing non–uniformly sampled and incomplete data. *The Visual Computer*, 24:435–448, 2008.