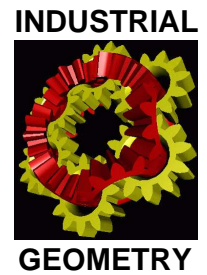


Forschungsschwerpunkt S92

Industrial Geometry

<http://www.industrial-geometry.at>



FSP Report No. 88

Robust Fitting of Implicitly Defined Surfaces using Gauss-Newton-type techniques

M. Aigner, B. Jüttler

October 2009

FWF

Der Wissenschaftsfonds.



Robust Fitting of Implicitly Defined Surfaces using Gauss-Newton-type techniques

Martin Aigner

Bert Jüttler

October 23, 2009

Abstract

We describe Gauss-Newton type methods for fitting implicitly defined curves and surfaces to given unorganized data points. The methods are suitable not only for least-squares approximation, but they can also deal with general error functions, such as approximations to the ℓ_1 or ℓ_∞ norm of the vector of residuals. Two different definitions of the residuals will be discussed, which lead to two different classes of methods: direct methods and data-based ones. In addition we discuss the continuous versions of the methods, which furnish geometric interpretations as evolution processes.

It is shown that the data-based methods – which are less costly, as they work without the computation of 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

We summarize the literature on surface fitting and describe the contributions of this paper.

1.1 Surface Fitting

Fitting a surface (or a curve) 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. In the case of parametric surfaces, due to the influence of the parameterization, the fitting prob-

lem leads to a non-linear optimization problem. Different approaches for dealing with the effects of this non-linearity have been developed, including various variants of Gauss-Newton type techniques [5, 7, 9, 13, 23, 24, 27, 29, 32, 36].

This paper focuses on the implicit representation of surfaces (see e.g. [35]), which offers various advantages for 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.

Various spaces of functions have been used to obtain the implicit representations. Besides polynomials, these include discretized level sets [22, 43], spline functions [17, 26] and spaces spanned by radial basis functions [10, 20].

Several techniques for approximating unorganized point cloud data by implicitly defined curves and surfaces are described in the literature. The fitting of conic sections to scattered data, whereby a special technique for estimating the residuals is used, is discussed in [28]. While a simple linear normalization is used in [25] in order to avoid the trivial solution, a data-dependent normalization is introduced in [34]. The evolution of discretized level sets for surface reconstruction is presented in [42, 43]. In [31] the reconstruction of implicit surfaces from given polygonal models is considered. Active implicit B-spline curves for fitting unorganized point clouds were used in [41], extending the geometric distance minimization in the case of implicitly defined curves using a trust-region algorithm.

Different methods for the case of orientable point cloud data, where each point can be equipped with a normal vector, are described in [10, 16, 33, 39]. These techniques use off-surface points or the simultaneous approximation

of points and normals in order to avoid the trivial solution.

The use of T-spline level sets (which were introduced in [30]) is proposed in [11, 39, 40] making it possible to adapt the distribution of the degrees of freedom to the given geometric data. These papers propose an evolution process combined with a distance field constraint that completely avoids the use of frequent re-initialization steps. The presented techniques are capable of dealing with various constraints such as convexity, area/volume and range constraints.

Most of these fitting techniques 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 on 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. [37] describes two methods for discrete ℓ_p approximation. [15] 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 [38]. [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. The latter is a classical tool in the field of robust statistics [14], by extending the observations in [18] to the case of vector-valued residuals.

1.2 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. As a main difference to the parametric case, the non–linearity – and hence the use of Gauss–Newton techniques – is not due to the influence of the parameterization, but to the minimization of a general non-linear function of the vector of residuals, which is not necessarily the sum of squares.

We consider two possible definitions of the residuals. This leads to two classes of methods, which will be called *direct methods* (see Section 3) and *data-based methods* (see Section 4), 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. 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.

2 Preliminaries

Let a number of points $\{\mathbf{p}_j\}_{j=1..M} \subset \mathbb{R}^d$ in the plane ($d = 2$) or in space ($d = 3$) be given. We are interested to find a hypersurface (i.e. a curve or a surface) that approximates these points. Since the used concepts do not differ for curves and surfaces we will refer mostly to the surface case.

A surface can be described implicitly as the zero-set of a (as we shall assume) \mathcal{C}^2 smooth trivariate function $f_{\mathbf{s}} : \mathbb{R}^3 \rightarrow \mathbb{R}$, i.e.

$$\mathcal{F} = \{\mathbf{x} \in \Omega \subset \mathbb{R}^3 \mid f_{\mathbf{s}}(\mathbf{x}) = 0\}. \quad (1)$$

The notation $f_{\mathbf{s}}$ shall emphasize that the underlying function can entirely be described by a vector $\mathbf{s} = (s_1, \dots, s_m)^\top$ of m shape parameters, where $\mathbf{s} \in \mathbb{R}^m$. We assume that $f_{\mathbf{s}}$ is twice differentiable with respect to the shape parame-

ters \mathbf{s} . The gradients with respect to \mathbf{x} and \mathbf{s} will be denoted by

$$\nabla_{\mathbf{x}} = \left(\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_d} \right) \quad \text{and} \quad \nabla_{\mathbf{s}} = \left(\frac{\partial}{\partial s_1}, \dots, \frac{\partial}{\partial s_m} \right),$$

respectively.

Example 2.1. The important class of algebraic curves/surfaces uses a polynomial of a given degree n as function $f_{\mathbf{s}}(\mathbf{x})$.

In the case $d = 2$, one may e.g. choose the bivariate Bernstein-Bézier basis with respect to a suitable domain triangle $\Delta ABC \subset \mathbb{R}^2$,

$$f_{\mathbf{s}}(\mathbf{x}) = \sum_{i+j+k=n} b_{ijk} \frac{n!}{i!j!k!} u^i v^j w^k, \quad (2)$$

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 2.2. More generally, one may consider implicitly defined surfaces

$$f_{\mathbf{s}}(\mathbf{x}) = \sum_{i=0}^N s_i \phi_i(\mathbf{x}), \quad (3)$$

where the basis functions ϕ_i can be chosen, e.g., as radial basis functions [10], tensor-product splines or subsets thereof [17] or T-splines [39].

3 Geometric Fitting

When fitting a curve/surface to given data, one minimizes usually certain geometric distances to the given data points \mathbf{p}_j . As the data exceed in general the degrees of freedom this is done via a least-squares approach. In the sequel we generalize this technique.

3.1 The generalized fitting problem and norm-like functions

In many application a least-squares approach is not appropriate as it assumes implicitly that the measured errors follow a Gaussian distribution. However, in the presence of erroneous data this assumption cannot be justified.

Thus we extend the usual least-squares approach to general functions by considering the objective function

$$F = \sum_{j=1}^M N(\|\mathbf{p}_j - \mathbf{c}_j(\mathbf{s})\|) \rightarrow \min \quad (4)$$

where

$$\mathbf{c}_j(\mathbf{s}) = \operatorname{argmin} \|\mathbf{p}_j - \mathbf{p}^c\| \quad \text{subject to } f_{\mathbf{s}}(\mathbf{p}^c) = 0, \quad (5)$$

i.e., $\mathbf{c}_j(\mathbf{s})$ is the closest point from the fitting curve/surface to the data point \mathbf{p}^c . This more general technique replaces the usual ℓ_2 norm of the residuals by a function $N(x)$. It is assumed to satisfy the following definition of a norm-like function (cf. [21, 3]):

Definition 3.1. 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

$$N'(x) = x \cdot w(x) \quad \text{for } x \in (0, \varepsilon] \quad (6)$$

where the associated **weight function** $w(x)$ is positive. If the weight function $w = w(x)$ is the restriction of a \mathcal{C}^1 function such that $w : [0, \varepsilon] \rightarrow [k, K]$ with $k, K \in \mathbb{R}^+$, then we will call it **positive** and **bounded**.

For example, the norm-like functions

$$N(x) = x^p \quad (7)$$

are associated with the weight functions

$$w(x) = px^{p-2}. \quad (8)$$

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 (due to $w(0) = 0$). For $p = 2$ the weight function is both, positive and bounded. Of course this special case reduces to the usual ℓ_2 approximation.

As another example, consider the norm-like function

$$N(x) = 1 - \exp(-x^2) \quad (9)$$

with the weight function

$$w(x) = 2 \exp(-x^2). \quad (10)$$

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

3.2 A Gauss–Newton–type method

In the following we apply the widely used Gauss–Newton approach to the non-linear problem (4). In a first step we provide the gradient and the Hessian of the objective function, where we use the abbreviation $\mathbf{R}_j = \mathbf{p}_j - \mathbf{c}_j(\mathbf{s})$ for the residual vectors,

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

The gradient $-\nabla_{\mathbf{s}} \mathbf{c}_j(\mathbf{s})$ can be computed from the condition $f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s})) \equiv 0$, which implies

$$(\nabla_{\mathbf{s}} f_{\mathbf{s}})(\mathbf{c}_j(\mathbf{s})) + (\nabla_{\mathbf{x}} f_{\mathbf{s}})(\mathbf{c}_j(\mathbf{s})) \nabla_{\mathbf{s}} \mathbf{c}_j(\mathbf{s}) = 0. \quad (11)$$

For any regular surface point, i.e. $\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s})) \neq 0$, the residual \mathbf{R}_j is parallel to the gradient $\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s}))$:

$$\frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s}))}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s}))\|} = \text{sign}(\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s})) \mathbf{R}_j) \frac{\mathbf{R}_j^\top}{\|\mathbf{R}_j\|}. \quad (12)$$

Combining (11) and (12) yields

$$\frac{\mathbf{R}_j^\top}{\|\mathbf{R}_j\|} \nabla_{\mathbf{s}} (-\mathbf{c}_j(\mathbf{s})) = \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s}))}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s}))\|} \text{sign}(\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s})) \mathbf{R}_j)$$

which gives the gradient of the objective function

$$\nabla F = \sum_{j=1}^M w_j \|\mathbf{R}_j\| \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s}))}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s}))\|} \text{sign}(\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{c}_j(\mathbf{s})) \mathbf{R}_j).$$

where we used the abbreviation $w_j = w(\|\mathbf{R}_j\|)$. As a further simplification we omit the arguments in the following discussion. However, we keep in mind that $f_{\mathbf{s}}$ and its gradients are evaluated at the surface points $\mathbf{c}_j(\mathbf{s})$. Thus the Hessian is

$$\begin{aligned}H_F &= \nabla(\nabla F^\top) \\ &= \sum_{j=1}^M w'_j \|\mathbf{R}_j\| \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} + w_j \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \\ &\quad + w_j \|\mathbf{R}_j\| \left(\frac{\nabla_{\mathbf{s}} \nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} - \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}^\top \nabla_{\mathbf{x}} f_{\mathbf{s}} \nabla_{\mathbf{s}} \nabla_{\mathbf{x}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|^3} \right)\end{aligned} \quad (13)$$

During the iteration, the Hessian has to be evaluated in each single step. Especially in the presence of a high number of data points this might be computationally expensive. Moreover, if the function F is only \mathcal{C}^1 , no second order derivatives are available and the Hessian does not exist. For this reason we do not use the exact Hessian, but apply some simplifications to (13) in order to obtain an approximate version. More precisely, we adopt the paradigm of Gauss–Newton methods and omit the first and the last part of the expansion in (13),

$$H_F^* = \sum_{j=1}^M w_j \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|}. \quad (14)$$

The following result justifies the choice of this simplification.

Proposition 1. Assume that $\bar{\mathbf{s}}$ is a minimizer of (4) such that $f_{\bar{\mathbf{s}}}(\mathbf{p}_j) = 0$ and $\|\nabla_{\mathbf{x}} f_{\bar{\mathbf{s}}}(\mathbf{p}_j)\| > 0$ for all data points \mathbf{p}_j . Let $\mathbf{s} \rightarrow f_{\bar{\mathbf{s}}}(\mathbf{x}) \in \mathcal{C}^2(\mathbb{R}^m, \mathbb{R})$ be bounded for all $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and w be 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 $f_{\bar{\mathbf{s}}}(\mathbf{x})$ and its \mathcal{C}^2 continuity implies that the derivatives of $f_{\bar{\mathbf{s}}}(\mathbf{x})$ up to second order are bounded as well. Now we consider the first and the third term in (13) separately. For the first term we obtain

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

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 $\|\nabla_{\mathbf{x}} f_{\bar{\mathbf{s}}}(\mathbf{p}_j)\| > 0$. The third term can be bounded by

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

Again, this part vanishes due to the boundedness of all involved derivatives and $\|\nabla_{\mathbf{x}} f_{\bar{\mathbf{s}}}\| > 0$. The remaining second term coincides with the approximate Hessian H_F^* . \square

From this result we can conclude immediately - similar to the well known Gauss–Newton method for ℓ^2 approximation - a quadratic convergence rate in the limit.

Summing up, we obtain indeed a generalized Gauss–Newton–type method with the desired properties if we built the linear system

$$H_F^* \Delta \mathbf{s} + \nabla F^\top = 0 \quad (15)$$

which leads to

$$\begin{aligned} \sum_{j=1}^M w_j \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \Delta \mathbf{s} \\ + \sum_{j=1}^M w_j \|\mathbf{R}_j\| \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \text{sign}(\nabla_{\mathbf{x}} f_{\mathbf{s}} \mathbf{R}_j) = 0. \end{aligned} \quad (16)$$

As all functions and gradients are evaluated at points of the surface, we will refer to (16) as the *surface-based Gauss-Newton-type method* for minimizing general functions $N = N(\|\mathbf{R}_j\|)$ of the residuals in the case of implicitly defined surfaces.

3.3 Continuous version and its geometrical interpretation

In this section we propose an interpretation of the Gauss-Newton method that shall provide additional geometrical insight in the fitting process. Recall that in each step of the approximation described in the last section the initial surface moves closer to the target points. This movement shall now be interpreted as discrete steps of a continuous evolution process.

In fact, equation (16) can be seen as the minimum condition of (where we used (12)):

$$\sum_{j=1}^M w_j \left(\left(\mathbf{R}_j + \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \frac{\nabla_{\mathbf{s}} f_{\mathbf{s}}}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \dot{\mathbf{s}} \right)^\top \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|} \right)^2 \rightarrow \min_{\dot{\mathbf{s}}}, \quad (17)$$

where we simply replaced the discrete update $\Delta \mathbf{s}$ by its continuous counter part $\dot{\mathbf{s}}$.

Note that

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

is the unit normal of a regular point on the surface and that

$$\mathbf{v}^n = - \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|^2} \dot{f}_{\mathbf{s}} = - \frac{\nabla_{\mathbf{x}} f_{\mathbf{s}}^\top}{\|\nabla_{\mathbf{x}} f_{\mathbf{s}}\|^2} \nabla_{\mathbf{s}} f_{\mathbf{s}} \dot{\mathbf{s}},$$

is the normal velocity of a regular surface point. Hence we can rewrite (17) as the least-squares problem

$$\sum_{j=1}^M w_j ((\mathbf{R}_j - \mathbf{v}^n)^\top \mathbf{n}_j)^2 \rightarrow \min_{\dot{\mathbf{s}}}. \quad (18)$$

This reformulation furnishes following interpretation:

Each data point attracts its associated closest point on the surface with a certain velocity \mathbf{v}^n . In the normal direction of the surface, the velocity shall be such that it has the same magnitude as the distance to the data point. The difference between the actual and the imposed velocity is minimized in the least-squares sense, where each point is weighted with the corresponding value of the weight function w_j .

This geometric interpretation of the geometric fitting is visualized in Fig. 1, top. See also [2, 4] for a more abstract discussion of this type of shape evolutions for the case of least-square approximation (i.e., $N(x) = x^2$).

3.4 Remarks on closest point computation

Note that for the direct distance minimization 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.

Consequently, it is desirable to avoid the computation of closest points as will be shown in the next section.

4 Algebraic Fitting

This method relies on the following observation. By definition, a point \mathbf{x} lies on an implicitly defined surface if its algebraic distance to the surface is zero, i.e., $f_{\mathbf{s}}(\mathbf{x}) = 0$. For points in the vicinity of the surface the function value is expected to be close to zero.

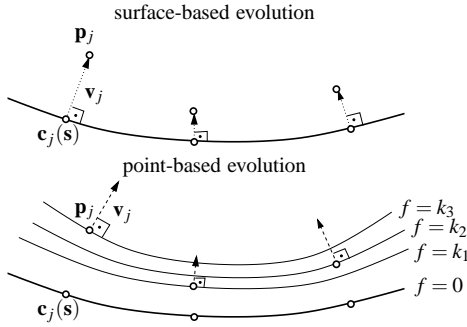


Figure 1: Velocities for surface-based evolution are imposed at closest points on surface (top). Velocities for point-based evolution are imposed at data points (bottom).

4.1 The generalized fitting problem for Sampson distances

Taubin [34] proposed to use the Sampson distance [28] as an approximation to the exact geometric distance. It is the gradient-weighted algebraic error,

$$d_S = \frac{|f_s(\mathbf{p}_j)|}{\|\nabla_{\mathbf{x}} f(\mathbf{p}_j)\|}. \quad (19)$$

With this we obtain the objective function

$$F = \sum_{j=1}^M N \left(\frac{|f_s(\mathbf{p}_j)|}{\|\nabla_{\mathbf{x}} f_s(\mathbf{p}_j)\|} \right) \rightarrow \min. \quad (20)$$

As an important observation we note that in contrast to the direct distance minimization, the gradients and functions are now evaluated at the data points.

4.2 A Gauss-Newton-type method

In the following discussion we skip again the arguments (keeping the dependency on the data points \mathbf{p}_j in mind). The gradient of (20) is

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

The second term vanishes for zero-residual problems. Omitting it and using $N'(x) = xw(x)$ yields an approxi-

mate gradient,

$$\nabla F^* = \sum_{j=1}^M w_j \frac{f_s \nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|^2}.$$

The Hessian of (20) is

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

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 2. Assume that $\bar{\mathbf{s}}$ is a minimizer of (20) such that $f_{\bar{\mathbf{s}}}(\mathbf{p}_j) = 0$ and $\|\nabla_{\mathbf{x}} f_{\bar{\mathbf{s}}}(\mathbf{p}_j)\| > 0$. Let $\mathbf{s} \rightarrow f_s(\mathbf{x}) \in \mathcal{C}^2(\mathbb{R}^m, \mathbb{R})$ be bounded for all $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and w be 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 1.

As for the geometric fitting we obtain a linear system for the update vector $\Delta \mathbf{s}$:

$$\sum_{j=1}^M w_j \frac{\nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|} \frac{\nabla_s f_s}{\|\nabla_{\mathbf{x}} f_s\|} \Delta \mathbf{s} + \sum_{j=1}^M w_j \frac{f_s \nabla_s f_s^\top}{\|\nabla_{\mathbf{x}} f_s\|^2} = 0 \quad (21)$$

However, the difference to the direct method is significant. No closest point computation is needed since all functions and derivatives are evaluated at the data points. Therefore we call (21) *data-based Gauss-Newton-type method*.

Fig. 2 shows a first example of a data-based Gauss-Newton-type method for ℓ_2 approximation.

4.3 Continuous version and geometric interpretation

Similar to the surface-based Gauss-Newton-type method we interpret also the data-based method as an evolution

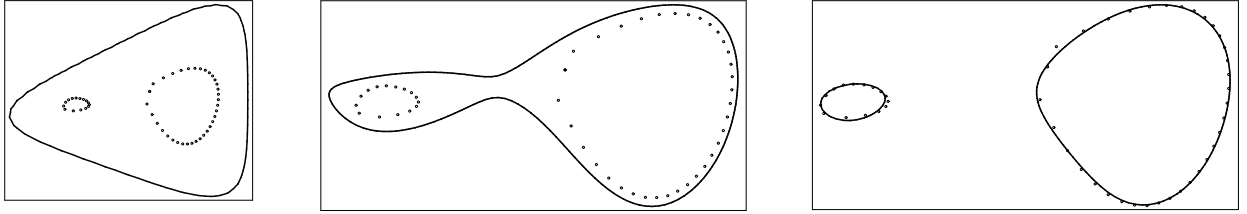


Figure 2: Initial value (left), an intermediate step (center) and the final result (right) of a data-based Gauss–Newton-type method (or equivalently: of an evolution process) for ℓ_2 -approximation.

with a suitable speed function. Again, it can easily be verified that

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

is the minimum condition of (21) where we replaced again $\Delta \mathbf{s}$ by $\dot{\mathbf{s}}$. This furnishes the following geometric interpretation, which is different from the surface-based case:

Each data point imposes a velocity on the level set of the surfaces passing through it. In the normal direction of the level set, the velocity shall be equal to the Sampson distance of the point to the surface. The difference between the actual and the imposed velocity is minimized in the least-squares sense, where each point is weighted with the corresponding value of the weight function w_j .

This geometric interpretation of the algebraic fitting is visualized in Fig. 1, bottom.

5 Discussion and examples

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

5.1 Direct methods vs. iterative methods

Many direct techniques, such as Taubin’s method [34], provide the result after a single step (though an iterative

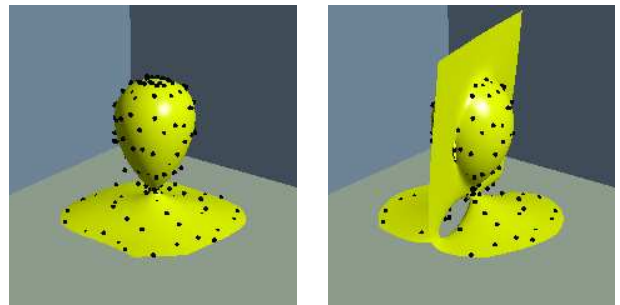


Figure 3: 180 points from a parametric surface and randomly generated error. Data-based evolution for ℓ_2 approximation with distance field constraint (left), see text for details. Taubin’s method (right) is not able to deal with data, due to the incorrectly chosen degree of the algebraic surface.

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 Fig. 3. The latter technique gives the correct result, while Taubin’s fit encounters 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.

5.2 Regularization

An inherent problem of fitting with implicit curves/surfaces is that the solution is not unique as all functions $\lambda f(\mathbf{x})$ with $\lambda \in \mathbb{R} \setminus \{0\}$ possess the same zero contour (1). Consequently, suitable regularization terms have to be incorporated in order to avoid this ambiguity.

A simple strategy is to constrain the values of some coefficients, such as the constant term in the case of algebraic curves [25]. However, this approach constrains the space of solutions as it prevents the curve from passing through the origin. Instead, one should prefer data-dependent regularization functionals as in [34]. Also, 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, [17].

In addition to these constraints that shall mainly ensure a unique solution to the optimization problem, one may also want to influence the shape of the resulting curve/surface. Especially algebraic surfaces suffer often from unwanted oscillations, which may produce unwanted branches in the domain of interest. These can be avoided by adding regularization terms such as the thin plate energy, see [17].

In the sequel we discuss the *distance field constraint*, which was proposed in [39, 40]. 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

$$\int_{\Omega} \left(\frac{d}{dt} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x})\| + \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x})\| - 1 \right)^2 d\mathbf{x} \rightarrow \min \quad (22)$$

to the objective function. Here we refer to the continuous version of the method, see Sections 3.3 and 4.3.

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 (22) by applying it to a number of points \mathbf{x}_i in the domain of interest. This leads to

$$\sum_i \left(\frac{d}{dt} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x}_i)\| + \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x}_i)\| - 1 \right)^2 \rightarrow \min. \quad (23)$$

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.

Alternatively, one may also impose a side condition similar to the one proposed in [34]. There it was required that the average norm of the gradients in the data points shall be equal to 1. In our setting this idea can be realized via

$$\left(\sum_{i=1}^M \frac{d}{dt} \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x}_i)\| + \sum_{i=1}^M \|\nabla_{\mathbf{x}} f_{\mathbf{s}}(\mathbf{x}_i)\| - M \right)^2 \rightarrow \min. \quad (24)$$

Note, that in contrast to the distance field constraint, this normalization term does not prevent the evolution from finding the exact solution. In fact it only rescales the defining equation $f_{\mathbf{s}}$. However, (24) is only one single condition on the coefficients.

In the presence of additional singularities, e.g. if the number of degrees of freedom of $f_{\mathbf{s}}$ has been chosen too large, one still needs the distance field constraint in order to avoid degeneracies such as additional branches in the domain of interest.

5.3 Step size control

In order to ensure a robust convergence behaviour, the step size parameter $h \leq 1$ has to be chosen carefully. Using the interpretation of an evolution, h can be computed such that the points $\mathbf{c}_{\mathbf{s}}(u_j)$ do not move more than a certain threshold. As a good strategy one may relate this maximal movement to the size of the data set (eg. the diameter of the bounding box). However, the step size should be relaxed to 1 in the limit to obtain fast convergence rates.

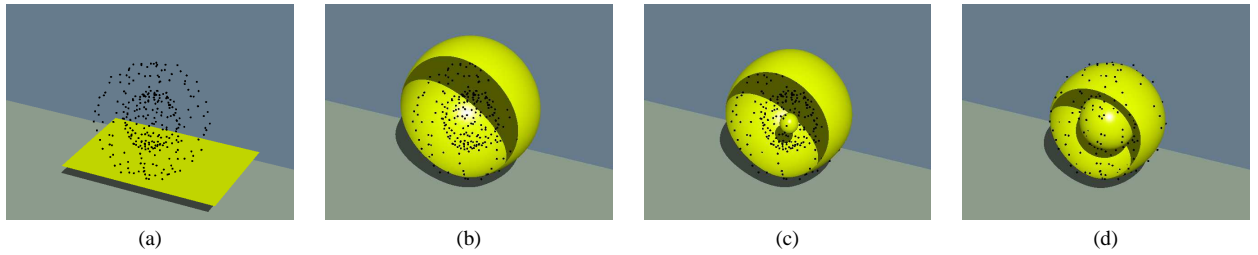


Figure 4: We consider a point cloud that consists of two nested spheres. As an initial value we choose a plane as displayed in (a). During the evolution process for ℓ_2 approximation of the Sampson distances, the surface matches first the outer part of the points (b). Simultaneously, a second branch of the surfaces emerges inside the outer sphere, which fits itself to the inner points (c). Note that for visualization purposes a part of the outer surface is clipped away.

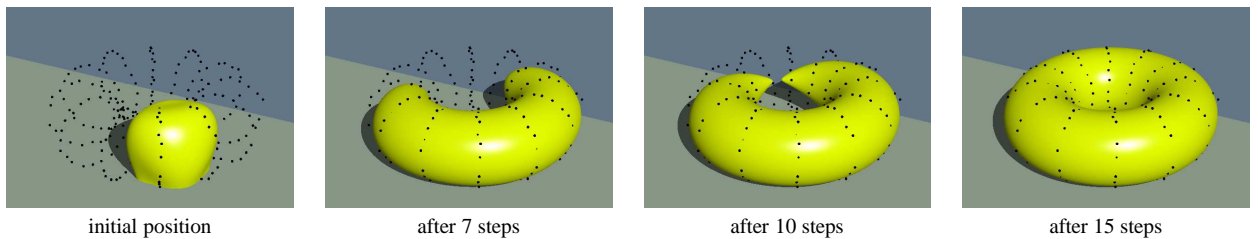


Figure 5: Step wise ℓ_2 approximation of 240 points sampled from a torus with additional random error.

5.4 Surface-based vs. data-based methods

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, see Fig. 1.

In the surface-based approach (top), 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 (bottom), 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 $f = \text{const.}$ 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 Fig. 4.

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 can deal with such situations as shown in Fig. 4 better than the surface-based approach. Using the latter method, the initial surface converges in the beginning towards the outer sphere 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 the surface-based evolution may be preferred 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 Fig. 5. A point cloud sampled from a torus was approximated using the ℓ_2 norm.

5.5 Robust fitting via general norm-like functions

Finally we demonstrate the advantages of general norm-like functions. We present a comparison of an approximate ℓ_1 , an ℓ_2 , and an approximate ℓ_∞ fit, see Fig. 6. While the first one uses again the weight function which is obtained from (9), 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 norms of the residual vectors are 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 (0.26 vs. 0.46 and 0.49 for the ℓ_2 and the approximate ℓ_1 approximation).

Fig. 7 shows the same effect for spatial data, sampled from a hyperboloid of revolution. 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.

6 Conclusion

We investigated several methods for fitting implicitly defined surfaces to given point data. More precisely, we generalized the usual Gauss-Newton technique for least-squares approximation by replacing the ℓ_2 norm of the

vector of residuals 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. The latter framework allows to introduce additional constraints, such as the distance field constraint (see Section 5.1 and [39]) as well as range, volume and convexity constraints, see [11].

If the residuals are chosen to be the exact geometric distances from the data points to the surface, 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 surface points a given velocity. The surface 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 may be devoted to the practical exploitation of these observations in a framework for 3D object reconstruction, similar to the results in [40].

Currently, our implementation - which is based on algebraic surfaces (i.e. zero sets of polynomials)- is limited to relatively small data sets, due to the global support of the basis functions.

In order to deal with real-world scattered data sets, the methods presented in this paper should be combined with

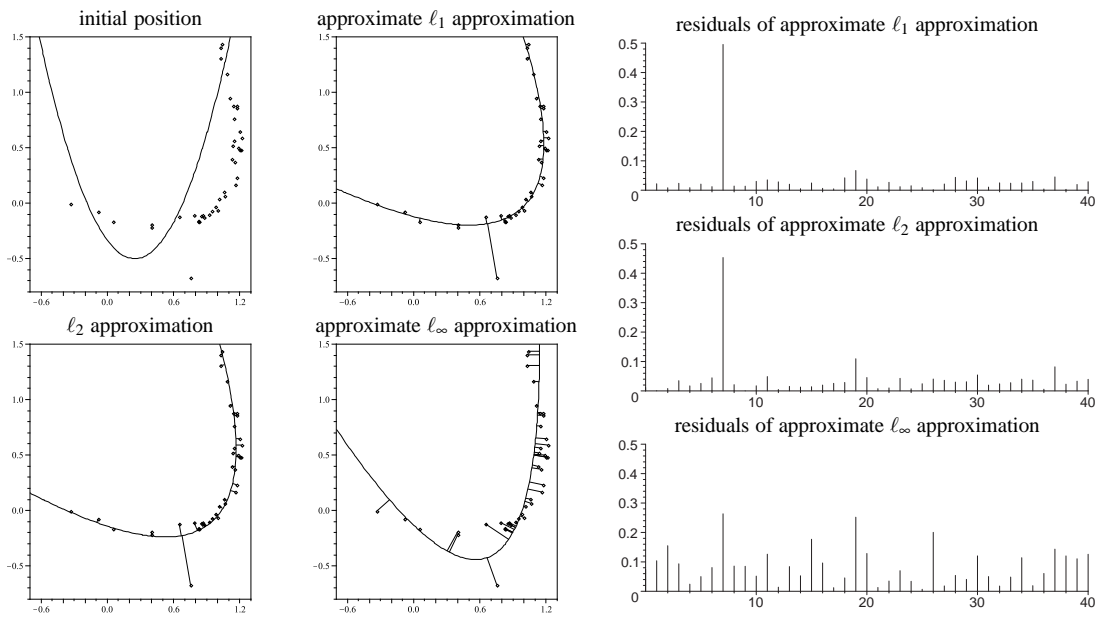


Figure 6: Comparison of approximate ℓ_1 , ℓ_2 , and approximate ℓ_∞ approximation of 40 data points with an outlier by a parabola.

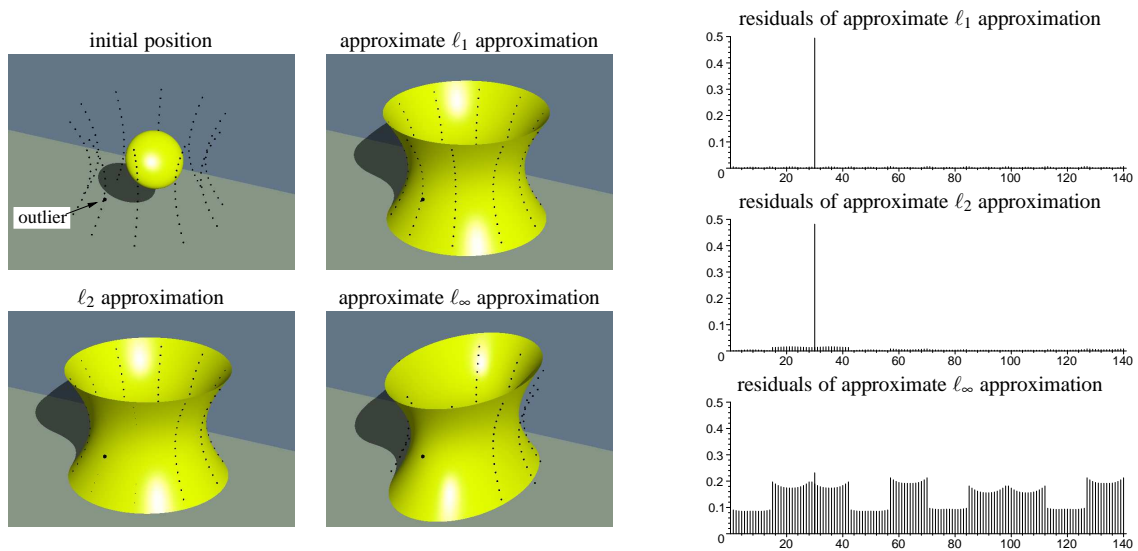


Figure 7: Comparison of approximate ℓ_1 , ℓ_2 , and approximate ℓ_∞ approximation of 140 data points with an outlier by a quadric.

hierarchical approaches and segmentation techniques, see [12, 19].

References

- [1] Aigner, M., Jüttler, B.: Robust computation of foot points on implicitly defined curves. In: M. Dæhlen, K. Mørken, L. Schumaker (eds.) *Mathematical Methods for Curves and Surfaces: Tromsø 2004*, pp. 1–10. Nashboro Press, Brentwood (2005)
- [2] Aigner, M., Jüttler, B.: Approximation flows in shape manifolds. In: P. Chenin, T. Lyche, L. Schumaker (eds.) *Curve and Surface Design: Avignon 2006*, pp. 1–10. Nashboro Press (2007)
- [3] Aigner, M., Jüttler, B.: Distance regression by Gauss-Newton-type methods and iteratively re-weighted least-squares. Tech. Rep. 54, FSP S92 “Industrial Geometry”, www.ig.jku.at (2007)
- [4] Aigner, M., Jüttler, B.: Hybrid curve fitting. *Computing* **79**, 237–247 (2007)
- [5] Aigner, M., Šír, Z., Jüttler, B.: Evolution-based least-squares fitting using pythagorean hodograph spline curves. *Comput. Aided Geom. Design* **24**, 310–322 (2007)
- [6] Al-Subaihi, I., Watson, G.A.: 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] Alhanaty, M., Bercovier, M.: Curve and surface fitting and design by optimal control methods. *Comp. Aided Design* **33**, 167–182 (2001)
- [8] Atieg, A., Watson, G.: 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] Blake, A., Isard, M. (eds.): *Active contours*. Springer, New York (2000)
- [10] Carr, J.C., et al.: Reconstruction and representation of 3D objects with radial basis functions. In: *SIGGRAPH’01*, pp. 67–76 (2001)
- [11] Feichtinger, R., Fuchs, M., Jüttler, B., Scherzer, O., Yang, H.: Dual evolution of planar parametric spline curves and T-spline level sets. *Comput. Aided Design* **40**, 13–24 (2008)
- [12] Fleishman, S., Cohen-Or, D., Silva, C.T.: Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.* **24**(3), 544–552 (2005)
- [13] Hoschek, J., Lasser, D.: *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley Mass. (1996)
- [14] Huber, P.J.: *Robust Statistics*. John Wiley and Sons, New York (1981)
- [15] Jüttler, B.: Computational methods for parametric discrete ℓ_1 and ℓ_∞ curve fitting. *Int. J. Shape Modelling* **4**, 21–34 (1998)
- [16] Jüttler, B.: Least-squares fitting of algebraic spline curves via normal vector estimation. In: *Proc. of the 9th IMA Conference on the Mathematics of Surfaces*, pp. 263–280. Springer (2000)
- [17] Jüttler, B., Felis, A.: Least-squares fitting of algebraic spline surfaces. *Adv. Comput. Math.* **17**, 135–152 (2002)
- [18] Mahadevan, V., Narasimha-Iyer, H., Roysam, B., Tanenbaum, H.L.: Robust model-based vasculature detection in noisy biomedical images. *IEEE Transactions on Information Technology in Biomedicine* **8**(3), 360–376 (2004)
- [19] Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicits. *ACM Trans. Graph.* **22**(3), 463–470 (2003)
- [20] Ohtake, Y., Belyaev, A., Seidel, H.P.: 3D scattered data approximation with adaptive compactly supported radial basis functions. In: *Proc. SMI’04*, pp. 31–39 (2004)
- [21] Osborne, M.R.: *Finite Algorithms in Optimization and Data Analysis*. John Wiley & Sons, New York (1985)

- [22] Osher, S., Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, New York (2002)
- [23] Pottmann, H., Leopoldseder, S.: A concept for parametric surface fitting which avoids the parametrization problem. *Comput. Aided Geom. Design* pp. 343–362 (2003)
- [24] Pottmann, H., et al.: *Industrial geometry: recent advances and applications in CAD*. *Comput. Aided Design* **37**, 751–766 (2005)
- [25] Pratt, V.: Direct least-squares fitting of algebraic surfaces. In: *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 145–152. ACM, New York, NY, USA (1987)
- [26] Raviv, A., Elber, G.: Three dimensional freeform sculpting via zero sets of scalar trivariate functions. In: *Proc. 5th ACM Symposium on Solid Modeling and Applications*, pp. 246–257 (1999)
- [27] Rogers, D., Fog, N.: Constrained B-spline curve and surface fitting. *Comput. Aided Design* pp. 641–648 (1989)
- [28] Sampson, P.D.: Fitting conic sections to 'very scattered' data: An iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing* **18**, 97–108 (1982)
- [29] Sarkar, B., Menq, C.H.: Parameter-optimization in approximating curves and surfaces to measurement data. *Comput. Aided Geom. Design* pp. 267–280 (1991)
- [30] Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A.: T-splines and T-NURCCs. *ACM Trans. Graph.* **22**(3), 477–484 (2003)
- [31] Shen, C., O'Brien, J.F., Shewchuk, J.R.: Interpolating and approximating implicit surfaces from polygon soup. In: *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pp. 896–904. ACM (2004)
- [32] Speer, T., Kuppe, M., Hoschek, J.: Global reparametrization for curve approximation. *Comput. Aided Geom. Design* **15** (1998)
- [33] Tasdizen, T., Tarel, J.P., Cooper, D.: Improving the stability of algebraic curves for applications. *IEEE Trans. Image Processing* **9**(3), 405–416 (2000)
- [34] Taubin, G.: 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)
- [35] Velho, L., Gomes, J., Figueiredo, L.H.: *Implicit Objects in Computer Graphics*. Springer Verlag, New York (2002)
- [36] Wang, W., Pottmann, H., Liu, Y.: Fitting B-spline curves to point clouds by squared distance minimization. *ACM Transactions on Graphics* **25**(2) (2006)
- [37] Watson, G.A.: On two methods for discrete ℓ_p approximation. *Computing* **18**(3), 263–266 (1977)
- [38] Watson, G.A.: On the Gauss-Newton method for ℓ_1 orthogonal distance regression. *IMA J. Num. Anal.* **22**, 345–357 (2001)
- [39] Yang, H., Fuchs, M., Jüttler, B., Scherzer, O.: Evolution of T-spline level sets with distance field constraints for geometry reconstruction and image segmentation. In: *Shape Modeling International*, pp. 247–252. IEEE Press (2006)
- [40] Yang, H., Jüttler, B.: Evolution of T-spline level sets for meshing non-uniformly sampled and incomplete data. *The Visual Computer* **24**(6), 435–448 (2008)
- [41] Yang, Z., Deng, J., Chen, F.: Fitting unorganized point clouds with active implicit B-spline curves. *The Visual Computer* **21**, 831–839 (2005)
- [42] Zeng, H.F., Liu, Z.G., Lin, Z.H.: Pde-driven implicit reconstruction of 3d object. In: *CGIV '05: Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, pp. 251–256. IEEE Computer Society (2005)
- [43] Zhao, H.K., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: *VLSM '01: Proc. of the IEEE Workshop on Variational and Level Set Methods*, pp. 194–201 (2001)