

Fitting Implicitly Defined Curves to Unorganized Points with Sharp Features

Huaiping Yang and Bert Jüttler

Abstract. In order to represent and reconstruct sharp features (i.e., corners) by an implicitly defined curve, we add so-called corner detector functions, which are special radial basis functions, to the function defining the initial curve. The parameters controlling them are automatically adjusted by applying an evolution process.

§1. Introduction

Implicitly defined curves and surfaces have found numerous applications in shape modeling and geometric computing, including object reconstruction from unorganized points [3, 7, 16, 15] and geometric modeling [4, 8].

While implicit representations offer some advantages for shape reconstruction from unorganized data points, such as the non-existence of the parameterization problem, repairing capabilities of incomplete data and simple operations of shape editing, they also face serious difficulties in accurate reconstruction of sharp features. In order to address this difficulty, [6] proposes anisotropic basis functions which model the asymmetric nature of the object near the sharp feature, where the direction of anisotropy is obtained through principal component analysis. More recently, [8] uses piecewise quadratic functions resulting from Boolean operations.

In the remainder of this paper we consider the following problem: given a set of unorganized data points $(\mathbf{p}_k)_{k=1,2,\dots,n}$ in the plane, compute an implicitly defined curve which approximates the points \mathbf{p}_k . The shape represented by the data points may contain sharp features, and it may consist of several components.

Our approach consists of two steps. In the first step, we generate an initial function f_0 , which is represented by a T-spline, in order to cover the basic shape by an implicitly defined curve. In the second step we detect the sharp features and use corner detector (CD) functions in order to

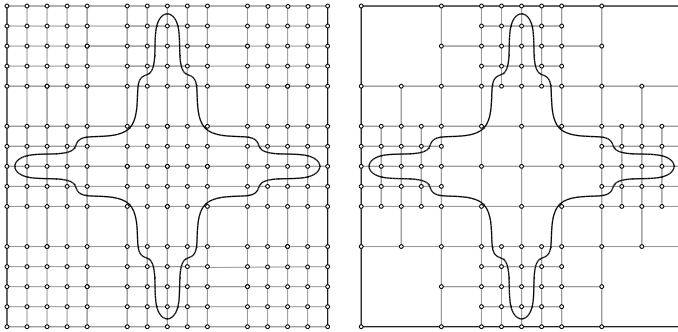


Fig. 1. Comparison a tensor product scalar B-spline (left) and a scalar T-spline (right). The figures show the control grid of the B-spline with 225 control coefficients (left), and that of the T-spline with 133 control coefficients (right).

improve the accuracy of the representation in the vicinity of these curves. The CD functions are special radial basis functions.

Usually, sharp features (vertices) are modeled by combining the implicit equations of the two curves which meet in this point, e.g., using Rvachev's R-functions (cf. [9]). We take a different approach: the vertex is obtained by adding a single CD function, which is described solely by the position of the vertex and the influence radius, to the implicit equation of an existing curve, which remains unchanged. This makes it easy to optimize the position of the vertex by using a local procedure.

§2. Fitting Unorganized Points by Implicitly Defined Curves

The implicitly defined curve $\Gamma = \Gamma(f_0)$ is obtained as the zero set of a suitable function f_0 . We use a bivariate scalar T-spline function, see [11] for details. On the one hand, since the T-spline function is piecewise rational, the implicitly defined curve is piecewise algebraic, and its segments can be pieced together with any desired level of differentiability. On the other hand, the use of T-splines leads to a sparse representation of the geometry. It can be refined locally, by adapting the number and distribution of the degrees of freedom to the particular data.

As an example, Fig. 1 shows the control grids of a tensor product B-spline and a T-spline which define the same curve Γ . In comparison with the use of tensor product splines, the use of T-splines normally needs less control coefficients (in this example: 59%).

In order to find the T-spline f_0 , we use the method described in [14]. First, the T-spline control grid (or T-mesh) is generated according to the distribution of data points. In order to generate the coefficients, the method applies an evolution process to an initial curve. The evolution is governed by a combination of prescribed, data-driven normal velocities

(which are motivated by earlier work in the field of image processing [5], where they have been successfully applied to contour detection and segmentation in images) with additional distance field constraints. The constraints help to avoid additional branches and singularities of the implicit curve, without having to use re-initialization steps.

As the result, we obtain a slightly smoothed version of the target shape described by the data points. Most parts of the target shape are already well fitted, except for sharp features (cf. Fig. 2 (a)). More details can be found in [14]. In particular, that paper discusses a ‘final refinement’ step, which can be used to improve the result, especially for noisy data.

The remainder of the paper is independent of the particular fitting method and on the specific representation of the function f_0 . Other techniques and representations (such as tensor-product splines, radial basis functions, hierarchically defined functions or grid-based discretizations) can be used instead [3, 4, 7, 10, 16].

§3. Dealing with sharp features

After introducing corner detectors and estimating the number and the parameters controlling them, we apply an optimization step in order to obtain the final result.

3.1. Corner detectors

In order to model the sharp feature at a point of the implicitly defined curve, we define the *corner detector* (CD) function

$$g(\mathbf{x}) = g_{a,r,\mathbf{c}}(\mathbf{x}) = a \cdot (\|\mathbf{x} - \mathbf{c}\| - r)_+^2, \quad (1)$$

where $r > 0$ and $a \neq 0$ are constant coefficients which represent the radius of influence region and the magnitude of function value of g , respectively, while \mathbf{c} is the center of the region of influence. Clearly, this is a special instance of a radial basis function (RBF), cf. [3, 13].

By adding this function to an existing bivariate function f_0 , we obtain the new function $f_0 + g$. By setting suitable parameters of the CD function, we are able to model sharp features of the implicit curve $\Gamma(f_0 + g)$.

The CD function g has the following properties:

- (1) g is locally supported, $\text{supp } g = \{ \mathbf{x} \mid \|\mathbf{x} - \mathbf{c}\| \leq r \}$.
 - (2) g is non-negative (non-positive) whenever $a > 0$ ($a < 0$).
 - (3) g is C^∞ within its support, except for the point \mathbf{c} .
 - (4) For any C^1 function f_0 with domain D , the sum $f_0 + g$ is C^1 in $D \setminus \{\mathbf{c}\}$.
- CD functions with higher smoothness can be obtained by increasing the exponent in (1). The shape of the curve $\Gamma(f_0 + g)$ will be discussed in the next section.

3.2. Initial estimates and a bound on the radius

By using the smooth T-spline function f_0 only, the data points around sharp features are not fitted very well (i.e., with a relatively large distance to the curve $\Gamma(f_0)$), and they often correspond to regions with relatively high curvature. Based on this observation we propose the following procedure for estimating the number and initial values of the parameters of the CD functions (cf. Fig. 2):

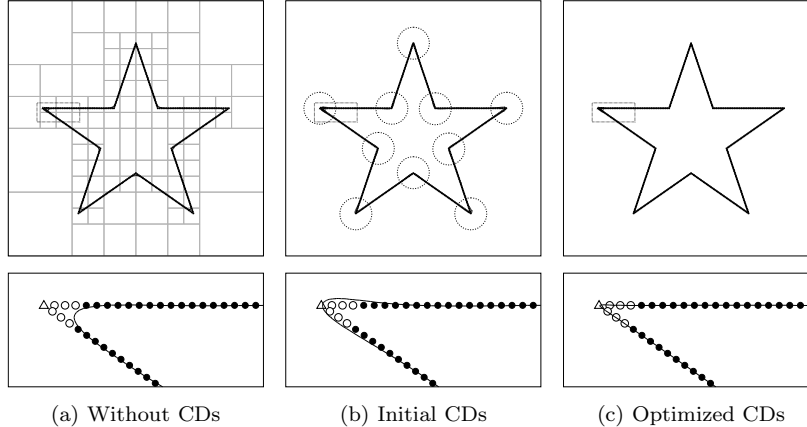


Fig. 2. Estimation of sharp features of a pentacle. The figures show graphs of the data points (solid dots), the implicit curve (solid line), and the control grid of the T-spline function f_0 . The bottom row shows the close-up view of the mid-left corner of the pentacle, where the clustered points (hollow dots) with large approximation errors, and the estimated feature points (hollow triangles) are also shown.

- (1) Project each data point \mathbf{p}_k onto the implicit curve $\Gamma(f_0)$, in order to get the closest point \mathbf{q}_k and compute the curvature radius ρ_k of $\Gamma(f_0)$ at \mathbf{q}_k . Robust methods for computing foot points on implicitly defined curves have been studied in [1].
- (2) Let Q be the set of all closest points \mathbf{q}_k where the distance $\|\mathbf{p}_k - \mathbf{q}_k\|$ exceeds a certain threshold ϵ_d and where the curvature radius satisfies $\rho_k < \epsilon_\rho$. Here, ϵ_d and ϵ_ρ are user-defined constants.
- (3) Use a region-growing-type process in order to cluster the points in Q into several groups $(Q_j)_{j=1, \dots, n_Q}$ (n_Q is the number of clustered groups). More precisely, for any two points \mathbf{q}_a and \mathbf{q}_b in the same group Q_j , we can always find a polygon $(\mathbf{q}_a = \mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_l = \mathbf{q}_b)$ with $\mathbf{q}_i \in Q_j$, $i = 0, \dots, l$ satisfying $\|\mathbf{q}_{i+1} - \mathbf{q}_i\| < \epsilon_q$, where ϵ_q is a user-specified constant.
- (4) Within each group Q_j , we identify the point \mathbf{q}_{0j} with the maximum distance value, i.e., $\|\mathbf{p}_{0j} - \mathbf{q}_{0j}\| = \max_{\mathbf{q}_i \in Q_j} \|\mathbf{p}_i - \mathbf{q}_i\|$. The corresponding data $(\mathbf{p}_{0j})_{j=1, \dots, n_Q}$ are the estimated sharp feature points.

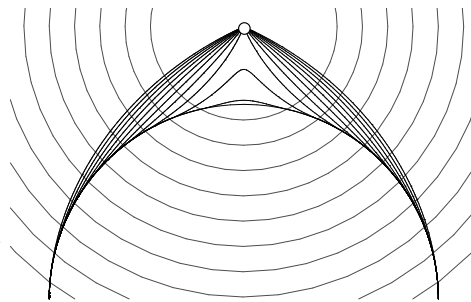


Fig. 3. Implicitly defined curves obtained by combining a circle and CD functions with different values of the radius r .

During this procedure we use three user-specified constants: ϵ_d , ϵ_ρ and ϵ_q . A natural choice of ϵ_d is the expected accuracy of the fitting method, i.e., the tolerated fitting error. The value of the curvature radius ϵ_ρ has the same order of magnitude. Finally, the value of ϵ_q , which is needed to form clusters of points, is chosen as $\epsilon_q = l_T$, where l_T is the diameter of the cells at the finest level of the T-mesh.

Each feature point \mathbf{p}_{0j} is now used to create a CD function

$$g_j(\mathbf{x}) = g_{a_j, r_j, \mathbf{c}_j}(\mathbf{x}), \quad j = 1, \dots, n_Q, \quad (2)$$

with the following initial values of the parameters:

$$\mathbf{c}_j = \mathbf{p}_{0j}, \quad r_j = l_T, \quad \text{and} \quad (3)$$

$$a_j = -f_0(\mathbf{c}_j)/r_j^2. \quad (4)$$

Consequently, the support of g_j is contained in a certain neighborhood of cells in the T-mesh to the cell containing \mathbf{p}_{0j} . Moreover, the point $\mathbf{c}_j = \mathbf{p}_{0j}$ is contained in $\Gamma(f_0 + g_j)$ and the implicitly defined curve has a singular point there.

By combining the CD functions $(g_j)_{j=1,2,\dots,n_Q}$ with the T-spline f_0 , we define the new curve $\Gamma(f)$ as the zero set of the sum

$$f = f_0 + g \quad \text{where} \quad g = \sum_{j=1}^{n_Q} g_j. \quad (5)$$

As a simple example, Fig. 3 shows the curves which are obtained by adding a CD function to the equation of the unit circle. Depending on the choice of the radius r , which is visualized by the grey circles, we obtain a family of curves. Note that this family contains the tangents of the circle passing through the center \mathbf{c} . We mention two properties of these curves:

- (1) If f_0 is a quadratic polynomial and the tangent of $\Gamma(f_0)$ at one of its intersections \mathbf{p} with the boundary circle of $\text{supp}g_j$ passes through the center \mathbf{c} , then $\Gamma(f_0 + g_j)$ contains the line segment connecting \mathbf{p} and \mathbf{c} .

- (2) If the radius parameter is below a certain threshold (see below), then the implicitly defined curve splits into a regular part and a isolated point at the center \mathbf{c} .

Lemma 1. *If the radius r_j satisfies*

$$r_j > \frac{2f_0(\mathbf{c}_j)}{\|\nabla f_0(\mathbf{c}_j)\|}, \quad (6)$$

then the $\Gamma(f_0 + g_j)$ has the corner \mathbf{c}_j with two different tangent directions. If the radius is smaller than the right-hand side, then \mathbf{c}_j is an isolated point.

Proof: The slope of the CD function g_j at its apex \mathbf{c}_j equals $-2a_j r_j$, and the slope of the T-spline is $\|\nabla f_0\|$. If the slope of the CD function at its apex is smaller than the slope of the T-spline, then we obtain a corner with two different tangent directions. The inequality (6) is now obtained by using (4). \square

The value of $|f_0|/\|\nabla f_0\|$, which is sometimes called the Sampson distance [12], is an estimate of the distance of a point from $\Gamma(f_0)$. The bound in (6) is approximately equal to twice the distance between \mathbf{c}_j and $\Gamma(f_0)$.

Remark 1. We assume that the support of each CD functions does not contain the center of any other CD function. This can always be guaranteed by decreasing the values of the radii r_j .

3.3. Optimizing the parameters of the CD functions

The parameter values controlling the CD functions g_j will now be optimized, in order to reduce the least-squares error

$$E = \sum_{k=1}^n \|\mathbf{d}_k\|^2 \quad \text{with} \quad \mathbf{d}_k = \mathbf{p}_k - \mathbf{q}_k \quad (7)$$

between the data points \mathbf{p}_k and the associated closest points \mathbf{q}_k on $\Gamma(f)$.

We use an evolution process in order to find optimal values of the parameter values controlling the CD functions. More precisely, we assume that the parameters r_j , a_j and \mathbf{c}_j controlling the CD functions depend on a time parameter τ . Starting from the initial values, we evolve them according to a differential equation.

The parameters r_j (influence radius of the CD function) and \mathbf{c}_j (position of the vertex) are independent parameters. The value of a_j , however, is chosen according to (4). Consequently, the implicitly defined curve $\Gamma(f)$ has always vertices at \mathbf{c}_j , $j = 1, \dots, n_Q$.

The evolution is to move the closest points \mathbf{q}_k towards the data points \mathbf{p}_k . We consider two cases:

Case 1: $\mathbf{q}_k \neq \mathbf{c}_j$, i.e., the closest point \mathbf{q}_k is different from all centers \mathbf{c}_j . In this situation, the *normal velocity* of the implicitly defined curve at \mathbf{q}_k is to satisfy

$$\dot{\mathbf{q}}_k \cdot \vec{\mathbf{n}}_k \approx \mathbf{d}_k \cdot \vec{\mathbf{n}}_k, \quad k = 1, \dots, n, \quad (8)$$

where $\vec{\mathbf{n}}_k = \nabla f(\mathbf{q}_k) / \|\nabla f(\mathbf{q}_k)\|$ is the unit normal of $\Gamma(f)$ and the dot “ \cdot ” indicates the differentiation with respect to the time variable τ . By a short computation, this can be shown to be equivalent to

$$-\frac{\dot{f}(\mathbf{q}_k)}{\|\nabla f(\mathbf{q}_k)\|} \approx \mathbf{d}_k \cdot \vec{\mathbf{n}}_k, \quad k = 1, \dots, n \quad (9)$$

Case 2: $\mathbf{q}_k = \mathbf{c}_j$, i.e., the closest point \mathbf{q}_k is one of the centers. Since the implicitly defined curve has a vertex at \mathbf{c}_j , we cannot consider the normal velocity. Instead, the velocity vector of the vertex \mathbf{c}_j is to satisfy

$$\dot{\mathbf{c}}_j \approx \mathbf{d}_k. \quad (10)$$

In general, it is impossible to satisfy all conditions (9) and (10) exactly. Instead, we adopt a least-squares approach and try to satisfy them approximately by minimizing the objective function

$$G = \sum_{\substack{k=1, \dots, n \\ \mathbf{q}_k \neq \mathbf{c}_j}} \left(\frac{\dot{f}(\mathbf{q}_k)}{\|\nabla f(\mathbf{q}_k)\|} + \mathbf{d}_k \cdot \vec{\mathbf{n}}_k \right)^2 + \sum_{\substack{k=1, \dots, n \\ j=1, \dots, n_Q \\ \mathbf{q}_k = \mathbf{c}_j}} (\dot{\mathbf{c}}_j - \mathbf{d}_k)^2 \rightarrow \text{Min.} \quad (11)$$

Note that we optimize only the parameters controlling the CD functions g_j , while the T-spline coefficients remain unchanged. Hence,

$$\dot{f} = \sum_{j=1}^{n_Q} \nabla g_j(\mathbf{c}_j) \cdot \dot{\mathbf{c}}_j + \frac{\partial g_j}{\partial r_j} \dot{r}_j + \frac{\partial g_j}{\partial a_j} \left(\frac{-\nabla f_0(\mathbf{c}_j) \cdot \dot{\mathbf{c}}_j}{r_j^2} + \frac{2f_0(\mathbf{c}_j)\dot{r}_j}{r_j^3} \right) \quad (12)$$

By substituting (12) into (11) we obtain a non-negative definite quadratic function of the unknowns $\dot{\mathbf{c}}_j$ and \dot{r}_j , $j = 1, \dots, n_Q$. The solution to (11) is obtained by solving a sparse linear system of equations, which defines the evolution of the CD functions. For any values of the parameters \mathbf{c}_j and r_j , we compute its time derivatives by solving the linear system and use a simple Euler method to integrate the path of the evolution.

The step size is determined with the help of the (normal) velocities. More precisely, we assume that the user specifies the maximum allowed displacement D of a point and choose the step size h such that the inequalities

$$\frac{|\dot{f}(\mathbf{q}_k)|}{\|\nabla f(\mathbf{q}_k)\|} h \leq D \quad \text{if } \mathbf{q}_k \neq \mathbf{c}_j, \quad \text{and} \quad \|\dot{\mathbf{c}}_j\| h \leq D \quad (13)$$

$(k = 1, \dots, n; j = 1, \dots, n_Q)$ are satisfied. The evolution continues until the approximation error defined in (7) is no longer reduced.

According to Lemma 2 of [2], the evolution process defined by (11) produces a sequence of curves which converges to a local minimum of E , see (7), and the Euler steps with step size h are equivalent to Gauss–Newton iterations with the same step size, provided that the second case (i.e., the second sum in (11)) is not present. Moreover, the proof of that Lemma can even be extended to the situation where the second case is present. Consequently, the evolution defined by (9) can be shown to be equivalent to Gauss–Newton iterations for minimizing E .

- Remark 2.** (1) The evolution has to maintain the property that the support of each CD function does not contain the center of any other CD function. Under this assumption, the computation splits into n_Q independent optimization problems which can be dealt with separately. If such a ‘collision’ between two CD functions takes place, then one should modify the objective function G by adding a term which penalizes the collision, or add suitable inequality constraints to the problem. Alternatively, the parameters controlling two colliding CD functions can be optimized simultaneously, but (4) can then no longer be used.
- (2) One may also simultaneously optimize both the parameters controlling the CD functions g_j and the T-spline coefficients of f_0 . In order to prevent the linear system from being ill-posed, a regularization method, such as a Levenberg-Marquardt strategy, has to be used. In order to avoid additional branches of the implicitly defined curve, the T-spline should be constrained to stay relatively close to the initial one (with respect to a suitable norm, e.g., in the coefficient space).
- (3) In practice, in order to improve the robustness of the method, it is helpful to optimize only the radii r_j first and to optimize all parameters \mathbf{c}_j, r_j later.

3.4. Experimental results

We present some examples to demonstrate the effectiveness of our method. All the given data points are contained in a square domain $(-1 \leq x, y \leq 1)$.

Example 1. The data points are sampled on a pentacle, see Fig. 2. The scalar T-spline function f_0 has 133 control points, and the implicit curve $\Gamma(f_0)$ without using of CD functions is shown in (a). The 10 CD functions are initialized for the detected sharp features in (b). After optimizing the parameters of the CD functions, the final result of $\Gamma(f_0 + g)$ is shown in (c).

Example 2. The data points are sampled from the contour of a Chinese character, see Fig. 4. The scalar T-spline function f_0 has 262 control

points, and the implicit curve $\Gamma(f_0)$ is shown in (a). The 11 CD functions are initialized for the detected sharp features in (b). After optimizing the parameters of the CD functions, the final result of $\Gamma(f_0+g)$ is shown in (c).

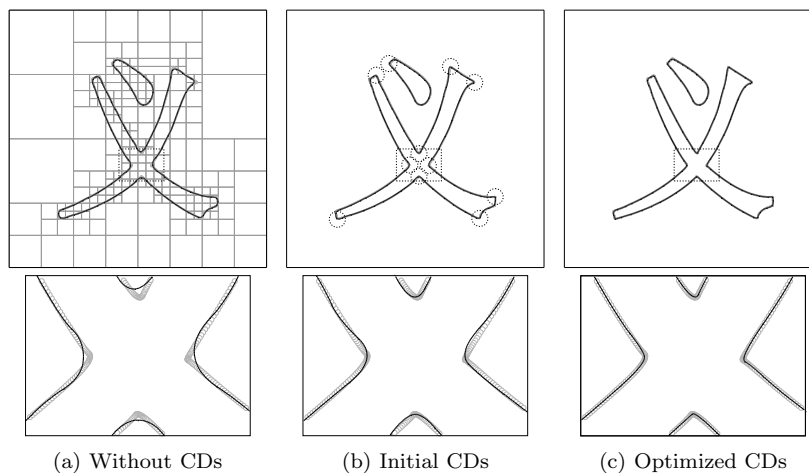


Fig. 4. Fitting a Chinese character. The bottom row shows the close-up view of the middle part of the character.

§4. Conclusion and future work

We have shown how to detect and represent sharp features with corner detector (CD) functions for implicit curve reconstruction from unorganized data points. These functions can be combined with a smooth implicit representation, such that the sharp features of the object can be well represented, without having to estimate the normal orientation at each point. The parameters controlling the CD functions are obtained by a local optimization procedure. Consequently, the relative computational costs to that of generating the function f_0 , which is often obtained from a global computation, are small.

As a matter of future research, we plan to extend our results to the case of edge detector functions in 3D.

Acknowledgment. H. Yang was supported by the European Union through its Marie Curie program (IIF) , project no. 22073 (ISIS).

References

1. M. Aigner and B. Jüttler, Robust computation of foot points on implicitly defined curves, in: M. Dæhlen, K. Mørken, and L. Schumaker, eds., *Math. Meth. for Curves and Surfaces*, 1–10. Nashboro Press, 2005.

2. M. Aigner, Z. Šír and B. Jüttler, Least-squares approximation by Pythagorean hodograph spline curves via an evolution process, in: M.-S. Kim, K. Shimada, eds., *GMP 2006*, Springer LNCS vol. 4077, 45–58.
3. J.C. Carr et al., Reconstruction and representation of 3D objects with radial basis functions, *Proc. Siggraph'01*, 67–76, ACM, New York.
4. R. Cartwright, V. Adzhiev, A. Pasko, Y. Goto and T.L. Kunii, Web-based shape modeling with HyperFun, *IEEE CG&A* **25** (2005), 60–69.
5. V. Caselles, R. Kimmel and G. Sapiro, Geodesic active contours, *Int. J. of Computer Vision* **22** (1997), 61–79.
6. H. Q. Dinh, G. Slabaugh and G. Turk, Reconstructing surfaces using anisotropic basis functions, in: *Proc. ICCV*, 606–613, 2001.
7. B. Jüttler and A. Felis, Least-squares fitting of algebraic spline surfaces, *Advances in Computational Mathematics* **17** (2002), 135–152.
8. Y. Ohtake, et al., Multi-level partition of unity implicits, *ACM Transactions on Graphics* **22** (Siggraph'03), 463–470.
9. G. Pasko, A. Pasko and T. Kunii, Bounded blending for function-based shape modeling, *IEEE CG & A* **25** (2005), 36–45.
10. A. Raviv and G. Elber, Three dimensional freeform sculpting via zero sets of scalar trivariate functions, in: *Proc. Solid Modeling and Applications*, ACM 1999, 246–257.
11. T.W. Sederberg, J. Zheng, A. Bakenov and A. Nasri, T-splines and T-NURCCs, *ACM Transactions on Graphics* **22** (Siggraph'03), 477–484.
12. P. Sampson, Fitting conic sections to very scattered data: An iterative improvement of the Bookstein algorithm, *Computer Graphics and Image Processing* **18** (1982), 97–108.
13. H. Wendland, *Scattered data approximation*, Cambridge University Press, 2005.
14. 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: *Proc. SMI'06*, 247–252, IEEE 2006. An extended version is available as FSP report at www.ig.jku.at.
15. Z. Yang, J. Deng and F. Chen, Fitting unorganized point clouds with active implicit B-spline curves, *The Visual Comp.* **21** (2005), 831–839.
16. H.-K. Zhao, S. Osher and R. Fedkiw, Fast surface reconstruction using the level set method, in: *Proc. 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 194–201, Vancouver, 2001.

Huaiping Yang and Bert Jüttler

Institute of Applied Geometry, Johannes Kepler University

Altenberger Str. 69, 4040 Linz, AUSTRIA

{yang.huaiping,bert.juettler}@jku.at, <http://www.ag.jku.at>