

Gauss-Newton-type Techniques for Robustly Fitting Implicitly Defined Curves and Surfaces to Unorganized Data Points

Martin Aigner *

Bert Jüttler†

Institute of Applied Geometry, Johannes Kepler University, Linz, Austria

ABSTRACT

We describe Gauss-Newton type methods for fitting implicitly defined curves and surfaces to given unorganized data points. The methods can deal with general error functions, such as approximations to the ℓ_1 or ℓ_∞ norm of the vector of residuals. Depending on the definition of the residuals, we distinguish between direct and data-based methods. In addition, we show that these methods can either be seen as (discrete) iterative methods, where an update of the unknown shape parameters is computed in each step, or as continuous evolution processes, that generate a time-dependent family of curves or surfaces, which converges towards the final result.

It is shown that the data-based methods – which are less costly, as they work without the need of computing the closest points – can efficiently deal with error functions that are adapted to noisy and uncertain data. In addition, we observe that the interpretation as evolution process allows to deal with the issues of regularization and with additional constraints.

1 INTRODUCTION

The next section summarizes the literature on Gauss–Newton–type techniques for parametric curve and surface fitting. We then consider implicit representations and techniques of robust fitting, before describing the content and the contributions of this paper.

1.1 Gauss–Newton–type techniques for parametric curve and surface fitting

Fitting a curve or surface to a given set of unorganized points (e.g., laser range data) is an important problem in various fields, including geometric modeling and computer vision. Many techniques have been developed for curves and surfaces which are described by parametric representations (NURBS curves and surfaces). Due to the influence of the parameterization, the fitting problem then leads to a non-linear optimization problem. Different approaches for dealing with the effects of this non-linearity have been developed.

[24] presents a general optimization-based approach to B-spline curve and surface fitting. The optimization of the parameterization has been studied in [26]. A simple alternating method, which switches between fitting and optimization of the parameterization (parameter correction) is described in [12]. [29] presents a Gauss–Newton methods for simultaneous optimization of an approximating spline curve and of its parameterization. Methods from the field of optimal control have been used in [7].

*e-mail: martin.aigner@jku.at

†e-mail: bert.juettler@jku.at

The method of squared distance minimization, which uses a modified quadratic approximation of the signed distance function of curves and surfaces, was presented in [21]. It has later been extended to scattered data approximation [20, 32] and it was compared with the Gauss–newton type technique of normal (or tangent) distance minimization of [9].

The choice of a good initial solution is of utmost importance for the success of the optimization. Geometrically motivated optimization strategies [20, 32] replace the initial solution by an initial curve or surface. It is tempting to view the intermediate results of an iterative (Gauss–Newton–type) optimization method as a time-dependent curve or surface which adapts itself to the unorganized point data [20, 32]. This is related to the concept of active curves and surfaces, which are used for image and video segmentation in Computer Vision [9].

In our previous work [5, 4] we considered geometric evolution processes driving a curve (and similarly a surface) towards given point cloud data, where the evolution is governed by a differential equation. [2] discusses this processes in a more abstract setting and shows that the result is independent of the choice of the particular geometric representation.

1.2 Fitting of implicitly defined curves and surfaces

Implicit representations of curves and surfaces [31] offer various advantages for curve and surface fitting. The most important ones are the non-existence of the parameterization problem, repairing capabilities of incomplete data and simple operations of shape editing.

Different representations of implicitly defined curves and surface have been used. Besides polynomials, these representations include discretized level sets [19, 40], scalar spline functions [23, 16] and scalar fields defined by radial basis functions [10, 18].

Several computational techniques for approximating unorganized point cloud data by implicitly defined curves and surfaces exist. The fitting of conic sections to scattered data is discussed in [25]. [22] uses a simple linear normalization in order to avoid the trivial solution (the null scalar field). In [30], a data-dependent normalization is introduced, which gives results that are independent of the choice of the coordinate system. The fitting is formulated as a non-linear least-squares problem that leads to a generalized eigenvalue problem.

Discretized Level set evolution for surface reconstruction is presented in [40, 39]. In [28] the reconstruction from implicit surfaces from given polygonal models is considered.

Different techniques for the case of orientable point cloud data, where each point can be equipped with a normal vector, are described in [15, 10, 16]. These techniques use off-surface points or the simultaneous approximation of points and normals in order to avoid the trivial solution.

[38] have used active implicit B-spline curves for fitting unorganized point clouds, by extending the geometric distance minimization in the case of implicitly defined curves using a trust-region algorithm.

The evolution of T-spline level sets is proposed in [35, 36]. The use of T-splines, which were presented on [27], makes it possible to adapt the distribution of the degrees of freedom to the given geometric data. The first paper proposes to use a distance field constraint that completely avoids the use of frequent re-initialization steps. A technique that is capable of dealing with various constraints such as convexity, area/volume and range constraints are described in the forthcoming publication [11].

1.3 Robust fitting – other norms

Most fitting techniques for curves and surfaces are based on variants of least-squares approximation, i.e., they consider the ℓ_2 norm of the vector of residuals (which may be defined in different ways, depending in the representation). However, this may not be the optimal approach in many situations, as one tacitly assumes a Gaussian distribution of the error that may be present in the data.

Often, this implicit assumption concerning the error is not justified. In the presence of outliers (data with large error), these may destroy the quality of the approximation, since their influence grows quadratically with the distance to the curve or surface. If the data are very precise, then it is more appropriate to minimize the maximum deviation between the model and the data. Obviously, it is important to adjust the norm carefully to the problem.

As an alternative, one may use other norms, such as ℓ_p norms of the vector of residuals or approximations thereof. So far, this has been done mostly for curves and surfaces that are defined by parametric representations. [33] describes two methods for discrete ℓ_p approximation. [14] uses linear programming for approximate ℓ_1 and approximate ℓ_∞ fitting of parametric curves.

The Gauss–Newton method for ℓ_1 orthogonal distance regression is studied in [34]. [6] use the ℓ_1 and ℓ_∞ for fitting parametric curves and surfaces. The case of general ℓ_p norms is described in [8]. In a recent manuscript [3], we study the relation between Gauss–Newton–type methods for approximation with respect to general norm–like functions and the technique of iteratively re-weighted least squares, which is a classical tool in the field of robust statistics [13], by extending the observations in [17] to the case of vector-valued residuals.

1.4 Contributions and outline

In the present paper we describe Gauss–Newton type methods for fitting implicitly defined curves and surfaces to given unorganized data points. Here we extend the results concerning parametric curves and surfaces to the implicit case. As another new contribution, the methods can deal with general error functions, such as approximations to the ℓ_1 or ℓ_∞ norm of the vector of residuals.

We consider two different possible definitions of the residuals. This leads to two classes of methods, which will be called *direct methods* and *data-based methods*, respectively.

Both methods can be seen as (discrete) iterative methods, where an update of the unknown shape parameters is computed in each step. Alternatively they can be identified with continuous evolution processes which generate a time-dependent family of curves or surfaces. We show that both viewpoints are equivalent.

The data-based methods are less costly, as they work without the need of computing the closest points. They can efficiently deal with error functions that are adapted to noisy and uncertain data. The

interpretation as evolution process allows to deal with the issues of regularization and with additional constraints.

The remainder of the paper is organized as follows. In the second section we recall some basics about least-squares fitting for implicit curves. We mention exemplarily three existing methods for curve fitting that base on a least-squares approximation. In the third section we generalize the least-squares technique and provide some theoretical results. Section 4 is dedicated to the discussion of the method and presents several examples. Finally, in Section 5, we conclude this paper.

2 FITTING BY IMPLICITLY DEFINED CURVES AND SURFACES

First we recall the description of implicitly defined curves and surfaces. Then we formulate two approaches to the fitting problem: the *direct distance minimization* and the *algebraic distance minimization*. Finally we summarize the evolution-based approach to fitting, where the result is obtained via an evolution process.

2.1 Implicitly defined curves and surfaces

Consider the problem of fitting an implicitly defined planar curve or surface to a given point set $\{\mathbf{p}_j\}_{j=1..M} \subset \mathbb{R}^d$, where the dimension d satisfies $d = 2$ in the curve case and $d = 3$ in the curve of surfaces. More precisely, we consider hypersurfaces in dimension two or three. For the sake of brevity, we will almost always refer simply to surfaces.

The curve/surface is described as the zero-set of a \mathcal{C}^2 smooth bivariate or trivariate function $f_s : \mathbb{R}^d \rightarrow \mathbb{R}$, i.e.

$$\mathcal{F} = \{\mathbf{x} \in \Omega \subset \mathbb{R}^d \mid f_s(\mathbf{x}) = 0\}. \quad (1)$$

We assume that the scalar field f_s is described by a vector

$$\mathbf{s} = (s_1, \dots, s_m)^\top \quad (2)$$

of m shape parameters, where $\mathbf{s} \in \mathbb{R}^m$. We assume that f_s is twice differentiable with respect to the shape parameters \mathbf{s} . The gradients with respect to \mathbf{x} and \mathbf{s} will be denoted by

$$\nabla_{\mathbf{x}} = \left(\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_d} \right) \quad (3)$$

and

$$\nabla_{\mathbf{s}} = \left(\frac{\partial}{\partial s_1}, \dots, \frac{\partial}{\partial s_m} \right), \quad (4)$$

respectively.

Example 1 For instance, the function $f_s(\mathbf{x})$ can be chosen as a bivariate polynomial of a given degree n . In the case $d = 2$, one may e.g. choose the bivariate Bernstein–Bézier basis with respect to a suitable domain triangle $\Delta ABC \subset \mathbb{R}^2$,

$$f_s(\mathbf{x}) = \sum_{i+j+k=n} b_{ijk} \frac{n!}{i!j!k!} u^i v^j w^k, \quad (5)$$

where (u, v, w) are the barycentric coordinates of the point $\mathbf{x} \in \mathbb{R}^2$ with respect to the domain triangle. In this case, the vector of shape parameters is simply the collection of all coefficients b_{ijk} .

Example 2 More generally, one may consider implicitly defined surfaces where

$$f_s(\mathbf{x}) = \sum_{i=0}^N s_i \phi_i(\mathbf{x}), \quad (6)$$

where the basis functions ϕ_i can be chosen, e.g., as radial basis functions [10], tensor-product splines or subsets thereof [16] or T-splines [37].

2.2 Direct distance minimization

When fitting a surface to a given set of data, one minimizes certain distances from the data points to the surface. These distances can be measured in different ways.

The geometric distance from a data point \mathbf{p}_j to the surface is given by its (minimal) Euclidean distance to the surface,

$$d_j = \min \|\mathbf{p}_j - \mathbf{p}_j^c\| \quad (7)$$

$$\text{subject to } f_s(\mathbf{p}_j^c) = 0. \quad (8)$$

The minimization of the squares of the geometric distances $d_j = \|\mathbf{R}_j\|$, where

$$\mathbf{R}_j = \mathbf{p}_j - \mathbf{p}_j^c, \quad (9)$$

leads to the least-squares problem

$$\sum_{j=1}^M \mathbf{R}_j^2 = \sum_{j=1}^M d_j^2 \rightarrow \min_{\mathbf{s}} \quad (10)$$

Due to the non-linearity, an iterative approach is used to solve this problem. Starting with an initial surface, one computes – for each data point – the associated closest points. Substituting these points in (10) one obtains a least-squares problem for the shape parameters \mathbf{s} that can be solved using Newton or Newton-like techniques.

Note that the solution of (10) is not unique since all functions $\lambda f(x, y)$ with $\lambda \in \mathbb{R} \setminus \{0\}$ possess the same zero contour (1). In order to avoid this ambiguity, suitable regularization terms have to be incorporated.

For instance, if additional normal vector information is available (that is, if the given data are orientable in the sense that each point can be equipped with a normal vector), then one may use this information to define additional terms in the objective function that measure the deviation between the given normals and the real ones, see [16]. Otherwise one may use data-dependent regularization functionals as in [30], or one may simply constrain the values of some coefficients, such as the constant term [22]. In addition, in order to avoid unwanted oscillations, one may add regularization terms which are based on the thin plate energy, see [16].

2.3 Algebraic distance minimization

A point \mathbf{x} lies on an implicitly defined surface if its algebraic distance to the surface is zero, i.e., $f(\mathbf{x}) = 0$. For points in the vicinity of the surface the function value is also expected to be close to zero. This gives rise to the idea of using

$$\sum_{j=1}^M |f_s(\mathbf{p}_j)|^2 \rightarrow \min_{\mathbf{s}}, \quad (11)$$

which minimizes the squared algebraic distances to all data points.

Clearly, this problem has the trivial solution $f_s(\mathbf{x}) \equiv 0$ which is not useful. Again, additional constraints have to be introduced in order to exclude this unwanted solution, see the discussion in the previous section and Section 4.1.

2.4 Surface fitting via an evolution process

In the case of curves, [5] proposed a geometrically motivated approach to the fitting problem, which can be extended to surfaces. The approximation process is interpreted as a continuous evolution process that drives an initial surface towards the target, which is specified by the data points. In particular, the closest points of the data points on the curve are attracted by the associated data points.

Assuming that the shape parameters \mathbf{s} depend on a time-like parameter t , the normal velocity of a point on the surface is

$$\mathbf{v}^n = -\frac{\nabla_{\mathbf{x}} f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|^2} \dot{f}_s = -\frac{\nabla_{\mathbf{x}} f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|^2} \nabla_{\mathbf{s}} f_s \dot{\mathbf{s}}, \quad (12)$$

where $\dot{\mathbf{s}}$ is the time derivative of \mathbf{s} . The velocity of the closest points in normal direction shall be equal to their distance to the data point. As this requirement cannot be fulfilled for all points simultaneously, we formulate this as the least squares problem

$$\sum_{j=1}^M ((\mathbf{R}_j - \mathbf{v}^n) \mathbf{n}_j)^2 \rightarrow \min_{\dot{\mathbf{s}}}. \quad (13)$$

The shape parameters $\mathbf{s}^+ = \mathbf{s}(t+h)$ for the updated surface can now be found via an explicit Euler step $\mathbf{s}^+ = \mathbf{s} + h\dot{\mathbf{s}}$ where $0 < h \leq 1$ is a given step size.

Again, in order to avoid a convergence to the trivial solution $f_s \equiv 0$, suitable regularization terms have to be added. See Section 4.1 for a detailed discussion.

2.5 Computation of closest points

Note that for both, the direct distance minimization and the evolution approach, the closest point of each data point has to be computed. See [1] for a robust method for closest point computation. Although the computation of a single closest point requires only the solution of a non-linear system of two or three unknowns, the overall effort may be quite substantial.

First, the systems are in general non-linear and can not be solved directly. Second, the number of data points can be quite high, and the closest points need to be computed in each step of the iterative method and in each time-step of the evolution process.

Consequently, it is desirable to avoid the computation of closest points, and this will be achieved by the use of point-based approximation techniques.

3 EXTENSION TO GENERALIZATION NORM-LIKE FUNCTIONS

We generalize the direct and the algebraic distance minimization from the last section to the case of general norm-like functions of the residuals. These general functions include approximations of the ℓ_1 and the ℓ_∞ norm of the vector of residuals. They are particularly well suited for data containing outliers, and for highly accurate data, respectively.

3.1 Generalized direct distance minimization

3.1.1 The generalized fitting problem and norm-like functions

We extend the exact geometric distance minimization by considering the objective function

$$F = \sum_{j=1}^M N(\|\mathbf{p}_j - \mathbf{p}_j^c\|) \rightarrow \min \quad (14)$$

$$\text{subject to } f_s(\mathbf{p}_j^c) = 0. \quad (15)$$

The function $N(x)$ replaces the usual ℓ_2 norm. It is assumed to satisfy the following definition of a norm-like function (cf. [3]):

Definition 3 A \mathcal{C}^2 function $N(x) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is said to be **norm-like** if there exists $\varepsilon \in \mathbb{R}^+$ such that the derivative satisfies

$$N'(x) = xw(x) \quad \text{for } x \in (0, \varepsilon] \quad (16)$$

where the associated **weight function** $w(x)$ is positive. If the weight function $w(x)$ can smoothly be extended such that $w(x) : [0, \varepsilon] \rightarrow [c, C]$ with $c, C \in \mathbb{R}^+$, then we will call it **positive and bounded**.

The norm-like functions

$$N(x) = x^p \quad (17)$$

are associated with the weight functions

$$w(x) = px^{p-2}. \quad (18)$$

For $1 < p < 2$, the weight functions are positive, but not bounded. In the case $p > 2$ the weight functions are bounded but not positive. For $p = 2$ the weight function is both, positive and bounded.

The norm-like function

$$N(x) = 1 - \exp(-x^2) \quad (19)$$

has the weight function

$$w(x) = 2\exp(-x^2) \quad (20)$$

and can be used as an alternative to the ℓ_1 norm. The weight function is positive and bounded.

3.1.2 A Gauss–Newton–type method

In order to cope with the non-linearity of (14), we apply an iterative technique, more precisely a Gauss-Newton-type method. Consequently, we need the gradient and the Hessian of the objective function. First we consider the gradient

$$\begin{aligned} \nabla F &= \sum_{j=1}^M N'(\|\mathbf{R}_j\|) \frac{\mathbf{R}_j^\top}{\|\mathbf{R}_j\|} \nabla_s \mathbf{R}_j \\ &= \sum_{j=1}^M w(\|\mathbf{R}_j\|) \|\mathbf{R}_j\| \frac{\mathbf{R}_j^\top}{\|\mathbf{R}_j\|} \nabla_s (-\mathbf{p}_j^c). \end{aligned}$$

The gradient $\nabla_s(-\mathbf{p}_j^c)$ can be computed from the condition $f_s(\mathbf{p}_j^c) \equiv 0$, which implies

$$\nabla_s f_s(\mathbf{p}_j^c) + \nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c) \nabla_s \mathbf{p}_j^c = 0 \quad (21)$$

and hence

$$\frac{\nabla_s f_s(\mathbf{p}_j^c)}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)\|} + \frac{\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)\|} \nabla_s \mathbf{p}_j^c = 0 \quad (22)$$

Note that for a regular surface point, i.e. $\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c) \neq 0$, the residual \mathbf{R}_j is parallel to the gradient $\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)$. Consequently, we have that

$$\frac{\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)\|} \text{sign}(\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c) \mathbf{R}_j) = \frac{\mathbf{R}_j^\top}{\|\mathbf{R}_j\|}. \quad (23)$$

Combining (22) and (23) yields

$$\frac{\mathbf{R}_j^\top}{\|\mathbf{R}_j\|} \nabla_s(-\mathbf{p}_j^c) = \frac{\nabla_s f_s(\mathbf{p}_j^c)}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)\|} \text{sign}(\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c) \mathbf{R}_j). \quad (24)$$

Summing up, the gradient of the objective function is

$$\nabla F = \sum_{j=1}^M w(\|\mathbf{R}_j\|) \|\mathbf{R}_j\| \frac{\nabla_s f_s(\mathbf{p}_j^c)}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c)\|} \text{sign}(\nabla_{\mathbf{x}} f_s(\mathbf{p}_j^c) \mathbf{R}_j) \quad (25)$$

Note that the function f_s and its gradient are evaluated at the point \mathbf{p}_j^c . We keep this fact in mind and suppress the arguments from now on. In particular, we use the abbreviation

$$w_j = w(\|\mathbf{R}_j\|). \quad (26)$$

In this sense the Hessian is

$$\begin{aligned} H_F &= \nabla(\nabla F^\top) \\ &= \sum_{j=1}^M w'_j \|\mathbf{R}_j\| \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \\ &\quad + w_j \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \\ &\quad + w_j \|\mathbf{R}_j\| \left(\frac{\nabla_s \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} - \frac{\nabla_s f_s^\top \nabla_{\mathbf{x}} f_s \nabla_s \nabla_{\mathbf{x}} f_s}{\|\nabla_{\mathbf{x}} f_s\|^3} \right) \end{aligned} \quad (27)$$

The computation of the exact Hessian (27) might be very costly or even impossible, if no second order derivatives are available. For this reason – adopting the paradigm of Gauss–Newton methods – we consider a simplified Hessian, where the first and the last part of the expansion (27) are omitted,

$$H_F^* = \sum_{j=1}^M w_j \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|}. \quad (28)$$

The following result justifies the choice of this simplification.

Proposition 4 Let $\bar{\mathbf{s}}$ be a minimizer of (14) such that $f_s(\mathbf{p}_j) = 0$ and $\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j)\| > 0$ for all data points \mathbf{p}_j . Assume that $f_s(\mathbf{x}) \in \mathcal{C}^2$ is bounded for all $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and that w is a positive and bounded weight function. Then

$$\lim_{\mathbf{s} \rightarrow \bar{\mathbf{s}}} H_F = H_F^*.$$

Proof: First we note that the boundedness of $f_s(\mathbf{x})$ implies that the derivatives of $f_s(\mathbf{x})$ of all orders are bounded as well. Now we consider the three terms in (27) separately. For the first term we obtain

$$T_1 \leq |w'_j| \|\mathbf{R}_j\| \left\| \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \right\| \leq |w'_j| \|\mathbf{R}_j\| \frac{\|\nabla_s f_s\|^2}{\|\nabla_{\mathbf{x}} f_s\|^2}.$$

Consequently, T_1 vanishes in the limit as $\|\mathbf{R}_j\| \rightarrow 0$ for $\mathbf{s} \rightarrow \bar{\mathbf{s}}$ since w' is bounded and $\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j)\| > 0$. The third part can be bounded by

$$T_3 \leq |w_j| \|\mathbf{R}_j\| \left(\frac{\|\nabla_s \nabla_s f_s\|}{\|\nabla_{\mathbf{x}} f_s\|} + \|\nabla_s f_s^\top\| \frac{\|\nabla_{\mathbf{x}} f_s\|}{\|\nabla_{\mathbf{x}} f_s\|^3} \|\nabla_s \nabla_{\mathbf{x}} f_s\| \right).$$

Again, this part vanishes due to the boundedness of all involved derivatives and $\|\nabla_{\mathbf{x}} f_s\| > 0$. \square

Summing up, the simplified and the exact Hessian coincide in the limit of a zero-residual case.

Now we use the simplified Hessian in order to build the system

$$H_F^* \Delta \mathbf{s} + \nabla F^\top = 0 \quad (29)$$

which leads to

$$\begin{aligned} \sum_{j=1}^M w_j \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \Delta \mathbf{s} \\ + \sum_{j=1}^M w_j \|\mathbf{R}_j\| \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \text{sign}(\nabla_{\mathbf{x}} f_s \mathbf{R}_j) = 0. \end{aligned} \quad (30)$$

3.1.3 Relation to surface evolution

As all functions and gradients are evaluated at points of the surface, we will refer to (30) as the *surface-based Gauss-Newton-type method* for minimizing general functions of the residuals in the case of implicitly defined surfaces.

Proposition 5 The surface-based Gauss-Newton-type method in the sense of (30) gives exactly the same system of equations as a weighted version of the surface evolution (13).

Proof. The system (30) can be interpreted as the minimum condition of

$$\sum_{j=1}^M w_j \left(\left(\mathbf{R}_j + \frac{\nabla_{\mathbf{x}} f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \Delta \mathbf{s} \right)^\top \frac{\nabla_{\mathbf{x}} f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \right)^2 \rightarrow \min_{\Delta \mathbf{s}}.$$

With the setting

$$\mathbf{v}^n = - \frac{\nabla_{\mathbf{x}} f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \dot{\mathbf{s}}$$

and

$$\mathbf{n}_j = \frac{\nabla_{\mathbf{x}} f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|}$$

we get

$$\sum_{j=1}^M w_j ((\mathbf{R}_j - \mathbf{v}^n)^\top \mathbf{n}_j)^2 \rightarrow \min_{\Delta \mathbf{s}}, \quad (31)$$

which is the weighted version of equation (13) that governs the surface evolution. \square

Note that the weights in (31) depend on the residuals. Hence (31) cannot be minimized directly. Most often, such problems are solved via re-weighting, i.e. during the iteration process the weights are computed from the residuals of the previous step. Thus, replacing the usual ℓ_2 norm by a general norm like function $N(x)$ increases the computational costs only marginally. The additional effort lies solely in the computation of the additional weights that have to be incorporated.

In the statistics community, this weighting technique is known as Iteratively-Reweighted Least-Squares (IRLS), see [13, 17]. Also [30] has used this (iterative) re-weighting procedure in order to refine the result obtained from the (direct) eigenvector fit applied to the algebraic distances.

3.2 Minimizing an approximate distance

3.2.1 The generalized fitting problem for Sampson distances

As pointed out in Section 2.5, the closest point computation for methods relying on the exact distance can be quite costly. This is particularly true as it may be needed many times during the evolution / iteration process.

For this reason we consider an approximation instead of the exact geometric distance. In the following we discuss the use of the Sampson distance [25], which is also known as Taubin distance [30]. It is the gradient-weighted algebraic error,

$$d_S = \frac{|f_s(\mathbf{p}_j)|}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j)\|}. \quad (32)$$

With this, the objective function for exact distance minimization transforms into

$$F = \sum_{j=1}^M N\left(\frac{|f_s(\mathbf{p}_j)|}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j)\|}\right) \rightarrow \min. \quad (33)$$

In contrast to the direct distance minimization, the gradients and functions are now evaluated at the data points.

3.2.2 A Gauss-Newton-type method

Again we skip the arguments in the following discussion. The gradient of (33) is

$$\nabla F = \sum_{j=1}^M N'\left(\frac{|f_s|}{\|\nabla_{\mathbf{x}} f_s\|}\right) \left(\frac{\text{sign}(f_s) \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} - \frac{|f_s| \nabla_{\mathbf{x}} f_s \nabla_s \nabla_{\mathbf{x}} f_s}{\|\nabla_{\mathbf{x}} f_s\|^3} \right) \quad (34)$$

The second term vanishes for zero-residual problems. Omitting it yields an approximate gradient,

$$\nabla F^* = \sum_{j=1}^M w_j \left(\frac{|f_s|}{\|\nabla_{\mathbf{x}} f_s\|} \right) \frac{f_s \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|^2}. \quad (35)$$

The Hessian of (33) is

$$\begin{aligned} H_F &= \nabla(\nabla F^{*\top}) = \\ &= \sum_{j=1}^M w_j' \frac{f_s \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|^2} \left(\frac{\text{sign}(f_s) \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} - \frac{|f_s| \nabla_{\mathbf{x}} f_s \nabla_s \nabla_{\mathbf{x}} f_s}{\|\nabla_{\mathbf{x}} f_s\|^3} \right) \\ &+ w_j \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \\ &+ w_j f_s \left(\frac{\nabla_s \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|^2} - \frac{2 \nabla_s f_s \nabla_s \nabla_{\mathbf{x}} f_s}{\|\nabla_{\mathbf{x}} f_s\|^4} \right). \end{aligned}$$

Again, we consider a simplified version of the exact Hessian:

$$H_F^* = \sum_{j=1}^M w_j \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|}$$

Proposition 6 Let $\bar{\mathbf{s}}$ be a minimizer of (33) such that $f_s(\mathbf{p}_j) = 0$ and $\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j)\| > 0$. Assume that $f_s(\mathbf{x})$ is bounded for all $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and that w is a positive and bounded weight function. Then

$$\lim_{\mathbf{s} \rightarrow \bar{\mathbf{s}}} H_F = H_F^*.$$

The proof is similar to the one of Proposition 4.

Now we can formulate the system for the update vector $\Delta \mathbf{s}$:

$$H_F^* \Delta \mathbf{s} + \nabla F^{*\top} = 0$$

which leads to

$$\sum_{j=1}^M w_j \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \Delta \mathbf{s} + \sum_{j=1}^M w_j \frac{f_s \nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|^2} = 0 \quad (36)$$

Note that equation (36) has the same shape as (30). Nevertheless, the difference is significant. No closest point computation is needed since all functions and derivatives are evaluated at the data points. Therefore we call (36) *data-based Gauss-Newton-type method*.

Figure 1 shows a first example of a data-based Gauss-Newton-type method for ℓ_2 approximation. We will see shortly that it is equivalent to an evolution process.

3.2.3 Relation to surface evolution

Now we can proceed similar to Proposition 5 and obtain the following result.

Proposition 7 The data-based Gauss-Newton-type method (36) is equivalent to a weighted evolution, where the error distances are measured with the Sampson distance and the velocities are applied at the data points.

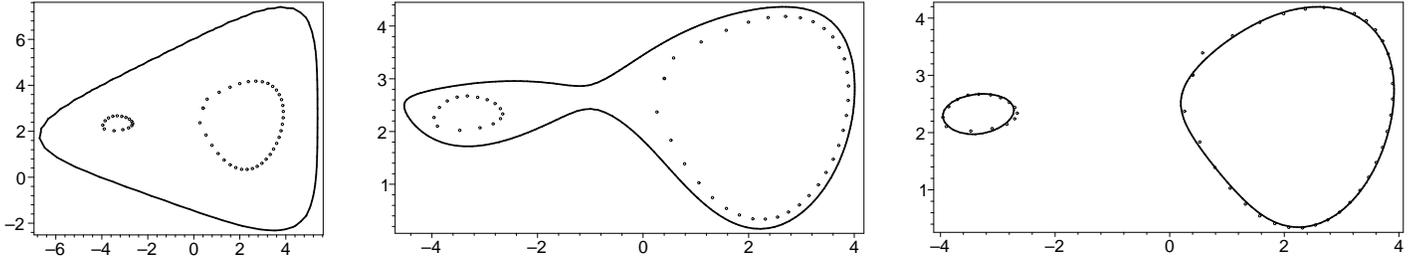


Figure 1: Initial value (left), an intermediate step (center) and the final result (right) of a data-based Gauss-Newton-type method (or equivalently: of an evolution process) for ℓ_2 -approximation.

Proof. Formulating a weighted evolution according to (13), where the residuals are computed via the Sampson distance, yields

$$\sum_{j=1}^M w_j \left(\frac{\nabla_{\mathbf{x}} f_s f_s}{\|\nabla_{\mathbf{x}} f_s\|^2} + \frac{\nabla_{\mathbf{x}} f_s \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|^2} \dot{s} \frac{\nabla_{\mathbf{x}} f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \right)^2 \rightarrow \min_s. \quad (37)$$

The necessary condition for a stationary point of this minimization problem is now given by (36) where we rename Δs into \dot{s} . \square

4 DISCUSSION AND EXAMPLES

We compare direct methods and iterative (evolution-based) methods and describe the use of regularization by the distance field constraint. Next we analyze the differences between surface-based and data-based methods. Finally we address the use of general norm-like functions.

4.1 Direct methods vs. iterative methods and evolution, regularization by distance field constraint

Many direct techniques, such as Taubin's method [30], provide the result after a single step (though an iterative method may be needed in order to solve the generalized eigenvalue problem). On the one hand this seems to be an advantage, since no iterations and therefore no convergence analysis is needed. On the other hand, once the result is computed it can no longer be influenced. Iterative methods provide the flexibility to adjust certain parameters – such as regularization parameters – during the approximation.

An example – which compares Taubin's method and an ℓ_2 approximation by evolution – is shown in Figure 2. It can be seen that the latter technique gives the correct result, while Taubin's fit has some problems.

The interpretation of an iteration as discrete steps of a continuous evolution offers even more flexibility. This is especially true for the choice of the regularization parameters.

We use the technique of *distance field constraint*, which was proposed in [37]. As the underlying idea, in each step, the level set function is to be pushed towards a signed distance function. This can be achieved by adding the term

$$\int_{\Omega} \left(\frac{d}{dt} \|\nabla_{\mathbf{x}} f_s(\mathbf{x})\| + \|\nabla_{\mathbf{x}} f_s(\mathbf{x})\| - 1 \right)^2 d\mathbf{x} \rightarrow \min \quad (38)$$

to the objective function.

The interpretation of this regularization term is as follows. If the norm of the gradient in a point equals 1 then its time derivative is zero, hence it shall remain unchanged. Otherwise the norm of the gradient is modified such that it gets closer to 1. Clearly this condition avoids the zero solution for the level set function.

Since the integration over the domain of interest might be complicated, we adopt the following approach. We discretize (38) by applying it to a number of points \mathbf{x}_i in the domain of interest. This leads to

$$\sum_i \left(\frac{d}{dt} \|\nabla_{\mathbf{x}} f_s(\mathbf{x}_i)\| + \|\nabla_{\mathbf{x}} f_s(\mathbf{x}_i)\| - 1 \right)^2 \rightarrow \min. \quad (39)$$

Theoretically, the distance field constraint can be imposed to a very dense grid of points in the domain of interest. However, a unit gradient field exists only in some neighborhood of the zero contour, bounded by the evolute of the curve or by the focal surfaces of the surface.

Since this neighborhood is unknown we use the following strategy. We sample a number of points on a regular grid and choose those points that are close to the curve or surface. In order to avoid the time consuming distance computation we allow all points that have a small Sampson distance or alternatively, a small absolute function value.

4.2 Surface-based evolution vs. data-based iteration or evolution

As shown earlier, the Gauss-Newton approach to exact distance minimization led to the surface-based evolution process. On the other hand, the approximate distance minimization (based on the Sampson distance) can be interpreted as a data-based evolution process. Figure 3 shows a comparison of the two evolution methods.

In the first (surface-based) approach, the velocities (dotted arrows) are directly prescribed at the closest points which lie on the surface. Thus the closest points – and consequently the curve / surface $f = 0$ – is pushed towards the data points. The magnitude of the movement is derived from the Euclidean distance from a data point to its associated closest point.

Using the data-based approach, the velocities are applied to the data points. The geometric interpretation is now slightly different. We do not move the surface directly, but instead the forces are applied to a certain level set surface $f = c$ that passes through the data point. The magnitude of the velocity is given by the Sampson distance from the data point to the surface. In the sketch these velocities are represented by the dashed arrows.

As an obvious difference between the data-based and the surface-based evolution, no closest points are needed for the first technique. When considering the computational costs, this is a powerful argument to favor the approximate distance minimization. Moreover, this method can handle a specific class of topology changes more easily as demonstrated in the next example which is shown in Figure 4.

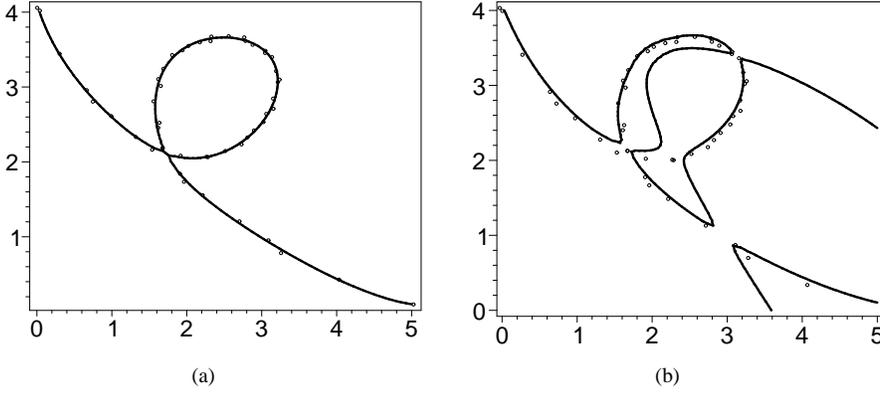


Figure 2: The figures show 50 points which were sampled from a parametric curve and perturbed with randomly generated error. The left picture (a) shows an ℓ_2 fit obtained with the data-based evolution technique combined with the regularization technique of distance field constraint, see text for details. In the second picture (b), the same point cloud is approximated with Taubin's method.

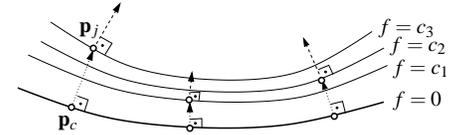


Figure 3: Comparison of data-based and surface-based evolution. Velocities at data points are shown as dashed arrows. Velocity at surface points are shown as dotted arrows.

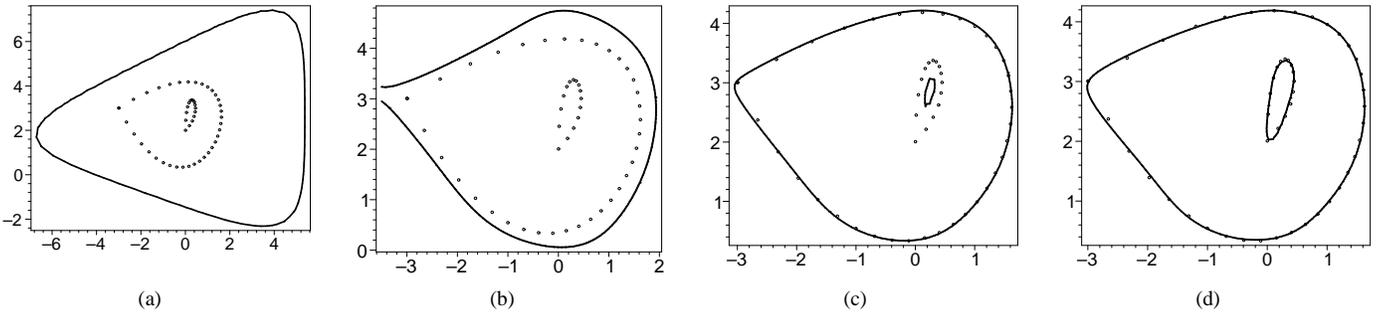


Figure 4: We consider a point cloud that consists of two nested loops. As an initial value we choose a closed curve that contains all data points. During the evolution process for ℓ_2 approximation of the Sampson distances, the curve shrinks until it matches the outer part of the points (b). Simultaneously, a second branch of the curve emerges inside the outer loop, which fits itself to the second part of the point cloud (c).

On the other hand, the surface-based technique acts directly on the true geometric distance errors, and not on an approximation. Consequently, one may expect that the results are more reliable. The choice of the most appropriate method depends on the application background.

Following our experiences, the data-based method is able to cope with such situations as in Figure 4 better than the surface-based approach. Using the latter method, the initial surface converges in the beginning towards the outer loop as expected. But then it does not stop at the boundary but is pulled inside the point cloud as it is attracted by the inner points too.

The data-based method seems to be more appropriate to deal with this kind of topology changes. However, this flexibility causes also problems, since unwanted branches may appear during the evolution.

Summing up, for simple shapes one may choose the surface-based evolution since it is not so vulnerable to unwanted topology changes. If one needs more flexibility in order to fit complicated shapes, the data-based evolution is more appropriate. But one has to use more rigorous regularization techniques in order to guarantee a stable evolution.

The application of the data-based technique to spatial data is shown in Figure 6. A point cloud sampled from two spheres was

approximated using the ℓ_2 norm.

4.3 Robust fitting via general norm-like functions

Finally we demonstrate the advantages of using general norm-like functions.

Figure 5 compares the results of data-based ℓ_2 and ℓ_1 approximation, where one of the data has been moved away, thereby creating an outlier.

The approximate ℓ_1 approximation (which uses the norm like function (19)) simply ignores the outlier, while the result of the ℓ_2 approximation has additional near-singular points. This confirms the theoretically expected higher robustness of ℓ_1 approximation with respect to outliers.

Finally we present a comparison of an approximate ℓ_1 , an ℓ_2 , and an approximate ℓ_∞ fit, see Figure 7. While the first one uses again the weight function which is obtained from (19), the last one uses ℓ_p approximation for a relatively large value of p .

Starting from the initial position shown in the top left corner, we obtain three different approximations. The figure also shows the residual vectors.

The norm of the residuals is shown in the three plots on the right-hand side. One may clearly see that the approximate ℓ_∞ approximation produces the smallest maximum distance error (0.4 vs. 0.52

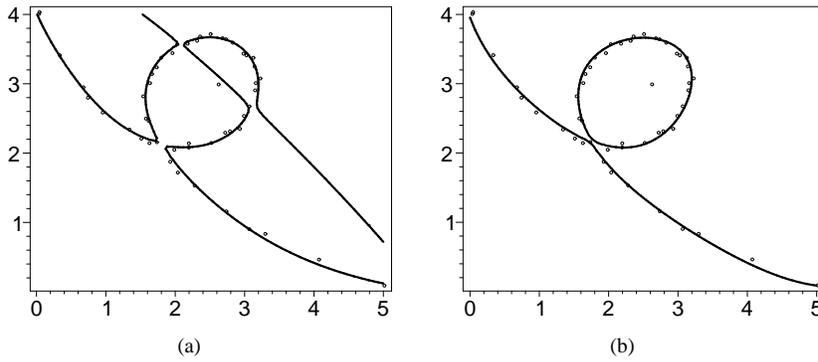


Figure 5: Approximation of a data set with an outlier, via data-based ℓ_2 evolution (a) and via data-based approximate ℓ_1 evolution (b).

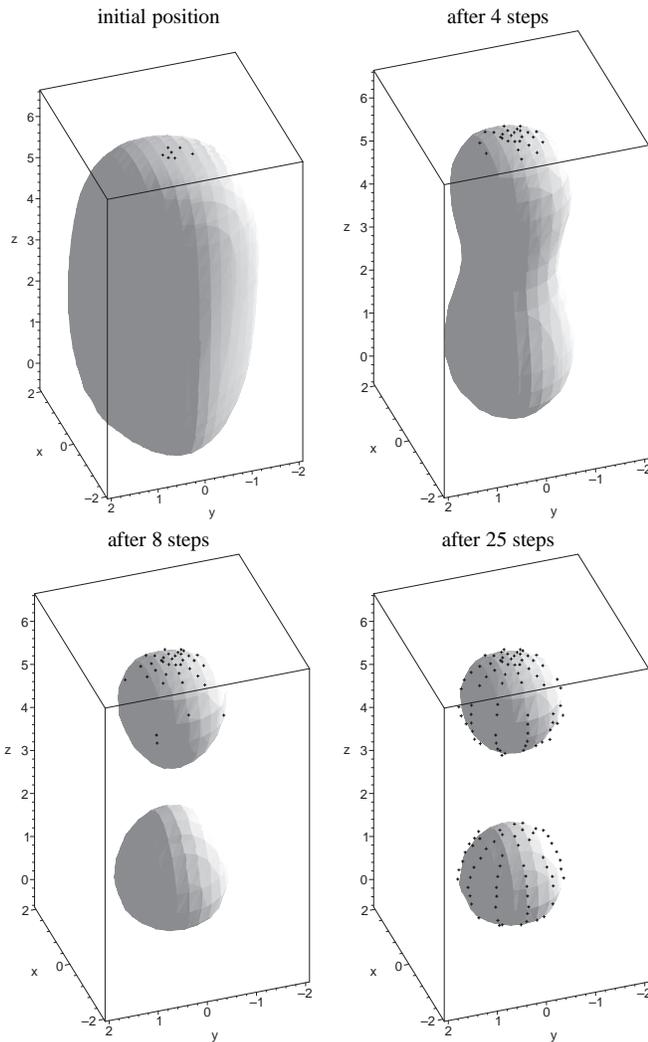


Figure 6: Step wise ℓ_2 approximation of 280 points sampled from two spheres with additional random error of 0.1.

and 0.62 for the ℓ_2 and the approximate ℓ_1 approximation).

On the other hand, one of the data points can again be seen as an outlier, and the approximate ℓ_1 ignores this point. More precisely, if the sum of the residuals can be decreased at the expense of one of them, then the ℓ_1 approximation will do it.

Figure 8 shows the same effect for spatial data, sampled from a sphere. Again one can see the approximations obtained with different norm like functions along with the different residuals of the 141 data points.

Summing up by using the various approximation methods, different assumptions or experiences concerning the error distribution can be taken into account.

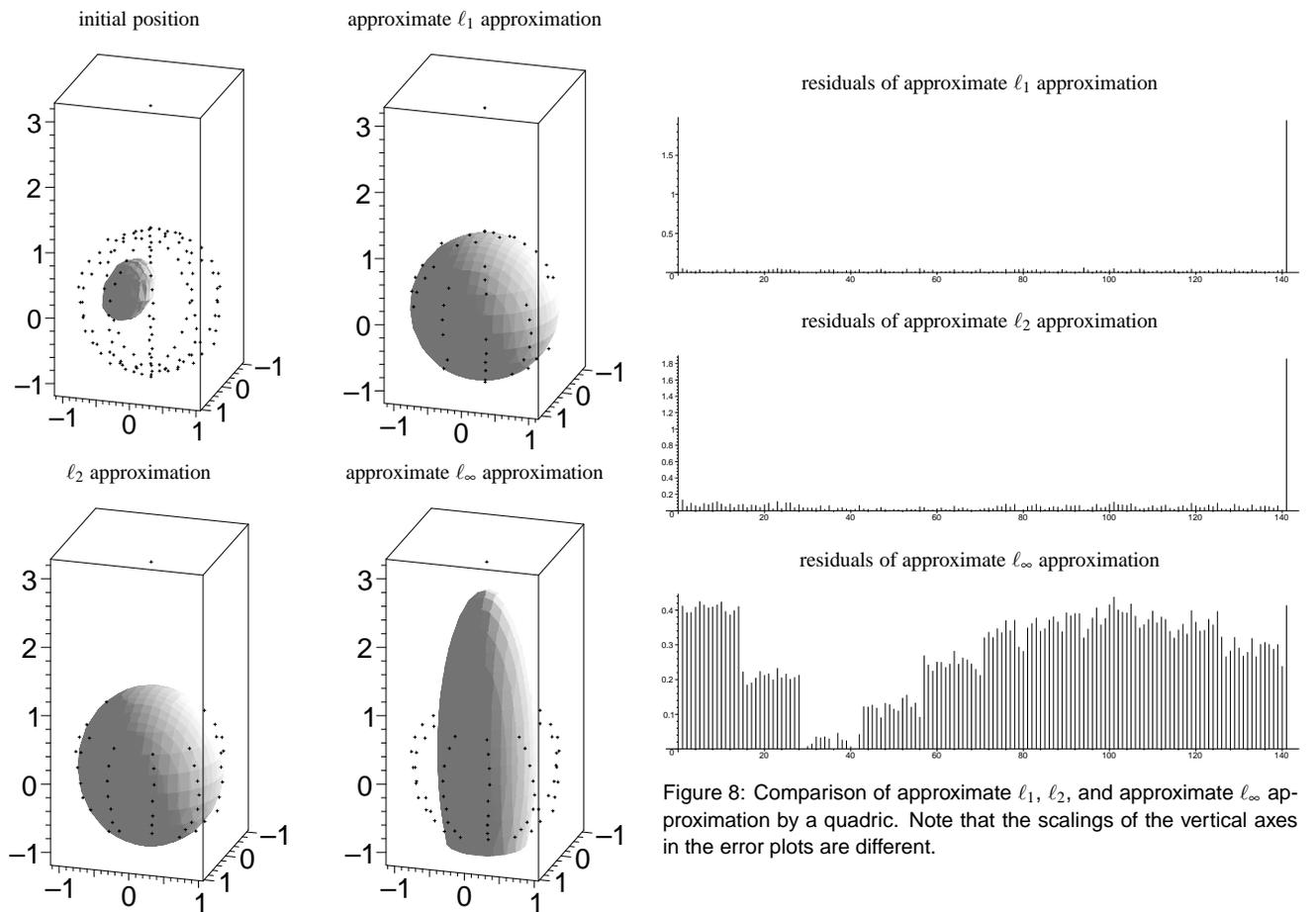
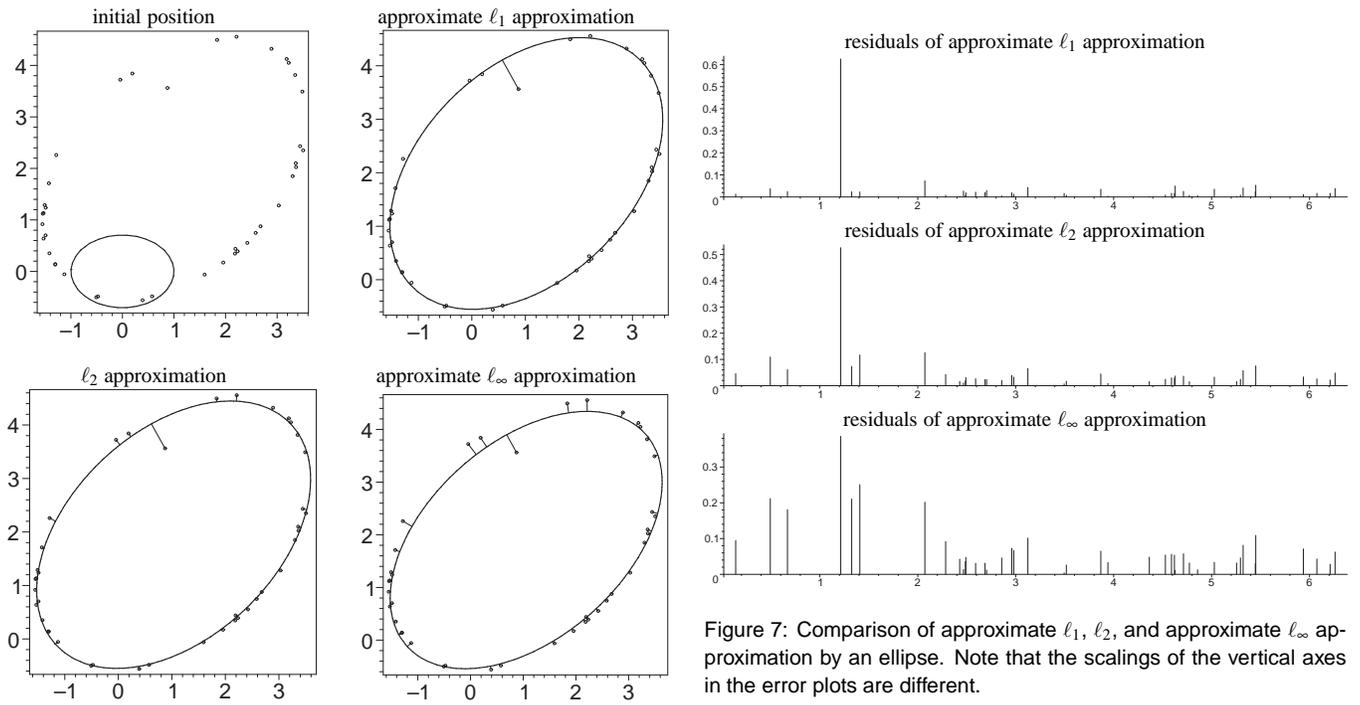
5 CONCLUSION

We investigated several methods for fitting implicit curves/surfaces to a given data set. More precisely, we generalized the usual Gauss-Newton technique for a least-squares approximation by replacing the ℓ_2 norm of the vector of residuals vector by a norm-like function $N(x)$. In particular, this norm like function can be chosen as an approximation of ℓ_1 or ℓ_∞ . Each choice provides certain advantages, depending on the error distribution.

All methods can be equipped with two equivalent interpretations. One may either see them as discrete iterative methods, or as evolution processes, where the evolution of a shape is governed by a differential equation. We prefer the latter framework, as it allows to introduce additional constraints, such as the distance field constraint (see Section 4.1 and [35]) as well as range, volume and convexity constraints, see [11].

If the residuals are chosen to be the geometric distances from the data point to the curve, then one obtains a weighted version of the evolution method introduced by [5], but now for the case of implicitly defined curves and surfaces. In this setting the evolution is obtained by assigning to certain curve points a given velocity. The curve points are simply the closest points on the curve and the velocities are obtained from the distances from these points to their associated data points. This technique has been called the *direct* approach.

Alternatively, one can replace the exact geometric distances by some approximate distance. When choosing the Sampson distance, the obtained Gauss-Newton-type technique yields again a method that can be interpreted as an evolution. But in contrast to the previous method the velocities are assigned at the data points, and the method has been called the *data-based* one.



Clearly, the latter method does not need closest point computations, which is an advantage. On the other hand, one may feel more comfortable with the direct method, as it works with the real distances, and not with approximations.

The aim of this paper was to analyze the different methods from a theoretical point of view, and to illustrate the theory by a few simple but representative examples. Future work will be devoted to the practical exploitation of these observations in a framework for 3D object reconstruction, similar to the results in [36].

REFERENCES

- [1] M. Aigner and B. Jüttler. Robust computation of foot points on implicitly defined curves. In M. Dæhlen, K. Mørken, and L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces: Tromsø 2004*, pages 1–10. Nashboro Press, Brentwood, 2005.
- [2] M. Aigner and B. Jüttler. Approximation flows in shape manifolds. In P. Chenin, T. Lyche, and L. Schumaker, editors, *Curve and Surface Design: Avignon 2006*, pages 1–10. Nashboro Press, 2007.
- [3] M. Aigner and B. Jüttler. Distance regression by Gauss-Newton-type methods and iteratively re-weighted least-squares. *submitted*, 2007. FSP report no. 54, available at www.ig.jku.at.
- [4] M. Aigner and B. Jüttler. Hybrid curve fitting. *Computing*, 79:237–247, 2007.
- [5] M. Aigner, Z. Šír, and B. Jüttler. Evolution-based least-squares fitting using pythagorean hodograph spline curves. *Comput. Aided Geom. Design*, 24:310–322, 2007.
- [6] I. Al-Subaihi and G. A. Watson. The use of the l_1 and l_∞ norms in fitting parametric curves and surfaces to data. *Appl. Numer. Anal. Comput. Math.*, 1:363–376, 2004.
- [7] M. Alhanaty and M. Bercovier. Curve and surface fitting and design by optimal control methods. *Computer-Aided Design*, 33:167–182, 2001.
- [8] A. Atieg and G. Watson. Use of ℓ_p norms in fitting curves and surfaces to data. *Australian and New Zealand Industrial and Applied Mathematics Journal*, 45 (E):C187–C200, 2004.
- [9] A. Blake and M. Isard, editors. *Active contours*. Springer, New York, 2000.
- [10] J. C. Carr et al. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. SIGGRAPH'01*, pages 67–76, 2001.
- [11] R. Feichtinger, M. Fuchs, B. Jüttler, O. Scherzer, and H. Yang. Dual evolution of planar parametric spline curves and T-spline level sets. *Computer-Aided Design*, 2008. in press.
- [12] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley Mass., 1996.
- [13] P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.
- [14] B. Jüttler. Computational methods for parametric discrete ℓ_1 and ℓ_∞ curve fitting. *Int. J. Shape Modelling*, 4:21–34, 1998.
- [15] B. Jüttler. Least-squares fitting of algebraic spline curves via normal vector estimation. In *Proceedings of the 9th IMA Conference on the Mathematics of Surfaces*, pages 263–280, London, UK, 2000. Springer-Verlag.
- [16] B. Jüttler and A. Felis. Least squares fitting of algebraic spline surfaces. *Adv. Comput. Math.*, 17:135–152, 2002.
- [17] V. Mahadevan, H. Narasimha-Iyer, B. Roysam, and H. L. Tanenbaum. Robust model-based vasculature detection in noisy biomedical images. *IEEE Transactions on Information Technology in Biomedicine*, 8(3):360–376, 2004.
- [18] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Proc. SMI'04*, pages 31–39, 2004.
- [19] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, New York, 2002.
- [20] H. Pottmann et al. Industrial geometry: recent advances and applications in CAD. *Computer-Aided Design*, 37:751–766, 2005.
- [21] H. Pottmann and S. Leopoldseder. A concept for parametric surface fitting which avoids the parametrization problem. *Computer Aided Geometric Design*, pages 343–362, 2003.
- [22] V. Pratt. Direct least-squares fitting of algebraic surfaces. In *SIGGRAPH'87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 145–152, New York, NY, USA, 1987. ACM.
- [23] A. Raviv and G. Elber. Three dimensional freeform sculpting via zero sets of scalar trivariate functions. In *Proc. 5th ACM Symposium on Solid Modeling and Applications*, pages 246–257, 1999.
- [24] D. Rogers and N. Fog. Constrained B-spline curve and surface fitting. *Computer-Aided Design*, pages 641–648, 1989.
- [25] P. D. Sampson. Fitting conic sections to 'very scattered' data: An iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, 18:97–108, 1982.
- [26] B. Sarkar and C.-H. Menq. Parameter-optimization in approximating curves and surfaces to measurement data. *Computer Aided Geometric Design*, pages 267–280, 1991.
- [27] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Trans. Graph. (Proc. SIGGRAPH'03)*, 22(3):477–484, 2003.
- [28] C. Shen, J. F. O'Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 896–904, New York, NY, USA, 2004. ACM.
- [29] T. Speer, M. Kuppe, and J. Hoschek. Global reparametrization for curve approximation. *Computer Aided Geometric Design*, 15, 1998.
- [30] 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.
- [31] L. Velho, J. Gomes, and L. H. Figueiredo. *Implicit Objects in Computer Graphics*. Springer Verlag, New York, 2002.
- [32] W. Wang, H. Pottmann, and Y. Liu. Fitting B-spline curves to point clouds by squared distance minimization. *ACM Transactions on Graphics*, 25(2), 2006.
- [33] G. A. Watson. On two methods for discrete ℓ_p approximation. *Computing*, 18(3):263–266, 1977.
- [34] G. A. Watson. On the Gauss-Newton method for ℓ_1 orthogonal distance regression. *IMA J. Num. Anal.*, 22:345–357, 2001.
- [35] 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.
- [36] H. Yang and B. Jüttler. Meshing non-uniformly sampled and incomplete data based on displaced T-spline level sets. In *Proc. Shape Modeling and Applications*, pages 251–260. IEEE Computer Society, 2007.
- [37] H. Yang, B. Jüttler, O. Scherzer, and M. Fuchs. Evolution of t-spline level sets with distance field constraints for geometry reconstruction and image segmentation. In *Shape Modeling and Applications 2006*, pages 247–252. IEEE Press, 2006.
- [38] Z. Yang, J. Deng, and F. Chen. Fitting unorganized point clouds with active implicit b-spline curves. *The Visual Computer*, 21:831–839, 2005.
- [39] H. F. Zeng, Z. G. Liu, and Z. H. Lin. Pde-driven implicit reconstruction of 3d object. In *CGIV '05: Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, pages 251–256, Washington, DC, USA, 2005. IEEE Computer Society.
- [40] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods*, pages 194–201, 2001.