Vis Comput (2010) 26: 801–811
DOI 10.1007/s00371-010-0476-4

ORIGINAL ARTICLE

An additional branch free algebraic B-spline curve fitting method

Mingxiao Hu · Jieqing Feng · Jianmin Zheng

Published online: 14 April 2010

Abstract Algebraic curve fitting based on the algebraic distance is simple, but it has the disadvantage of inclining to a trivial solution. Researchers therefore introduce some constraints into the objective function in order to avoid the trivial solution. However, this often causes additional branches. Fitting based on geometric distance can avoid additional branches, but it does not offer sufficient fitting precision. In this paper we present a novel algebraic B-spline curve fitting method which combines both geometric distance and algebraic distance. The method first generates an initial curve by a distance field fitting that takes geometric distance as the objective function. Then local topology-preserving calibrations based on algebraic distance are performed so that each calibration does not produce any additional branches. In this way, we obtain an additional branch free fitting result whose precision is close to or even better than that produced by purely algebraic distance based methods. The adopted precision criterion is the geometric distance error rather than the algebraic one. In addition, we find a calibration fatigue phenomenon about calibrating strategy and propose a hybrid mode to solve it.

M. Hu · J. Feng (✉)
State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310058, China
e-mail: jqfeng@cad.zju.edu.cn

Present address:
M. Hu
College of Physical & Electronic Information Engineering, Wenzhou University, Wenzhou 325035, China

J. Zheng
School of Computer Engineering, Nanyang Technological University, Singapore 639798, Republic of Singapore

Keywords Curve reconstruction · Curve fitting · Algebraic B-spline curve · Additional branch · Geometric distance · Distance field · Local calibration

1 Introduction

With the rapid development of 3D scanning or measurement devices, the acquisition of 3D geometric shape becomes more convenient and the cost becomes lower. To process point cloud or scattered data points, many techniques have been studied. Curve fitting is a fundamental one that is widely applied in data reconstruction [1], computer graphics [2], computer vision and pattern recognition [3], etc. In this paper we study curve fitting using algebraic B-spline curves.

It is well-known that algebraic curves have a wide representing domain and B-spline has many nice properties (such as local control, inherent continuity, etc.) for shape description and modeling. Therefore in this paper we adopt algebraic curves based on uniform B-splines with a fixed order and a fixed basis function number to address the planar point cloud fitting problem. Let the planar point cloud be denoted by $\{\mathbf{P}_k\}_{k=1}^N$, where $\mathbf{P}_k = (x^{(k)}, y^{(k)})$ are 2D points called the raw points. We want to find a K-order algebraic curve C in terms of the tensor product B-spline:

$$f(x, y) = \sum_{t=1}^{n} \sum_{s=1}^{m} c_{st} N_{t,K}^{[x]}(x) N_{s,K}^{[y]}(y) = 0 \tag{1}$$

to fit the point cloud $\{\mathbf{P}_k\}_{k=1}^N$, where $N_{t,K}^{[x]}(x)$, $N_{s,K}^{[y]}(y)$ are the uniform B-spline basis functions in the X and Y directions respectively. For the sake of simplicity, we denote them by $N_t(x)$, $N_s(y)$ instead, omitting the direction flags within square brackets and order K in subscript. The goal of the

fitting problem is to seek the "best" tensor product coefficients $\mathbf{c} = \{c_{st}\}$ that minimize a certain objective function. Each coefficient c_{st} is also called a Greville ordinate whose corresponding abscissa is the center point of the support of basis function $N_t(x)N_s(y)$ [4].

There are two conventional fitting methods. The first one is to minimize the following objective function:

$$\varphi(\mathbf{c}) = \sum_{k=1}^{N} f(\mathbf{P}_k)^2 = \sum_{k=1}^{N} f(x^k, y^k)^2 \tag{2}$$

This is the so-called algebraic distance method. However, if the algebraic distance fitting method is not properly specified, it tends to converge to a trivial solution where all the tensor product coefficients are zero. Jüttler minimized the above objective function by imposing normal constraints and a smoothing item to avoid the trivial solution. Another disadvantage of the algebraic distance fitting method is the frequent appearance of additional branches. Because the relation between the coefficients and the curve topology is not obvious, the additional branches are difficult to be avoided. The additional branches may cause unfavorable problems such as branch separation, correct branch selection, more complex footpoint calculation, geometric distance distortion, etc. In some cases, we have to reduce fitting precision to avoid additional branches, which becomes quite an indispensable treatment in curve fitting problem.

The second method is the geometric distance fitting that minimizes the following objective function:

$$\psi(\mathbf{c}) = \sum_{k=1}^{N} d(\mathbf{P}_k, C)^2 \tag{3}$$

where

$$\begin{aligned} d(\mathbf{P}_k, C) &= \min_{\mathbf{a}\in C} d(\mathbf{P}_k, \mathbf{a}) \\ &= \min_{f(x,y)=0} \left[(x^{(k)} - x)^2 + (y^{(k)} - y)^2\right]^{1/2} \end{aligned}$$

In general, the geometric distance based fitting approach offers following advantages [5]:

- it has geometry invariability;
- fitting error complies with visional intuition;
- unbiased model parameters;
- and the fitting procedure seldom ends with an unintended model feature.

Therefore, it is strongly suggested to take geometric distance as the objective function in curve fitting. The substantial difficulty in the geometric distance method is the calculation of precise Euclidean distance from a point to a curve, which has no analytic expression, even for a quadric curve. Therefore an iteration method is often adopted to obtain a numerical solution instead, which is time-consuming.

Whatever method we adopt, the fitting error is defined as the average of the squared geometric distance of raw points to the curve:

$$G = \frac{1}{N} \sum_{k=1}^{N} d(\mathbf{P}_k, C)^2 \tag{4}$$

In this paper, the above error G is adopted as a fair precision comparison standard to assess different fitting methods.

This paper presents an additional branch free fitting method (ABFF, for short) for planar closed raw points with algebraic B-spline curves, where the order of B-spline is 3 or 4. Section 2 briefly reviews the related works on implicit curve fitting. Section 3 describes the method in detail. Section 4 gives the experimental results, which shows our fitting method not only has high fitting precision, but also is free of additional branches. Finally, conclusions are drawn and future research problem is indicated.

2 Related work

Implicit curve fitting based on algebraic distance has been researched since 1987. Pratt proposed a "simple fit" method which keeps a special coefficient unchanged by linear normalization [6]. However, the process is not a geometric one, since it depends on the selection of coordination system. Later, Taubin added a gradient constraints to the objective function and converted the curve fitting problem into a general eigenvalue problem [7]. Jüttler et al. adopted algebraic curves and surfaces in terms of the tensor product B-spline to fit 2D and 3D point clouds respectively [8].

Ahn et al. summarized the disadvantage of algebraic distance fitting, i.e., the error definition does not comply with the measurement guidelines. They also asserted that geometric distance fitting is a feasible solution to overcome above disadvantages caused by algebraic distance [9]. Unfortunately, the geometric distance is highly non-linear about the model parameters compared with algebraic distance $f(x, y)$ as in (1). The analytical expression of geometric distance is astonishingly complex or even impossible. Computational complexity of finding accurate geometric distance is very high. Many efforts have been spent to calculate approximate geometric distance in either a global way or local way [10, 11]. Aigner and Jüttler proposed a robust method to compute the footpoint via shrinking circles [12].

Due to the analytical computation difficulty, geometric distance fitting solution unanimously employs iteration approaches. Kristek and Anton worked on 2D straight line fitting based on weighted geometric distance. They reduce the line fitting to a one-dimensional searching problem. The real geometric distance is regarded as a special case where the two direction weights are equal [13]. Ahn et al. worked

on orthogonal fitting of quadric primitives, such as circle, sphere, ellipse, hyperbola and parabola. They employed orthogonal contacting conditions and Jacobi matrix of coordinate transformation and proposed a fast convergent iteration method [9]. Wang et al. fitted a parametric B-spline curve to planar point cloud fitting based on a second order approximation of squared distance. They compared their square distance minimization methods (SDM) with the point distance iteration method (PD) and the tangent distance iteration method (TD), declared that SDM's iteration speed is faster than those of PD and TD [14].

Sederberg et al. studied approximate implicitization using monoid curves and surfaces. The use of monoid curves and surfaces helps eliminate undesirable additional branches ("phantom" branches) which were normally associated with implicitization process [15].

To improve algebraic B-spline curve (surface) method, Jüttler et al. proposed a method by adding norm constraints and a smoothing item into algebraic distance objective function [16]. It does avoid the trivial solution, but often produces additional branches. Their objective function is defined as

$$g(\mathbf{c}) = \sum_{k=1}^{N} \left[f(x^{(k)}, y^{(k)}) \right]^2 + \omega_1 M(\mathbf{c}) + \omega_2 G(\mathbf{c}) \tag{5}$$

where $M(\mathbf{c}) = \sum_{k=1}^{N} \|\nabla f(x^{(k)}, y^{(k)}) - \mathbf{n}_k\|^2$ is a norm constraint item, $\mathbf{n}_k = (n_x^{(k)}, n_y^{(k)})$ is the estimated unit outer norm at $\mathbf{P}_k = (x^{(k)}, y^{(k)})$,

$$G(\mathbf{c}) = \iint_{\Omega} \left(f_{xx}^2(s, y) + 2 f_{xy}^2(x, y) + f_{yy}^2(x, y) \right) dx\, dy$$

is a smoothing item, ω_1 is the weight of the norm constraint, usually between 0.00001 and 1, and ω_2 is the weight of the smoothing item, usually between 0 and 0.01. For a given point cloud, additional branches can be eliminated by specifying proper weights ω_1 and ω_2. However, tuning the weights to eliminate additional branches is an interactive and time-consuming procedure and may result in precision loss. In general, the smaller ω_2 is, the more frequently additional branches appear. For the extreme case when $\omega_2 = 0$, the fitting results for almost all point clouds even a circle shape have additional branches in our test. On the contrary, when ω_2 becomes large, the fitting curve become more smooth and free of additional branches, but the fitting precision becomes poor. Thus determining appropriate weights that just eliminate additional branches without loss of fitting precision is still an open question. Yang et al. extended above Jüttler's method to the geometric distance case [17]. They locally approximate the geometric distance linearly and solve the non-linear objective function via trust region algorithm. However the footpoint computation is a time-consuming.

Li et al. proposed a fitting method using algebraic B-spline curves based on a signed distance field [18]. It first creates a high quality distance field from the raw point set and then fits to this distance field, thus fits to the point cloud indirectly. Experiments show that the field fitting method can effectively control the fitting curve's topology and guarantee its shape not deviated from the underlying raw point set. Therefore it never produces any additional branches. However, its fitting precision is much lower than Jüttler's method.

3 Additional branch free algebraic B-spline curve fitting

3.1 Direct distance field fitting

In the proposed method, a signed distance field that surrounds the raw point set is constructed. The sampling density of the signed distance field should be high enough to depict the shape (potentially determined by the raw point set) faithfully [18]. Then its objective function for the direct distance field fitting is defined as:

$$Err(\mathbf{c}) = \sum_{j=1}^{N_x} \sum_{i=1}^{N_y} (f(\mathbf{p}_{ij}) - d_{ij})^2 = \sum_{j=1}^{N_x} \sum_{i=1}^{N_y} (f(x_j, y_i) - d_{ij})^2 \tag{6}$$

where N_x and N_y are the distance field sampling densities along the X and Y directions respectively, d_{ij} is the signed distance value at sampling point $\mathbf{p}_{ij} = (x_j, y_i) = (x_0 + (j-1)\Delta x, y_0 + (i-1)\Delta y)$ with (x_0, y_0) the lower-left point coordinates of the distance field, Δx, Δy are the sampling steps along the X and Y directions respectively. In our method, the sampling densities along the two directions are chosen to guarantee that: (1) both Δx and Δy are not greater than a half of the minimal estimated curvature radius from the raw point set [18]; (2) there are at least two sampling points in every B-spline knot interval, i.e., $N_x = 2(n - K + 1)$, $N_y = 2(m - K + 1)$.

Note that the above field fitting converts a geometric distance fitting to an algebraic distance fitting. Supposing that the fitting result $f(x, y)$ exactly interpolates the sampled distance values, i.e., $f(x_j, y_i) = d_{ij}$ for all i, j, then the algebraic distance resulting from fitting the sampled signed distance field should well approximate geometric distance resulting from geometric distance fitting the raw point set. In other words, the direct distance field fitting is an indirect approach to achieve geometric distance fitting effect with algebraic distance fitting complexity.

Unfortunately, experimental results show that the fitting precision of direct distance field fitting is poor due to the following reasons.

(1) The signed distance value d_{ij} is approximate rather than accurate.
(2) The discrete signed distance field is approximated rather than interpolated.
(3) The signed distance field of a planar curve is highly nonlinear in general so that it can not be well approximated via only polynomial fitting.

As indicated in the paper [18], the algebraic B-spline curve obtained via the direct distance field fitting is free of additional branches. Although its fitting precision is not satisfied, it is a very good initial start for further refinements via following local coefficients calibration.

3.2 Local coefficients topology-preserving calibration

Now we refine the algebraic B-spline curve obtained via direct distance field fitting so that it approaches the optimal solution as much as possible. The refinement is achieved via calibrating local coefficients $\{c_{st}\}$ of the algebraic B-spline curve, whilst preserving the topology of curve. We should keep the curve free of additional branches in each calibration step.

Before calibrating the coefficients, we describe two obvious facts. If geometric distance is adopted as the objective function, which has no analytic expression, the Gauss–Newton iteration cannot be applied trivially. If algebraic distance is adopted as the objective function, the calibrated curve will tend to contain additional branches. According to these observations, we propose a local coefficient calibration scheme whilst preserving the curve topology as follows.

(1) The algebraic distance of initial field fitting result $f(x, y)$ is an approximation of the geometric distance. Thus the special algebraic distance could be adopted to define our calibration objective function.
(2) The Gauss–Newton scheme is adopted to calibrate the coefficients locally.
(3) Only the calibration that do not change the curve topology are fully accepted and the others will be modified or discarded.

In each calibration, one or more coefficients are adjusted in order to reduce the special algebraic distance objective function. Here a coefficient is over-adjusted if its sign is changed before and after calibration. The over-adjusted coefficients may influence the curve topology potentially. Thus its corresponding calibration should be modified and discarded. If all coefficients are refined in each calibration, i.e. a global scheme, there always exist a coefficients over-adjusted. The calibration is discarded or modified and only local optimal solution is obtained. Thus we prefer the local calibration scheme since most calibrations are valid.

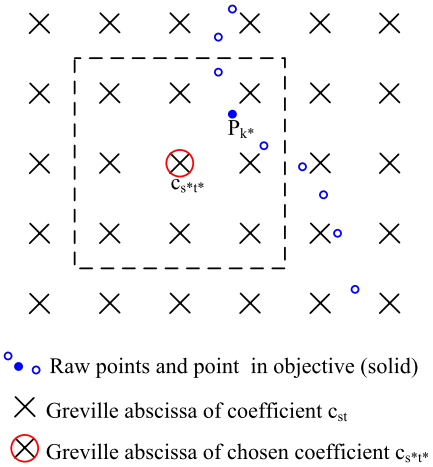


Fig. 1 1-1 mode calibration: the algebraic distance of one raw point P_{k^*} is minimized by calibrating one coefficient $c_{s^*t^*}$. The *dashed square* is the influence region of $c_{s^*t^*}$. Because P_{k^*} lies in the influence region of $c_{s^*t^*}$, the calibration can always reach the optimization, i.e., the algebraic distance of P_{k^*} becomes 0

The calibration is performed for each raw point $\mathbf{P}_{k^*} = (x^{(k^*)}, y^{(k^*)})$ so that the following algebraic distance to implicit curve $f(x, y)$ is optimized to be zero.

$$f\left(x^{(k^*)}, y^{(k^*)}\right) = \sum_{s=1}^{m}\sum_{t=1}^{n} c_{st} N_s\left(y^{(k^*)}\right) N_t\left(x^{(k^*)}\right) \tag{7}$$

If only one coefficient $c_{s^*t^*}$ is selected, its adjustment can be computed as:

$$c'_{s^*t^*} = c_{s^*t^*} - \frac{f(x^{(k^*)}, y^{(k^*)})}{N_{s^*}(y^{(k^*)}) N_{t^*}(x^{(k^*)})} \tag{8}$$

To make the calibration valid, the raw point $\mathbf{P}_{k^*}$ must lie in the influence region of the coefficient $c_{s^*t^*}$ so that $N_{s^*}(y^{(k^*)})\ N_{t^*}(x^{(k^*)}) \neq 0$. After $c_{s^*t^*}$ is calibrated to $c'_{s^*t^*}$, if $c_{s^*t^*}c'_{s^*t^*} < 0$, we keep the original value $c_{s^*t^*}$ unchanged, i.e. $c'_{s^*t^*} = c_{s^*t^*}$, or set the new coefficient to zero, i.e. $c'_{s^*t^*} = 0$. The former process is to discard the calibration and the later one is to modify calibration. This calibration scheme is called the 1-1 mode, which is illustrated in Fig. 1.

Of course, we can also adjust only one coefficient $c_{s^*t^*}$ to minimize the following total algebraic distances of all raw points in the influence region of the coefficient $c_{s^*t^*}$, as illustrated in Fig. 2(a):

$$D = \sum_{k\in K}\left[(c'_{s^*t^*} - c_{s^*t^*}) N_{s^*}(y^{(k)}) N_{t^*}(x^{(k)}) + f(x^{(k)}, y^{(k)})\right]^2 \tag{9}$$

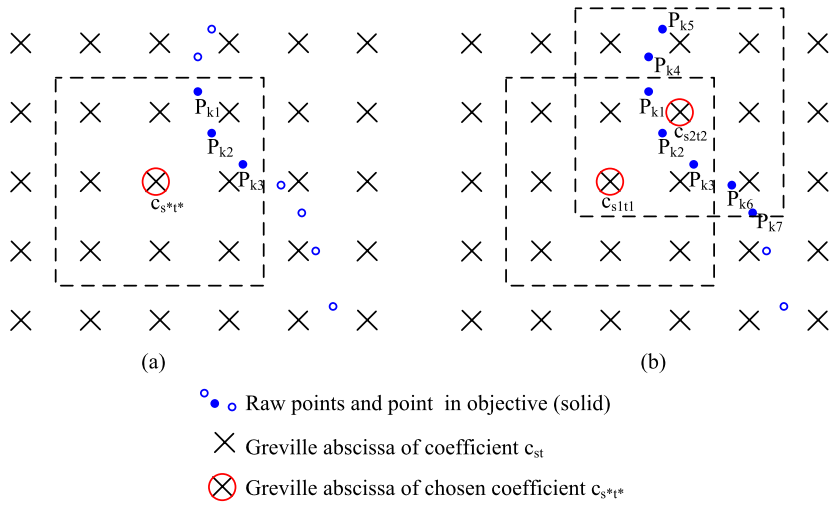


Fig. 2 1-N model and M-N mode calibrations. (**a**) 1-N mode: the algebraic distance squares sum of 3 raw points P_{ki} $(i = 1, 2, 3)$ in the influence region of $c_{s^*t^*}$ is minimized. The *dashed square* represents the influence region of $c_{s^*t^*}$. (**b**) M-N mode: the algebraic distance squares sum of 7 raw points in the joint influence regions of two coefficients $c_{s_1t_1}$ and $c_{s_2t_2}$ is minimized

where K is an index set of the raw points that lie in the influence region of $c_{s^*t^*}$. The minimization can be reached by solving equation $\frac{\partial D}{\partial c'_{s^*t^*}} = 0$. The result is

$$c'_{s^*t^*} = c_{s^*t^*} - \frac{\sum_{k\in K} f(x^{(k)}, y^{(k)}) N_{s^*}(y^{(k)}) N_{t^*}(x^{(k)})}{\sum_{k\in K} N_{s^*}(y^{(k)})^2 N_{t^*}(x^{(k)})^2} \quad (10)$$

If $c'_{s^*t^*}$ satisfies $c_{s^*t^*}c'_{s^*t^*} < 0$, we modify or discard it similar with that in 1-1 mode. We call this calibration the 1-N mode.

In general, we can adjust two or more coefficients simultaneously. It is called 2-N mode or M-N mode as illustrated in Fig. 2(b). Let $c_{s_1t_1}, \ldots, c_{s_Mt_M}$ be the calibrated coefficients. By using the Gauss–Newton method, $\Delta\mathbf{c} = (c'_{s_1t_1} - c_{s_1t_1}, \ldots, c'_{s_Mt_M} - c_{s_Mt_M})^{\mathrm{T}}$ should satisfy the normal equation

$$\mathbf{U}^{\mathrm{T}}\mathbf{U}\Delta\mathbf{c} = -\mathbf{U}^{\mathrm{T}}\mathbf{b} \quad (11)$$

where

$$\mathbf{U} = \begin{pmatrix} N_{s_1}(y^{(k_1)})N_{t_1}(x^{(k_1)}) & \cdots & N_{s_M}(y^{(k_1)})N_{t_M}(x^{(k_1)}) \\ \cdots & \cdots & \cdots \\ N_{s_1}(y^{(k_K)})N_{t_1}(x^{(k_K)}) & \cdots & N_{s_M}(y^{(k_K)})N_{t_M}(x^{(k_K)}) \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} f(x^{(k_1)}, y^{(k_1)}) \\ \vdots \\ f(x^{(k_K)}, y^{(k_K)}) \end{pmatrix}$$

By solving the normal equation, we can obtain coefficient calibrations. Finally, we modify or discard calibrations where at least one coefficient's sign is changed. Apparently, if $M = 1$, it degenerates to 1-N mode and the normal equation degenerates to a single-variable linear equation.

In above 1-N mode or M-N mode calibrations, the coefficient selection is critical to the calibration effects. According to our experiments, there are following typical selection methods (refer to Fig. 3).

(1) Selecting only one coefficient $c_{s^*t^*}$ corresponding to the raw point $\mathbf{P}_{k^*} = (x^{(k^*)}, y^{(k^*)})$. The Greville abscissa of the coefficient $c_{s^*t^*}$, i.e. center of support domain of the basis function, is determined as follows:
  (1a) (**1O***ut* case) the nearest point in a $\pi/2$ fan region which is emitted from $(x^{(k^*)}, y^{(k^*)})$ and whose center direction is coincided with the outer curvature normal of $f(x, y)$ at $(x^{(k^*)}, y^{(k^*)})$.
  (1b) (**1I***nner* case) the nearest point in a $\pi/2$ fan region which is emitted from $(x^{(k^*)}, y^{(k^*)})$ and whose center direction is coincided with the inner curvature normal of $f(x, y)$ at $(x^{(k^*)}, y^{(k^*)})$.
  (1c) (**1R***andom* case) a random point that influences the $f(x^{(k^*)}, y^{(k^*)})$.
  (1d) (**1H***ighest* case) the point that influences $f(x^{(k^*)}, y^{(k^*)})$ most.
(2) (**M***ultiple* case) Selecting all coefficients that have influences on $f(x^{(k^*)}, y^{(k^*)})$ except for the coefficients near the boundary of the influence region.
(3) (**2I***nner***O***uter* case) Selecting two coefficients: one from **1O** case and the other one from **1I** case.
(4) (**4I***nner***I***nner***O***uter***O***uter* case) Selecting four adjacent coefficients that influence $f(x^{(k^*)}, y^{(k^*)})$: two or less points lie inside of the curve $f(x, y)$ and the rest lie outside of the curve $f(x, y)$.
(5) (**4I***nner***O***ut***L***eft***R***ight* case) Choose four adjacent coefficients that influence $f(x^{(k^*)}, y^{(k^*)})$ and whose Greville abscissae surround the point $(x^{(k^*)}, y^{(k^*)})$.

3.3 Calibration strategy

To perform above calibrations, some important parameters should be determined. They are a proper calibration round number, a raw point calibrating order, a coefficient selection method, a local objective function and modifying-or-discarding decision of each round. A complete *calibration strategy* is as follows:

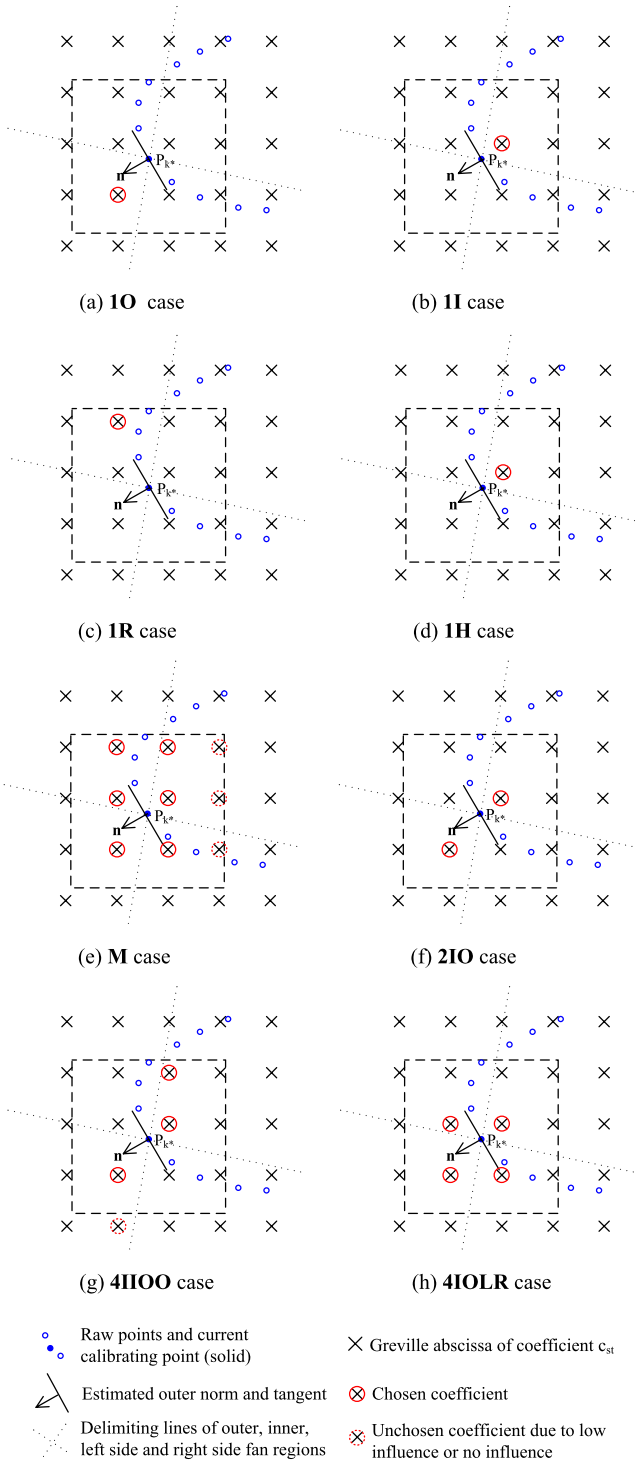


Fig. 3 Coefficient selection method: in (**e**) **M** case, 3 coefficients (*dash circled*) are not selected because they lie near the boundary of influence region of B-spline function. In (**g**) **4IIOO** method, one coefficient (*dash circled*) is not selected because it lies outside the influence region

(1) The calibration round number: how many Gauss–Newton iteration steps are adopted for the initial algebraic B-spline curve $f(x, y)$. According to our experiments, 6 is a suitable number.

(2) The calibrating order of raw points in each round. There are two options: a static error order and a dynamic error order. For the static error order option, we calibrate all raw points in a descending order of their algebraic distance errors; and for the dynamic error order option, we always calibrate the raw point with the highest algebraic distance error corresponding to currently calibrated curve $f(x, y)$.

(3) For each raw point, there are 8 coefficient selection methods. They are **1O**, **1I**, **1R**, **1H**, **2IO**, **2O**, **2I**, **4IIOO**, **4IOLR**, **M** cases, as shown in Fig. 3.

(4) The objective functions for locally calibration: in each calibration step, we can compute the algebraic distance error for the currently processing raw point or all influenced raw points.

(5) We modify or discard the calibration when there is at least one coefficient's sign changed.

The calibration strategy can be described as a five-tuple $\langle N, \mathrm{R}[N], \mathrm{C}[N], \mathrm{O}[N], \mathrm{D}[N]\rangle$, where N is the calibration round number, R[N], C[N], O[N] and D[N] are the raw points calibration order, the coefficient selection methods, the local objective function and the discarding-or-modifying calibration, respectively.

3.4 Fitting and calibration algorithm

Now we describe our fitting and calibration algorithm, which consists of the following steps:

(1) Perform the direct distance field fitting method to generate the initial algebraic B-spline curve;
(2) set the calibration strategy $\langle N, \mathrm{R}[N], \mathrm{C}[N], \mathrm{O}[N], \mathrm{D}[N]\rangle$;
(3) for ($i = 0$; $i < N$; i++) {

```
while (loop for raw points according to order R[i]) {
  choose B-spline coefficients c_{i1 j1}, ..., c_{iM jM} according to C[i];
  if (all selected coefficients were calibrated in current round)
    continue;
  calibrate c_{i1 j1}, ..., c_{iM jM} to c'_{i1 j1}, ..., c'_{iM jM} according to O[i];
  if (there exists c'_{ik jk} such that c'_{ik jk} c_{ik jk} < 0)
    discard or modify this calibration according to D[i]
}
}
```

Experiments show that if all round calibrations adopt the same strategy, e.g., "**4IIOO**" coefficients selection method + "discarding" topology-preserving method, then there will be many coefficient signs changed in the later calibrations. In this case, the calibrations will be discarded or modified,

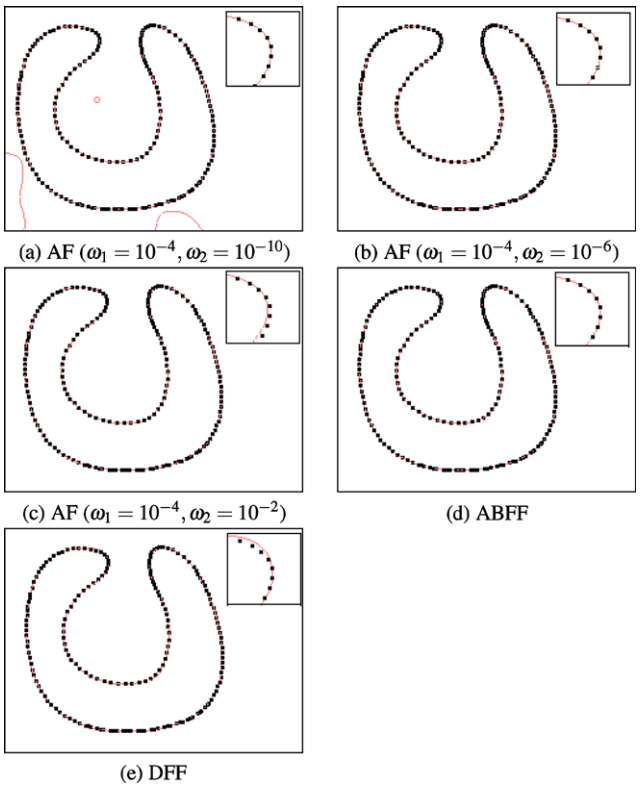


Fig. 4 Fitting CNCV2 data by using AF, ABFF and DFF

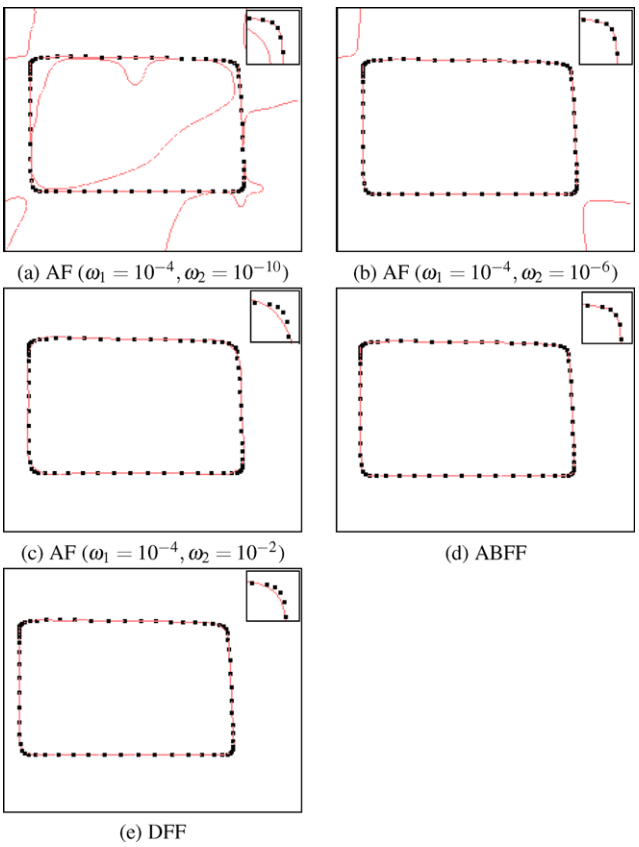


Fig. 5 Fitting rectangle data by using AF, ABFF and DFF

thus the fitting precision will remain almost without change. The phenomenon is called "calibration fatigue". Therefore, in practice we prefer to use different coefficient selection strategies in the successive rounds in order to rapidly improve the fitting precision.

4 Experimental results and discussion

The algorithm has been implemented on a PC with Intel P4 2.4 GHz CPU, 1.5 GB Memory and Windows XP. For the sake of comparison, the proposed additional branch free fitting method, Jüttler's method [16] and Li's distance field fitting method [18] are abbreviated as ABFF, AF, and DFF respectively. In ABFF, the calibration strategy is set as:

$< N = 6,$
R$[N]$ = dynamic algebraic error order,
C$[N]$ = {4IOLR, 4IIOO, 1R, 4IOLR, 4IIOO, 2IO },
O$[N]$ = all influenced raw points,
D$[N]$ = { discard, discard, modify, discard, discard, discard } $>$

4.1 Comparison with AF

4.1.1 Visual effects

Figures 4, 5, 6(a)–(d) are fitting results for three raw point sets visual by using AF and ABFF, where B-spline order is $K = 3$, tensor product coefficient numbers $m = n = 17$. For AF, normal constraints weight ω_1 is 10^{-4}, and smoothing term weight ω_2 is 10^{-10}, 10^{-6}, and 10^{-2} respectively. If the ω_2 is not set carefully in AF, there will be additional branches generated. However, our method is free of additional branches and shape parameter setting. Furthermore, the visual quality of fitting curves is similar for both AF and ABFF in the case of free of additional branches.

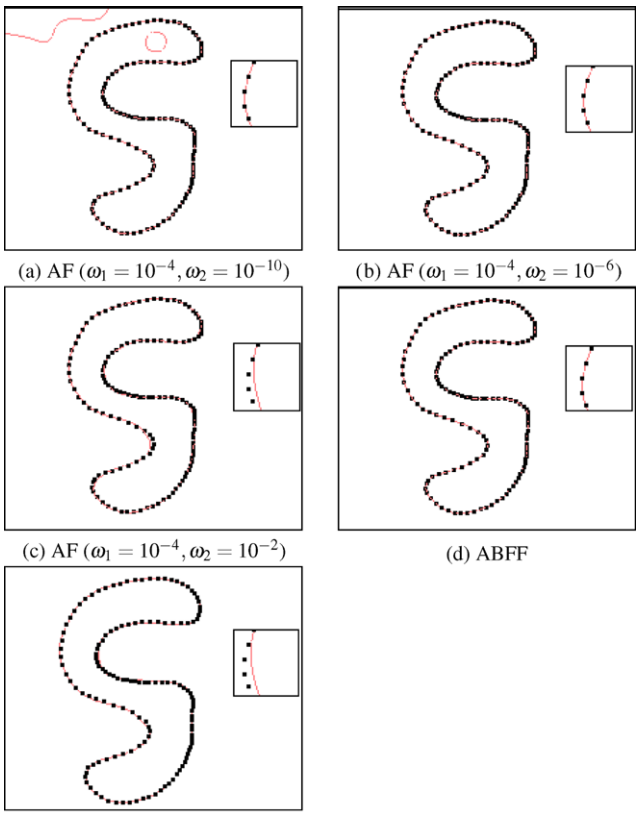
In these examples, AF always produces additional branches when $\omega_2 = 10^{-10}$. Figure 5(a) even contains many un-separable additional branches. Increasing ω_2 can eliminate additional branches. However, the fitting precision will drop with the increase of ω_2. As in Fig. 5, when ω_2 is 4.4×10^{-5}, AF just eliminates all additional branches, but its global average error is about ($\approx 1/0.163$) times of ABFF as shown in Table 1.

4.1.2 Fitting precision

For the given B-spline order $K = 3$ or $K = 4$, the fitting errors of ABFF are comparable to or better than that of AF when $\omega_2 = 10^{-6}$. Some errors of test data are even only $1/10$ of AF. When $\omega_2 = 10^{-8}$ in AF, although errors of some test data by ABFF are greater than those of AF, e.g., Rectangle, AF fitting results tend to contain additional branches

Table 1 Trade-off between AF precision and additional branches, where $K = 3$, $\omega_1 = 10^{-4}$. *GAG* represents average of geometric errors for 16 test raw point sets and C_{AB} represents the number of raw point sets that have additional branches

ω_2	10^{-2}	10^{-4}	$\mathbf{4.4\times10^{-5}}$	10^{-5}	10^{-6}	$\mathbf{1.2\times10^{-7}}$	10^{-7}	10^{-9}	10^{-11}
GAG_{ABFF} (10^{-6})					**0.392**				
GAG_{AF} (10^{-6})	242.4	4.320	**2.411**	1.069	0.556	**0.391**	0.381	0.261	0.238
GAG_{ABFF}/GAG_{AF}	0.002	0.091	**0.163**	0.367	0.705	**1.003**	1.029	1.502	1.647
C_{AB}	0	0	**0**	1	2	**5**	6	13	16



(a) AF ($\omega_1 = 10^{-4}, \omega_2 = 10^{-10}$) (b) AF ($\omega_1 = 10^{-4}, \omega_2 = 10^{-6}$)

(c) AF ($\omega_1 = 10^{-4}, \omega_2 = 10^{-2}$) (d) ABFF

(e) DFF

Fig. 6 Fitting Vers1 data by using AF, ABFF and DFF

(5 among 16 examples). When $\omega_2 = 10^{-4}$, the global average error of ABFF surpasses that of AF.

From Table 1, we can also investigate the overall fitting precision comparison for 16 test raw point sets between AF and ABFF. Let *GAG* be noted as global average error, and C_{AB} be noted as number of raw point sets with additional branches. It shows that if ω_2 is greater than 10^{-3}, the AF fitting precision is poor; and when ω_2 decreases to 1.2×10^{-7}, *GAG* of AF just equals to that of ABFF, but among 16 raw point sets there are 5 point sets containing additional branches, i.e. C_{AB}. When ω_2 decreases continuously, additional branches appear more and more frequently, whereas its *GAG* is only about 60% ($\approx$0.238/0.392) that of ABFF. Thus we assert that AF will suffer from additional branches problem seriously in order to improve its fitting precision. Furthermore, the precision has not enough room to be improved.

4.1.3 Fitting speed

The fitting time of AF is obviously faster than ABFF, especially when numbers of coefficients m(or n) is greater than 20. First the discrete distance field construction is very fast and occupies one small part of total fitting procedure. In the direct distance field fitting, it is performed direction-wise due the tensor product property of the algebraic B-spline curve. Finally, our calibrations are local operations. However, the AF should solve a large linear equation system whose size is $(mn)^2$. The detailed runtime statistic comparison is shown in Fig. 7.

4.2 Comparison with DFF

4.2.1 Visual effects

Three examples are shown in Figs. 4–6(d) and (e), where the B-spline order $K = 3$, tensor product coefficient number $m = n = 17$. As illustrated in the figures, although DFF free of additional branches, its fitting results apparently deviate from the raw point set. ABFF not only has no additional branches, but also always fits the raw point sets faithfully.

4.2.2 Fitting precision

The detailed fitting precision data can be found in the Appendix, where the error G is defined by (4). In the cases of $K = 3$ or $K = 4$, the fitting precision of ABFF is conspicuously better than that of DFF.

4.3 Calibration fatigue phenomenon

In ABFF calibration strategy, the coefficient selection method is hybrid. If a fixed coefficient selection method is adopted thoroughly in all calibration rounds, the fitting precision improvements will become smaller and smaller. Finally the improvement ratios approach 1.0. It is called "calibration fatigue". If we define "fatigue" as the case that the improvement ratio is less than 1.25, **1I** and **1H** will fatigue after the 1st round calibration, **1O**, **2IO**, **4IOLR** and **M** will

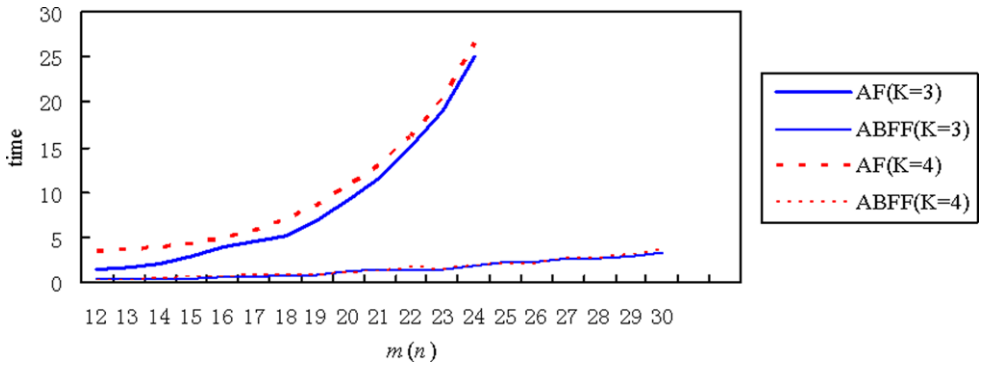


Fig. 7 Fitting time comparison between AF and ABFF: the tensor product coefficients number m and n ranged from 12 to 30. The data are total fitting times (sec) for 16 raw point sets

Table 2 Calibration fatigue phenomenon statistics: the precision improvement ratios are the average of 16 raw point set

Coefficient choosing method	Discard/Modify	Precision improving ratio r_i of round i					
		r_1	r_2	r_3	r_4	r_5	r_6
1O	Modify	3.221	1.658	1.099	1.048	1.019	1.016
1I	Modify	1.883	1.077	1.026	1.004	1.002	1.002
1H	Modify	2.553	1.134	1.092	1.033	1.007	1.012
1R	Modify	4.912	2.896	1.833	1.390	1.228	1.145
2IO	Discard	7.174	1.872	1.218	1.068	1.061	1.045
4IIOO	Discard	17.775	1.832	1.294	1.158	1.095	1.083
4IOLR	Discard	11.387	2.238	1.161	1.133	1.049	1.045
M	Discard	4.279	1.374	1.114	1.021	1.014	1.009
Hybrid mode		17.775	2.248	1.596	1.543	1.307	1.208

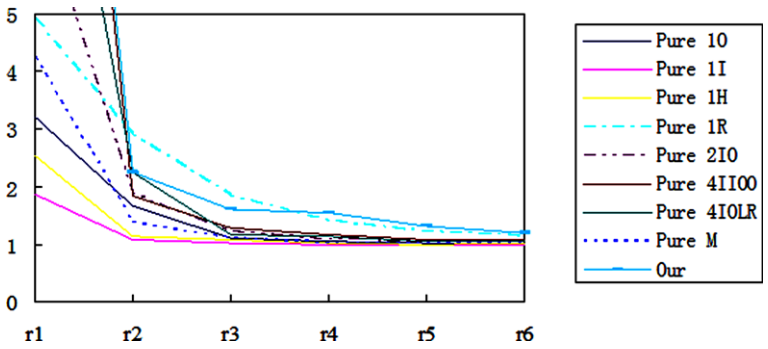


Fig. 8 Calibration fatigue phenomenon: the *horizontal axis* is the number of the calibration round and the *vertical axis* is the precision improvement ratio

fatigue after the 2nd round calibration, **4IIOO** will fatigue after the 3rd round calibration, **1R** will fatigue after the 4th round calibration. By adopting our hybrid modes, the fatigues could be postponed after the 5th round calibration so that it can improve the fitting precision effectively. The detailed statistics are shown in Table 2 or Fig. 8.

In addition, we also find that among all coefficient selection methods, **1I** is the poorest, **4IIOO** and **4IOLR** are the best. **1R** has the most anti-fatigue feature than the others except for our hybrid method. In other experiments we also find that modifying scheme is better than discarding one for 1-M calibration and discarding scheme is better than the modifying scheme for N-M calibration.

5 Conclusion

We proposed a novel algebraic B-spline curve fitting method which is composed of direct distance field fitting initialization and local topology-preserving calibrations. The method not only is free of additional branches as the direct distance field fitting [18] but also has high fitting precision as the Jüttler's method [16]. It is a non-iterative approach with a high fitting speed. Furthermore it does not need any manual tuning of fitting parameters.

In future, we will study the relation between calibration strategy and fitting performance further.

Acknowledgements This work is jointly supported by the NSF of China (60933007, 60736019), the 973 program of China (2009CB320801).

Appendix

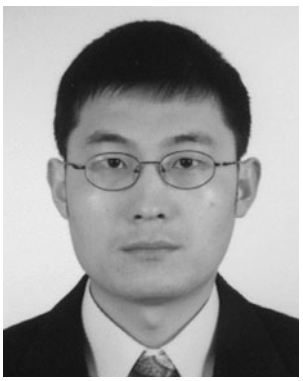
Table 3 Fitting precision comparison between AF and ABFF for 16 raw point sets: Item data are geometric errors (unit: 10^{-6}). The *last row GAG* is the average of geometric errors for all 16 raw point sets. The *shaded items* indicate the corresponding fitting results that contain additional branches

Raw point sets	Shape	$K=3, m=n=17$					$K=4, m=n=17$				
		AF $\omega_2=10^{-8}$	AF $\omega_2=10^{-6}$	AF $\omega_2=10^{-4}$	ABFF	DFF	AF $\omega_2=10^{-8}$	AF $\omega_2=10^{-6}$	AF $\omega_2=10^{-4}$	ABFF	DFF
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(9)	(9)	(10)
Bigcycle		0.0060	0.0363	0.3676	**0.0027**	2.0240	0.0028	0.0079	0.0559	**0.0017**	2.6453
Circle		0.0080	0.0170	0.2780	**0.0022**	1.5160	0.0026	0.0074	0.0670	**0.0007**	2.5804
Cncvl (300)		0.6159	0.3899	1.9007	**2.8129**	253.41	0.3989	0.3899	1.3411	**1.0627**	415.09
Cncv2		1.0829	1.1511	1.6717	**1.3163**	91.216	1.0484	1.0712	1.5556	**1.2218**	163.98
Concave		0.0852	0.1339	0.7525	**0.4577**	114.08	0.0881	0.0874	0.4811	**0.2929**	321.07
Crown 1		2.3480	3.3631	41.268	**1.3335**	1107.4	0.8684	2.6970	41.337	**1.4342**	2297.4
Crown2		1.6149	1.5684	10.522	**1.0767**	658.93	0.8209	0.8505	8.8124	**0.2222**	846.98
Fork		2.2422	2.2187	6.4363	**1.8434**	224.62	0.2475	0.8134	4.9885	**0.3452**	251.10
Leftcycle (54)		0.0020	0.0088	0.2718	**0.0004**	5.4689	0.0007	0.0021	0.1143	**0.0004**	6.2464
Littlecycle		0.0125	0.0445	0.5891	**0.0080**	3.3058	0.0024	0.0101	0.1097	**0.0006**	4.3326
Rectangle		1.2137	3.7518	18.920	**2.5754**	107.35	1.2919	2.6916	11.120	**3.0983**	266.24
Revs		0.6588	0.6258	5.9452	**0.3045**	222.32	0.0871	0.2442	4.6976	**0.3208**	132.23
Triangle		0.4868	2.0766	28.488	**4.8185**	768.68	0.3834	2.0004	19.523	**0.6875**	629.60
Vase		1.5745	1.8620	39.467	**7.2132**	2137.6	1.9811	4.4398	24.159	**13.2176**	689.33
Versl		0.7498	0.8799	5.3261	**1.1922**	200.0821	0.4801	0.5995	2.6494	**0.6147**	275.33
Xinzang		28.586	30.543	199.63	**102.60**	1058.1	16.759	21.3059	173.45	**4.2653**	2142.5
GAG		0.311	0.556	4.320	**0.392**	104.40	0.148	0.308	2.367	**0.164**	138.24

References

1. Stamati, V., Fudos, I.: A feature based approach to re-engineering objects of freeform design by exploiting point cloud morphology. In: Proceedings of the 2007 ACM symposium on Solid and Physical Modeling, pp. 347–353 (2007)
2. Hudson, J.: Processing Large Point Cloud Data in Computer Graphics. The Ohio State University, Columbus (2003)
3. Mahmoudi, M., Sapiro, G.: Three-dimensional point cloud recognition via distributions of geometric distances. Graph. Models **71**, 22–31 (2009)
4. Farin, G.: Curves and Surfaces for CAGD: A Practical Guide, Fifth Edition, pp. 119–146. Morgan Kaufmann, San Francisco (2002)
5. Ahn, S.: Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space, pp. 1–16. Springer, Berlin, Heidelberg (2004)
6. Pratt, V.: Direct least-squares fitting of algebraic surfaces. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 145–152 (1987)
7. 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**, 1115–1138 (1991)
8. Jüttler, B., Chalmovianský, P., Shalaby, M., Wurm, E.: Approximate algebraic methods for curves and surfaces and their applications. In: Proceedings of the 21st Spring Conference on Computer Graphics, pp. 13–18 (2005)
9. Ahn, S., Rauh, W., Warnecke, H.-J.: Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. Pattern Recogn. **34**, 2283–2303 (2001)
10. Redding, N.: Implicit polynomials, orthogonal distance regression, and the closest point on a curve. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 191–199 (2000)
11. Elber, G., Kim, M.-S.: Geometric constraint solver using multivariate rational spline functions. In: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, pp. 1–10 (2001)
12. Aigner, M., Jüttler, B.: Robust computation of foot points on implicitly defined curves. In: Duhlen, M., Schumaker, L. (eds.) Mathematical Methods for Curves and Surfaces: Robust Footpoint Computation. Nashboro Press (2005)
13. Krystek, M., Anton, M.: A weighted total least-squares algorithm for fitting a straight line. Meas. Sci. Technol. **18**, 3438–3442 (2007)
14. Wang, W., Pottmann, H., Liu, Y.: Fitting B-spline curves to point clouds by curvature-based squared distance minimization. ACM Trans. Graph. **25**, 214–238 (2006)
15. Sederberg, T., Zheng, J., Klimaszewski, K., Dokken, T.: Approximate implicitization using monoid curves and surfaces. Graph. Models Image Process. **61**, 177–198 (1999)
16. Jüttler, B., Felis, A.: Least-squares fitting of algebraic spline surfaces. Adv. Comput. Math. **17**, 135–152 (2002)
17. Yang, Z., Deng, J., Chen, F.: Fitting unorganized point clouds with active implicit B-spline curves. Vis. Comput. **21**, 831–839 (2005) (2005)
18. Li, Y., Feng, J., Jin, X.: Algebraic B-spline curve reconstruction based on signed distance field. J. Softw. **18**, 2306–2317 (2007)

Mingxiao Hu received the B.S. and M.S. degrees in computational mathematics from Peking University, People's Republic of China in 1986 and 1989, respectively. He is currently an assistant professor at Wenzhou University, P.R. China. In 2008, he was a visiting scholar at State Key Lab. of CAD & CG, Zhejiang University, P.R. China. His research interests include computer graphics and artificial intelligence.



Jieqing Feng is a professor in the State Key Lab of CAD&CG, Zhejiang University, People's Republic of China. He received his B.Sc. degree in applied mathematics from the National University of Defense Technology in 1992 and his Ph.D. degree in computer graphics from Zhejiang University in 1997. His research interests include geometric modeling, real-time rendering, 3D computer animation and 2D cartoon animation.

Jianmin Zheng received the B.S. and Ph.D. degrees from Zhejiang University, China. He is currently with School of Computer Engineering at Nanyang Technological University. Previously, he was a research faculty in Computer Science Department at Brigham Young University, USA and a faculty at Zhejiang University. He is a member of ACM SIGGRAPH. His research interest includes computer aided geometric design, CAD/CAM, computer graphics, animation, visualization and interactive digital media.