SCIENCE CHINA
Mathematics

• ARTICLES •

December 2014 Vol. 57 No. 12: 2621–2634
doi: 10.1007/s11425-014-4832-0

Parametric curve with an implicit domain

ZHANG WeiHong[1,2], DENG JianSong[1], YANG ZhouWang[1,*] & LIU LiGang[1]

[1]*School of Mathematical Sciences, University of Science and Technology of China, Hefei* 230026*, China;*
[2]*School of Mathematics and Statistics, Hefei Normal University, Hefei* 230601*, China*
Email: zwh@mail.ustc.edu.cn, dengjs@ustc.edu.cn, yangzw@ustc.edu.cn, lgliu@ustc.edu.cn

Received June 25, 2013; accepted January 3, 2014; published online June 26, 2014

Abstract In this paper we present a new representation of curve, named parametric curve with an implicit domain (PCID), which is a curve that exists in parametric form defined on an implicit domain. PCID provides a bridge between parametric curve and implicit curve because the conversion of parametric form or implicit form to PCID is very convenient and efficient. We propose a framework model for mapping given points to the implicit curve in a homeomorphic manner. The resulting map is continuous and does not overlap. This framework can be used for many applications such as compatible triangulation, image deformation and fisheye views. We also present some examples and experimental results to demonstrate the effectiveness of the framework of our proposed model.

Keywords parametric curve with an implicit domain, regularity, quadratic programming, image deformation

MSC(2010) 65D18, 65D17, 65D15

Citation: Zhang W H, Deng J S, Yang Z Y, et al. Parametric curve with an implicit domain. Sci China Math, 2014, 57: 2621–2634, doi: 10.1007/s11425-014-4832-0

1 Introduction

Curves and surfaces are the fundamental research objects in computer-aided geometric design (CAGD), and two methods are usually used to describe them: the parametric method and the implicit method. For the parametric method, early description of it is in [3]. Parametric curves and surfaces have become the main popular geometric representation in computer graphics (CG) and computer-aided design (CAD) to date. With the parametric method, each of the coordinates is expressed in terms of the geometric dimension of an object. For example, a one-dimensional curve is embedded in a 3D space and the coordinates are $x = x(t)$, $y = y(t)$, and $z = z(t)$; and, for a two-dimensional surface, $x = x(u, v)$, $y = y(u, v)$, and $z = z(u, v)$. These mappings relate a space of parameters to the curve or surface such that there is a correlation between points in these two spaces. Its advantage is easy to render and is helpful for some geometric operations, such as the computation of curvature or bounds and the control of position or tangency. There are many disadvantages and limitations to the parametric method. One problem is the difficulty in determining relationships between a fixed point and curves or surfaces. Another limitation is the difficulty in the representation of curves or surfaces with complex topology.

Due to the geometric and topological complexity of the curves and surfaces in industry, the modeling technology of implicit algebraic curves and surfaces is increasingly useful in more applications. Early implicit methods include scan-line rendering of algebraic surfaces [16]. More applications in computer graphics and geometric modeling have become available to use the implicit modeling techniques [1, 8, 14,

*Corresponding author

20]. Implicit equations, $f(x, y) = 0$ and $f(x, y, z) = 0$, are used to describe implicit curve and surface, which define the relation of points in an ambient space in relation to the curve and the surface. One of its advantages is its flexibility and ease of expression. Additionally, point location is done with ease, according to the sign of the implicit function. The implicit method also carries some disadvantages in that it is prone to have more branches. The implicit method is also not convenient for computer programming, not easy to control and adjust the object's shape.

Both methods have their inherent advantages, and they are complementary to some extent. In practical application, sometimes we need convert from one form to another in order to make use of their advantages. This motivates the researchers to explore means of converting from one method to another [9, 10, 15, 18]. The process of converting a parametric representation to an implicit representation is known as implicitization. The reverse of implicitization is parameterization, which is more difficult. It is well known that any parametric representation can be converted into an implicit representation. But, not every implicit representation admits a parameterization, even if it is algebraic. Thus, a combination of these two representations is needed.

In this paper, a new method for planar curve representation will be introduced, parametric curve with an implicit domain. This is defined on the implicit curve and the mapping form is the parametric format. It is helpful to express the curve in two aspects. Firstly, a more complex curve could be derived through the use of the new method because the domain of definition is an implicit curve. For example, the curve shown as the domain in Figure 1, is difficult to be represented in a parametric or an implicit form. However, it can be efficiently expressed in the following form:

$$\begin{aligned}
&\mathbf{r}(u, v) = (2vu - 3v, 4u + 2uv + v^2 + 0.6v^3u)\\
&\text{s.t.} \quad (u^2 + v^2 - 4)((u - 6)^2 + v^2 - 1)((u + 3)^2 + (v + 1)^2 - 1) = 0,
\end{aligned}$$

where the domain of definition is an implicit curve shown as the $[u, v]$ domain in Figure 1. Secondly, the mapping takes a parametric form, which provides sufficient flexibility and convenience for a few geometric operations: the computation of curvature, tangency, and the control of position.

Based on parametric curve with an implicit domain, the conversion between a parametric curve and an implicit curve can be carried out conveniently. In this process, the key step is the conversion from the implicit curve to the parametric curve with an implicit domain. In this regard, we propose a framework model for mapping points obtained from the implicit domain to the given implicit curve homeomorphically. In the model, the Sampson distance and Taylor expansion of an implicit curve are used in the goodness of fitting. Difference penalties on the control net are used for simplifying the problem. In the end, we formulate the issue as a standard quadratic programming problem subject to position constraints and regularity constraints. The resulting mapping generated by our algorithm is guaranteed to be smooth and does not overlap, and places the sample points to the implicit curve. We have applied our model on a variety of different problems, extended the model to general scenarios, and successfully verified it in image deformation, compatible triangulation and fisheye views.

Our contributions are summarized as follows.

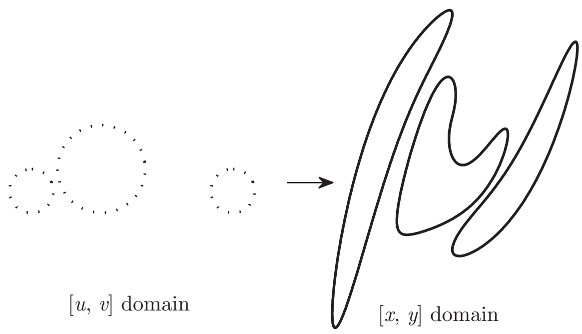

Figure 1 An example of parametric curve with an implicit domain

• We propose a new approach for representing curves, which combines the advantages of the parametric and implicit methods.

• Based on parametric curve with an implicit domain, an effective method to convert between parametric curve and implicit curve is now introduced.

• Verification of the method was achieved by utilizing various applications such as compatible triangulation, image deformation and the fisheye view, with results showing that the method was effective in all the aforementioned scenarios.

2 Definition

In this section, we give some basic notation on parametric curve with an implicit domain, and the mutual conversions between parametric curve with an implicit domain and other representations of a curve.

2.1 Notation

Definition 2.1. Let $\psi : D \longrightarrow \mathbb{R}^2$ be a continuous map, where $D \subset \mathbb{R}^2$. If

$$(u,v) \in V(g) = \{(u,v) \in D \mid g(u,v) = 0\},$$

the image of $\psi(u,v)$ is called a parametric curve with an implicit domain (PCID), and is denoted as $\psi|_{V(g)}$. More precisely, PCID $\psi|_{V(g)}$ is the restriction of ψ to $V(g)$ such that

$$\psi(u,v) = (x(u,v), y(u,v)) \quad \text{s.t.} \quad g(u,v) = 0.$$

Definition 2.2. A parametric curve with an implicit domain, $\psi|_{V(g)}$, is regular if it satisfies the following conditions:

• Both $x(u,v)$ and $y(u,v)$ are differentiable;

• $g(u,v) = 0$ is regular;

• $(|\frac{\partial(x,g)}{\partial(u,v)}|, |\frac{\partial(y,g)}{\partial(u,v)}|) \neq \mathbf{0}$ for all $(u,v) \in V(g)$.

Remark 2.3. Differential properties have been studied in [23].

As shown in Definition 2.1, curves can now be expressed in the parametric form, implicit form or PCID form generally. The following is an example of an ellipse.

Example 2.4. The three representations of an ellipse.

• Implicit:

$$e(x,y) = \frac{x^2}{9} + \frac{y^2}{4} - 1 = 0.$$

• Parametric:

$$\mathbf{r}(\theta) = (3\cos\theta, 2\sin\theta), \quad \theta \in [0, 2\pi).$$

• PCID:

$$\psi(u,v) = (3u, 2v) \quad \text{s.t.} \quad g(u,v) = u^2 + v^2 - 1 = 0.$$

Obviously, the implicit form is more compact while the parametric form is most convenient for generating points along a curve. The PCID $\psi|_{V(g)}$ maps points from $[u,v] \subset D$ domain to $[x,y] \subset \mathbb{R}^2$ domain. Usually the implicit domain $V(g)$ is set to be a curve which is as simple as possible and homeomorphic to the target shape. In this example, it is a unit circle. Also, the location of PCID is obtained by sweeping (u,v) through its implicit domain and substituting parameters (u,v) into the map $\psi(u,v)$. So it is beneficial to generate curve points for the purpose of display and also easy to determine whether a point lies on, inside or outside the PCID. Specially, if mapping $\psi(u,v)$ is represented with the tensor-product B-spline, for example as shown in Figure 2(a), the shape of the curve can easily be changed by moving the control points of B-spline as shown in Figure 2(b). The natural combination of the implicit form and the parametric form will be popular for some applications.

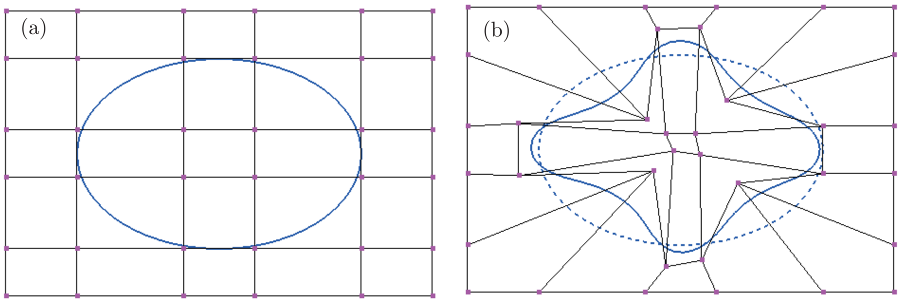

Figure 2 Changing the curve with the control points. (a) Original control net and curve; (b) Control net and the resultant curve

2.2 Conversion

With PCID, we now have three forms for a given curve: parametric, implicit, and PCID. We can obtain the conversion between PCID and any of the other two representations via precise or approximate calculations as shown in Figure 3.

I) PCID form to parametric form. For a given parametric curve with an implicit domain $\psi(u,v) = (x(u,v), y(u,v))$ s.t. $g(u,v) = 0$, the standard shape $g(u,v) = 0$ is generally a simple shape such as a circle, whose parametric form can be obtained easily. By substituting the parametric form of $g(u,v) = 0$ into $\psi(u,v)$, we can convert the PCID form to a parametric form. If the standard shape contains complicated topology, then we need to subdivide it into a union of simple shapes, each of which can be converted into its parametric form with the former approach.

II) PCID form to implicit form. For PCID $\psi(u,v) = (x(u,v), y(u,v))$ s.t. $g(u,v) = 0$, if $x(u,v)$ and $y(u,v)$ are rational functions, then the implicit form can be obtained by eliminating u and v from $\{x - x(u,v), y - y(u,v), g(u,v)\}$ via the Gröbner base method [4].

III) Parametric form to PCID form. For a parametric curve $x = x(t)$, $y = y(t)$ with $t \in [a,b]$, it can be considered as a special PCID by taking $\psi(u,v) = (x(u), y(u))$ and $g(u,v) = v$ with $(u,v) \in D = [a,b] \times [0,0]$.

IV) Implicit form to PCID form. An implicit curve $f(x,y) = 0$ can be considered as a PCID by taking $\psi(u,v) = (u,v)$ and $g(u,v) = f(u,v)$. If we want to obtain a normal standard shape, we can specify it according to the topology of $f(x,y) = 0$. Then we should construct a map from one region containing the standard shape to the other region containing the implicit curve. This map will then project points of the standard shape onto the implicit curve. This results in some good properties such as smoothness and low distortion. See Section 3 for more details.

As can be seen, an approximate process is needed from implicit representation to PCID, which can map given sample points from a standard shape to an implicit curve. As mentioned above, the mapping process should maintain good properties such as smoothness and low distortion to as high a degree as

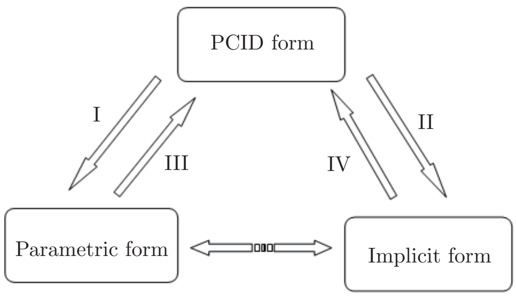

Figure 3 Conversions via different methods

possible. This process seems very simple and straightforward, but the implementation can be a bit tricky. In what follows, we focus on the conversion itself and will provide a model of it in Section 3.

3 Construction of mapping from points to an implicit curve

In this section, a model which to be used to construct a mapping from given sample points to an implicit curve, will be introduced.

Input: Given points $\{(x_k, y_k), k = 1, \ldots, N\}$ derived from implicit or parametric curves, and an implicit curve $f(u, v) = 0$.

Output: A continuous and bijective map.

In other words, a map $\Phi(x, y)$ from $[x, y] \in \mathbb{R}^2$ to $[u, v] \in \mathbb{R}^2$ as shown in Figure 4. Here, suppose the map has the form of a tensor-product B-spline. The formula should be expressed as

$$\Phi(x, y) = \sum_{i=1}^{n} \sum_{j=1}^{m} (P_{i,j}^1, P_{i,j}^2) N_i(x) N_j(y),$$

where $N_i(x)$ and $N_j(y)$ are B-spline basis functions,

$$(P_{i,j}^1, P_{i,j}^2), \quad i = 1, \ldots, n, \quad j = 1, \ldots, m,$$

is the control net. Note that the number and positions of control points, and therefore the number of knots of the spline curve, are changed during the process.

3.1 Objective function

The objective function consists of two terms. Its general form is

$$E = E_f + E_p,$$

where E_f is the goodness of fitting and E_p is a penalty term to obtain a fair mapping.

3.1.1 *Goodness of fitting*

The specification of constraints is an interesting part of the model. These constraints ensure that every given point (x_k, y_k) on the parametric or implicit curves can be mapped onto the implicit curve $f(u, v) = 0$. It means that points

$$(u_k, v_k) = \sum_{i=1}^{n} \sum_{j=1}^{m} (P_{i,j}^1, P_{i,j}^2) N_i(x_k) N_j(y_k)$$

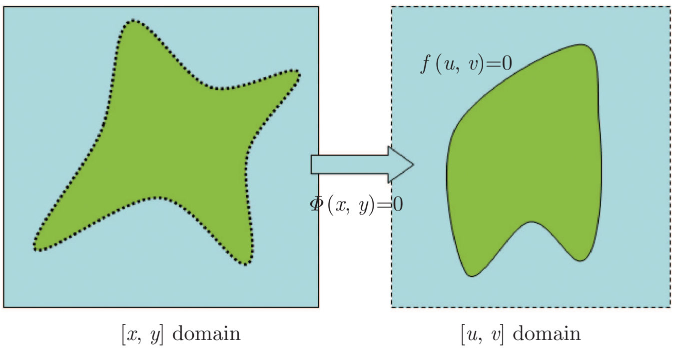

Figure 4 Mapping points to an implicit domain

should satisfy the equation $f(u_k, v_k) = 0$. Supposing the current value of control net is

$$\{(\tilde{P}^1_{i,j}, \tilde{P}^2_{i,j}), i = 1, \dots, n, j = 1, \dots, m\},$$

the current position in the $[u, v]$ domain is

$$(\tilde{u}_k, \tilde{v}_k) = \sum_{i=1}^{n} \sum_{j=1}^{m} (\tilde{P}^1_{i,j}, \tilde{P}^2_{i,j}) N_i(x_k) N_j(y_k)$$

for point (x_k, y_k).

In the algorithm, we expect the iteration of every point along the gradient direction of the implicit curve $\nabla f(u, v) = (f_u, f_v)$ as shown in Figure 5, where the algebraic distance of point $(\tilde{u}_k, \tilde{v}_k)$ is $f(\tilde{u}_k, \tilde{v}_k)$, and $\frac{f(\tilde{u}_k, \tilde{v}_k)}{\|\nabla f(\tilde{u}_k, \tilde{v}_k)\|}$ is the Sampson distance [13] which is an approximation of geometric distance. Then we should minimize the following equation

$$\sum_{k=1}^{N} \left(\frac{f(u_k, v_k)}{\|\nabla f(u_k, v_k)\|} \right)^2.$$

With the Taylor expansion of $f(u, v)$ at point$(\tilde{u}_k, \tilde{v}_k)$, here $f(u_k, v_k)$ can be replaced by

$$f(\tilde{u}_k, \tilde{v}_k) + \nabla f(\tilde{u}_k, \tilde{v}_k)((u_k, v_k) - (\tilde{u}_k, \tilde{v}_k))^{\mathrm{T}}$$

for the simplification of the calculation.

Hence, we will take the following constraint for the goodness of fitting in the objective function,

$$E_f = \sum_{k=1}^{N} \left(\frac{\nabla f(\tilde{u}_k, \tilde{v}_k)((u_k, v_k) - (\tilde{u}_k, \tilde{v}_k))^{\mathrm{T}}}{\|\nabla f(\tilde{u}_k, \tilde{v}_k)\|} + \frac{f(\tilde{u}_k, \tilde{v}_k)}{\|\nabla f(\tilde{u}_k, \tilde{v}_k)\|} \right)^2. \tag{3.1}$$

3.1.2 *Penalty term*

The fairness of tensor-product B-spline functions should be taken into consideration. So one penalty function reflecting the hardness and the smoothness to a certain extent should be provided in the model. A common smoothness penalty [12] is the integral of the square of the second derivative of B-spline function. But it is complex to compute and not easy to be used in a higher dimensional space for tensor-product B-spline functions. Here, appropriate difference penalties on the rows and columns of the control net are used. To shorten the scope of the penalties we assume the following:

$$E^1_{\cdot,j} = \sum_{i=2}^{n-1} (P^1_{i-1,j} - 2P^1_{i,j} + P^1_{i+1,j})^2, \quad E^1_{i,\cdot} = \sum_{j=2}^{m-1} (P^1_{i,j-1} - 2P^1_{i,j} + P^1_{i,j+1})^2,$$

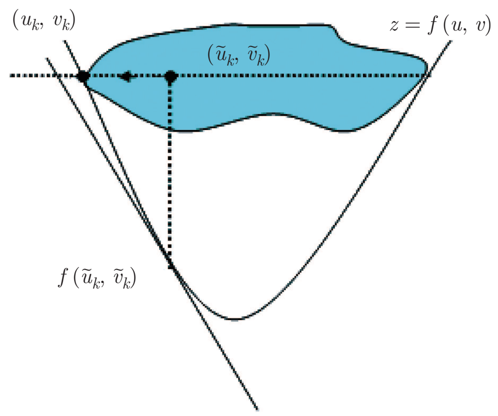

Figure 5 Sampson distance and iteration procedure

$$E_{\cdot,j}^2 = \sum_{i=2}^{n-1} (P_{i-1,j}^2 - 2P_{i,j}^2 + P_{i+1,j}^2)^2, \quad E_{i,\cdot}^2 = \sum_{j=2}^{m-1} (P_{i,j-1}^2 - 2P_{i,j}^2 + P_{i,j+1}^2)^2,$$

where $E_{i,\cdot}^t$ and $E_{\cdot,j}^t$ $(t = 1, 2)$ depend on the row and column of the control net respectively. The penalty term is then the sum of all the sets $\{E_{i,\cdot}^1, E_{i,\cdot}^2, i = 1, \dots, n\}$ and $\{E_{\cdot,j}^1, E_{\cdot,j}^2, j = 1, \dots, m\}$:

$$E_p = \sum_{i=1}^{n} (E_{i,\cdot}^1 + E_{i,\cdot}^2) + \sum_{j=1}^{m} (E_{\cdot,j}^1 + E_{\cdot,j}^2). \tag{3.2}$$

The difference penalty on the coefficients is a good discrete approximation to the integrated square of the 2nd-order derivatives of the tensor-product B-splines. It can be implemented easily, and can make the systems of equations have a lower bandwidth. Also it is easy to construct an automatic procedure for incorporating the penalty in the whole model.

3.2 Constraints

Two types of constraints will be introduced: position constraints which are used when the mapping result for given points is needed to pass fixed positions, and regularity constraints to ensure that the mapping does not overlap.

3.2.1 *Position constraint*

These constraints ensure that some given points $\{(x_i, y_i), i = 1, \dots, I\}$ on the $[x, y]$-domain are mapped to the corresponding given points $\{(u_i, v_i), i = 1, \dots, I\}$ of the $[u, v]$-domain. It means that the coordinates of points should be satisfied with the following constraints:

$$u_k = \sum_{i=1}^{n} \sum_{j=1}^{m} P_{i,j}^1 N_i(x_k) N_j(y_k), \tag{3.3}$$

$$v_k = \sum_{i=1}^{n} \sum_{j=1}^{m} P_{i,j}^2 N_i(x_k) N_j(y_k), \tag{3.4}$$

where $k = 1, \dots, I$.

3.2.2 *Regularity constraint*

For continuous mapping $\Phi(x, y)$, when the Jacobian $J_\Phi > 0$ [2, 6, 11], it is injective. Since the condition is nonlinear for control points, it is not convenient to use. In [21], a simple linear condition is provided based on boundaries of the convex cones. If two different linear equations l_1 and l_2 are given which can be considered as the boundaries of two special cones $\tau_1(C)$ and $\tau_2(C)$, where the cones $\tau_1(C)$ and $\tau_2(C)$ are defined by $l_1(\tau_1(C)) \geqslant 0$, $l_2(\tau_1(C)) \geqslant 0$, and $l_1(\tau_2(C)) > 0$, $l_2(\tau_2(C)) < 0$, respectively. Based on the two special transverse cones $\tau_1(C)$ and $\tau_2(C)$, more constraints for control net can be used to prevent overlap for our framework using the following forms:

$$\begin{aligned} &l_1(P_{i,j+1} - P_{i,j}) \geqslant 0, \quad l_2(P_{i,j+1} - P_{i,j}) \geqslant 0, \quad i = 1, \dots, n, \quad j = 1, \dots, m-1, \\ &l_1(P_{i+1,j} - P_{i,j}) \geqslant 0, \quad l_2(P_{i+1,j} - P_{i,j}) \leqslant 0, \quad i = 1, \dots, n-1, \quad j = 1, \dots, m. \end{aligned} \tag{3.5}$$

3.3 The framework model for PCID

First, we rewrite the term of the goodness of fitting in Subsection 3.1. For expedience, we stack the variables $P_{i,j}^k$ into a vector

$$\boldsymbol{X} = (\dots, P_{i,j}^1, \dots, P_{i,j}^2, \dots)^{\mathrm{T}}.$$

Then the goodness of fitting E_f can be rewritten as

$$E_f = \sum_{k=1}^{N} \left(\frac{\bigtriangledown f(\tilde{u}_k, \tilde{v}_k)(u_k, v_k)^{\mathrm{T}}}{\|\nabla f(\tilde{u}_k, \tilde{v}_k)\|} + \frac{f(\tilde{u}_k, \tilde{v}_k) - \bigtriangledown f(\tilde{u}_k, \tilde{v}_k)(\tilde{u}_k, \tilde{v}_k)^{\mathrm{T}}}{\|\nabla f(\tilde{u}_k, \tilde{v}_k)\|} \right)^2,$$

where

$$\frac{\bigtriangledown f(\tilde{u}_k, \tilde{v}_k)(u_k, v_k)^{\mathrm{T}}}{\|\nabla f(\tilde{u}_k, \tilde{v}_k)\|} = \boldsymbol{G}_k \boldsymbol{X},$$

$\boldsymbol{G}_k$ is a sparse row vector with $2nm$ elements, and

$$\frac{f(\tilde{u}_k, \tilde{v}_k) - \bigtriangledown f(\tilde{u}_k, \tilde{v}_k)(\tilde{u}_k, \tilde{v}_k)^{\mathrm{T}}}{\|\nabla f(\tilde{u}_k, \tilde{v}_k)\|} = q_k,$$

$q_k \in \mathbb{R}$. Then, putting all the row vectors $\boldsymbol{G}_k, k = 1, \ldots, N$ and $q_k, k = 1, \ldots, N$ on top of each other respectively, a matrix $\boldsymbol{G}$ and a vector $\boldsymbol{q}$ are obtained. The goodness of fitting can be rewritten as $\|\boldsymbol{G}\boldsymbol{X} - \boldsymbol{q}\|_2^2$, where $\boldsymbol{G}$ is a matrix of dimensions $N \times 2mn$. In the penalty term, each term $E_{i,j}^k$ can be written as

$$E_{i,j}^k = (\boldsymbol{b}_{i,j}^1 \boldsymbol{X})^2 + (\boldsymbol{b}_{i,j}^2 \boldsymbol{X})^2,$$

where $\boldsymbol{b}_{i,j}^s, s = 1, 2$ are sparse (only three non-zero entries) row vectors having the same length as the column vector $\boldsymbol{X}$. Let us then stack all of the row vectors $\{\boldsymbol{b}_{i,j}^s\}$ on top of each other, to get a matrix $\boldsymbol{L}$. Then the penalty can be simply written as $E_p = \|\boldsymbol{L}\boldsymbol{X}\|_2^2$. Note that $\boldsymbol{L}$ is a matrix of dimensions $(2m(n-1) + (m-1)n) \times 2mn$.

Now, turn to the constraints. For any point each coordinate is expressed via a linear equation as

$$u_k = \sum_{i=1}^{n} \sum_{j=1}^{m} P_{i,j}^1 N_i(x_k) N_j(y_k),$$

where k is $1, \ldots, I$. These then compose the linear equality constraints $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{b}$. The constraints to prevent overlap are of the form $l_1(P_{i,j+1} - P_{i,j}) \leqslant 0$ for all i and j, similar to other constraints, where l_1 and l_2 are all line equations. Thus, the constraints are all linear inequality constraints $\boldsymbol{C}\boldsymbol{X} \leqslant \boldsymbol{0}$. Based on the above analysis, the model for creating mapping from given sample points to an implicit domain can be described as follows,

$$\begin{aligned} \min \quad & (\|\boldsymbol{G}\boldsymbol{X} - \boldsymbol{q}\|_2^2 + \lambda \|\boldsymbol{L}\boldsymbol{X}\|_2^2) \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{X} = \boldsymbol{b}, \quad \boldsymbol{C}\boldsymbol{X} \leqslant \boldsymbol{0}, \end{aligned} \tag{3.6}$$

where $\boldsymbol{A}$ is matrix of dimensions $2I \times 2nm$, $\boldsymbol{C}$ is matrix of dimensions $2(n(m-1) + m(n-1)) \times 2mn$, λ is a parameter for continuous control over smoothness of the result and is determined by the generalized cross-validation (GCV) method [19] in our model. The constraint optimization problem is a standard quadratic programming and can be solved by the Matlab CVX [7]. And the initial control points are provided with the equivalent map and knots-inserting algorithm.

3.4 Algorithm

Step 1. Given the initial value $\tilde{X}$, the tolerance errors $\varepsilon_0 \geqslant 0$ and $\varepsilon_1 \geqslant 0$.

Step 2. Compute matrices G, A, C and vectors q, b in (3.6) with the given sample points $\{(x_k, y_k), k = 1, \ldots, N\}$, implicit function $f(u, v) = 0$ and initial value $\tilde{X}$.

Step 3. Solve the quadratic programming with the Matlab CVX, where λ is determined by GCV method. Calculate errors $e_1 = \|X - \tilde{X}\|$ and $e_0 = E_f$ with the result X.

Step 4. If $e_0 \leqslant \varepsilon_0$, stop the computation. Otherwise, if $e_1 \leqslant \varepsilon_1$, insert new knots and get a new vector X with knot insertion algorithm. Update $\tilde{X} = X$, and return to Step 2.

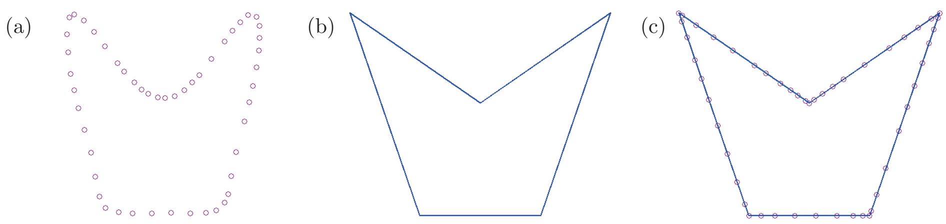

Figure 6 Sample points mapped to linear borders. (a) Sample points; (b) Linear borders; (c) Sample points mapped to the borders

3.5 General cases

For the above model, the input data includes given points $\{(x_k, y_k), k = 1, \ldots, N\}$ and an implicit curve $f(u, v) = 0$, while the implicit curve is only used to provide the gradient direction $\nabla f(u, v)$ and approximate the geometric distance $f(u, v)/\|\nabla f(u, v)\|$ in the algorithm. Hence the model could be extended to more general cases. It not only can be used to construct the mapping from the given sample points to an implicit curve, but also can provide mapping between two groups of given sample points $\{(x_k, y_k), k = 1, \ldots, N\}$ and $\{(U_r, V_r), r = 1, \ldots, R\}$. The signed distance for given points $\{(U_r, V_r), r = 1, \ldots, R\}$ is easily calculated and a direction can be given based on the signed distance, which can be used to replace the geometric distance and gradient direction in the above model. As an example the process which maps the given sample points to border lines is provided in Figure 6. This shows the scope of our model.

4 Applications and results

In this section, we illustrate our results by various examples and show comparison results between original, intermediate deformed and optimized results in the following figures. And we also successfully apply the proposed model to several different fields of digital image processing.

4.1 Results

For the given model, it is critical to construct a mapping from given sample points to standard shape or an implicit curve. Therefore, a few experiments are conducted over various sample points from implicit or parametric curves to show its effectiveness. And some of these results are illustrated in Figures 7–10, where the curve with blue points indicates the standard shape, and the black points are sample points in the experiment.

Example 4.1. Convert the implicit curve

$$x^2 + y^2 + yx - \sqrt[2]{x^2/6 + y^2} = 0$$

to PCID using the model of Section 2.

It is a closed curve in the domain $[-4, 4] \times [-4, 4]$. We select the unit circle as the standard shape and sample points are obtained from it in Figure 7(a). With the model, a mapping from the closed curve to the unit circle is constructed as shown in Figure 7(c), where the origin curve and target curve are unit circle and the given implicit curve respectively. This is the approximate PCID of the implicit curve. Figure 7(b) shows the result of the first step iteration in our algorithm, which has a big error. After several iterations, a high-quality result is derived in Figure 7(c). And the mapping result of isolines in domain $[-4, 4] \times [-4, 4]$ is shown in Figure 7(d).

Example 4.2. Construct the mapping from given sample points to the implicit curve, where sample points are obtained from four different polygons as shown in Figure 8(a), and the implicit curve is defined

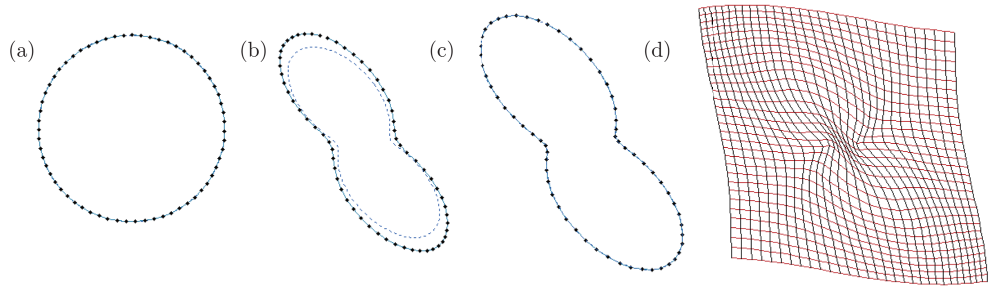

Figure 7 Mapping from standard shape to implicit curve for Example 4.1. (a) The sample points from standard; (b) Intermediate result; (c) Improved result; (d) The mapping results of isolines

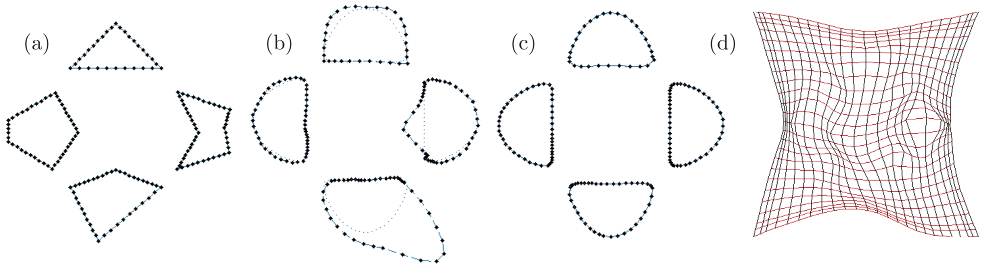

Figure 8 Mapping result for Example 4.2. (a) The sample points of several polygons; (b) Intermediate result; (c) Improved result; (d) The mapping results of isolines

by the zero set of

$$f = 4y^4 + 17x^2y^2 - 20y^2 + 4x^4 - 20x^2 + 17.$$

For this example, the origin curve are four polygons, and the target curve is the implicit curve $f = 0$. The implicit curve is more complicated as shown in Figure 8, which has four components in the bounded domain of $[-3, 3] \times [-3, 3]$. The mapping result is given in Figure 8(c) with our method, where the sample points for every polygon can be properly mapped to the corresponding branch of the implicit curve.

Example 4.3. Sample points $\{(x_i, y_i), i = 1, \ldots, k\}$ are taken from two curves shown in Figure 9(a), which are nested. And the implicit curve is $(u^2 + v^2 - 1)(u^2 + v^2 - 4) = 0$. Give the mapping result from the sample points to implicit curve.

Sample points are nested, so two nested circles $g(u, v) = (u^2 + v^2 - 1)(u^2 + v^2 - 4) = 0$ are selected as the standard shape, which is the target curve in this example. With our method, the good mapping result is obtained. And we show one intermediate result in Figure 9(b) and the improved result in Figure 9(c). The mapping result of isolines in domain $[-6, 6] \times [-6, 6]$ is shown in Figure 9(d). In fact, this process provides an approximate implicit representation of the sample points. Assuming the mapping result is $u = u(x, y), v = v(x, y)$, and we take it into $g(u, v) = 0$, thus implicit representation $F(x, y) = 0$

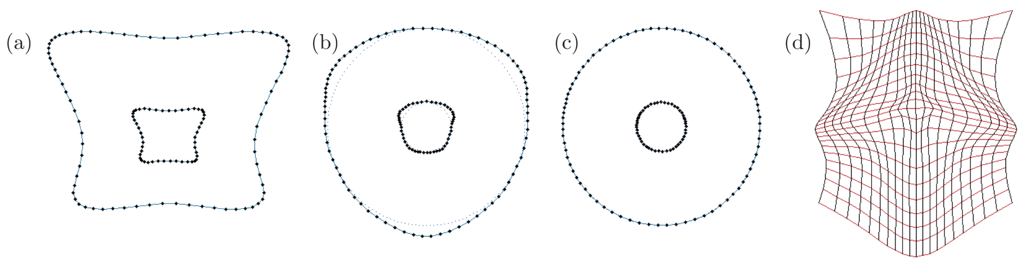

Figure 9 Mapping result for Example 4.3. (a) The sample points of given curves; (b) Intermediate result; (c) Improved result; (d) The mapping results of isolines

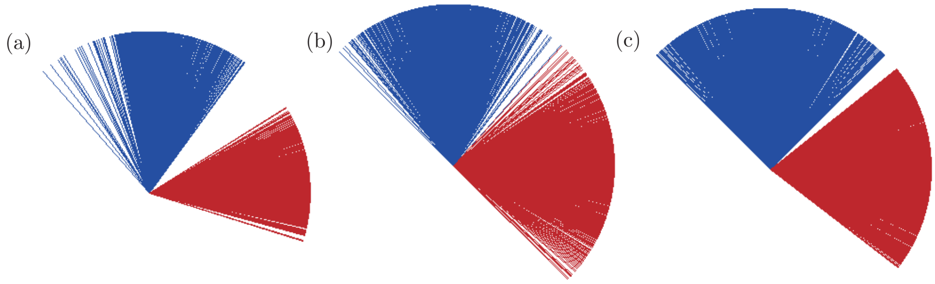

Figure 10 Transverse cones. (a) Transverse cone for Example 4.1; (b) Transverse cone for Example 4.2; (c) Transverse cone for Example 4.3

is obtained.

For these examples, cones generated by

$$\{P_{i+1,j} - P_{i,j}, i = 1, \ldots, n-1, j = 1, \ldots, m\} \quad \text{and} \quad \{P_{i,j+1} - P_{i,j}, i = 1, \ldots, n, j = 1, \ldots, m-1\}$$

are transverse, which are shown in Figure 10.

4.2 Applications

In fact, the mapping from one region to another, which maps given points to curves with some good properties, can be constructed with the model depicted in Section 3. According to this feature, it could be applied in some fields such as compatible triangulations, image deformation and interactive fisheye views.

4.2.1 *Compatible triangulations*

In computational geometry, so-called compatible triangulations [17] of two planar regions bounded by different polygons is an important problem. Let P and Q be simple polygons with vertex sets $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$. Firstly, we should construct an approximate implicit representation of one polygon such as polygon Q. And it is not difficult to obtain with our model since it maps the vertexes of polygon Q to a standard shape which has an implicit representation. Based on the approximate implicit representation of Q, a continuous one-to-one map ψ which maps each vertex $p_i \in P$ to approximate implicit curve, can be provided with the given model. In the process, polygon P and approximate implicit curve are the origin curve and target curve respectively. With the triangulations of polygon P, the compatible triangulation of Q can be carried out just by computing the corresponding triangulations using the mapping ψ. In Figure 11, the compatible triangulation results of two polygons are shown.

4.2.2 *Image deformation*

Image deformation [5] permits users to simply draw the outline Q of the source image region and sketch a new boundary shape S onto the location where this region is to be pasted. Since our method can be extended to construct mapping from sample points to polygon as shown in Subsection 3.4, a continuous map from the target image region to the source image region which maps each vertex $s_i \in S$ to the corresponding vertex $q_i \in Q$ can be provided easily. Based on the resultant mapping, the deformed image can be quickly computed using texture mapping. For an example, the source image and outline are shown in Figure 12(a). After the outline is sketched to the new place as shown in Figure 12(b), the image deformation result using our method is provided in Figure 12(b), where the visual distortions in the deformed image are very small by our approach. And we also provide a local image deformation result as shown in Figure 13, which can be done by fixing several inner points to given positions in this process. The origin fixing inner points and their target positions are shown in Figures 13(a) and 13(b) respectively, which are highlighted in green.

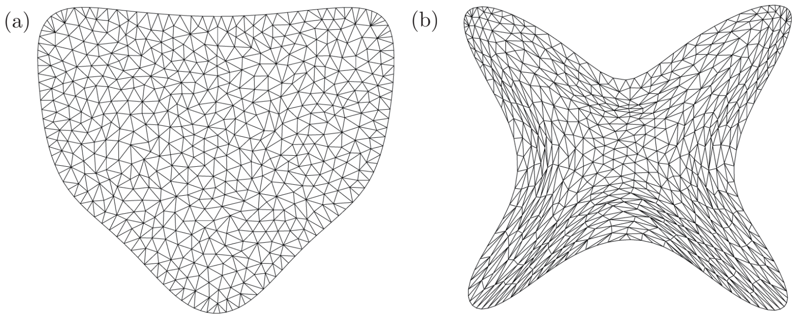

Figure 11 Compatible triangulation. (a) Source picture; (b) Target picture

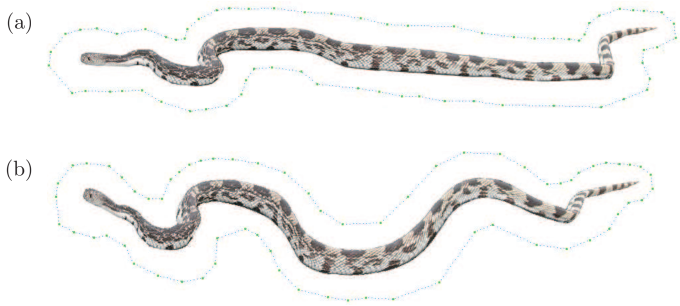

Figure 12 Snake image deformation result. (a) Source image; (b) Deformed image

4.2.3 *Interactive fisheye views*

Interactive fisheye views [22] are focus plus context techniques, which are characterized by the in-place magnification of the focus area and the continuous transition to the demagnification of the surrounding context. The result supports detailed inspection tasks and also maintains a sense of the entire data set. With our method, the sample points can be selected from a small circle for local magnification, and a large circle is used as the standard shape. It means that the small circle is selected to be the origin curve and the target curve is a large circle. Then, a mapping from the sample points to the large circle is given. The result is shown in Figure 14(b), where local details of the focus part becomes larger and the context area has some distortion. We can further learn from the techniques of focus plus glue plus context; a larger circle should be selected as the glue area at the same time. Then, more position constraints need be added in our model, which is used to ensure that there are fewer distortions after mapping in the context area. The mapping result is in Figure 14(c), which restricts the distortion to the glue area around the focus. Such results are valuable for helping users maintain a mental map of workspace.

As can be seen in the above examples, the results through our approach are excellent. Our method

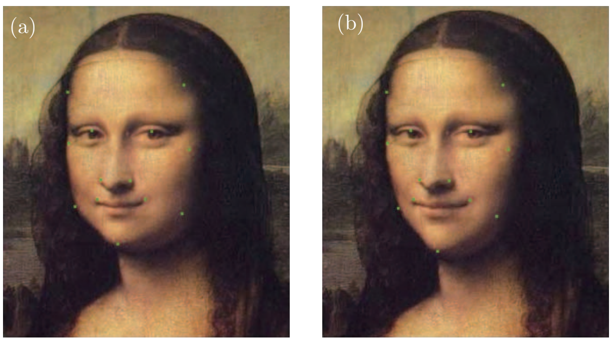

Figure 13 Mona image deformation result. (a) Source image; (b) Deformation image

Figure 14 Fisheye results. (a) Original picture; (b) Global context distortion; (c) Distortion area around the focus

has several advantages: (1) It is fast and robust since the proposed framework is quadratic programming; (2) it is a continuous method that guarantees the processing result without any folding over on itself; (3) the adaptive subdivision principle based on the B-spline can also be used during the processing.

5 Conclusion

We have discussed a new representation for a plane curve, which combines the advantages of parametric and implicit representations. Additionally, the model is provided to get the mapping from given sample points to the implicit domain. Based on this model the conversions among them will become easy. Several examples and applications are given to show its effectiveness. The main disadvantages of the method are that it can not be applied to some special cases (for example, a curve with singularities), and sometimes using the GCV computer the parameter consumes more time if there is a large matrix. Our future research work is to improve on our method, extend it to three-dimensional spaces and use it in volume data parametrization.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 11031007, 11171322, 61222206 and 11371341), One Hundred Talent Project of the Chinese Academy of Sciences, and the Program for New Century Excellent Talents in University (Grant No. NCET-11-0881). The authors would like to thank the anonymous reviewers for their constructive comments. The authors appreciate Jin Zhou, for his help on making some application figures for in this paper.

References

1 Bloomenthal J, Chandrajit B, Jim B, et al. Introduction to Implicit Surfaces. San Francisco: Morgan Kaufmann Publishers, 1997
2 Chen H H. The Schwarz-Pick lemma and Julia lemma for real planar harmonic mappings. Sci China Math, 2013, 56: 2327–2334
3 Coons S A. Surfaces for Computer-Aided Design of Space Forms. Cambridge: Massachusetts Institute of Technology Press, 1967
4 Cox D, Little J, O'Shea D. Ideals, Varieties, and Algorithms. New York: Springer-Verlag, 1997
5 Eitz M, Sorkine O, Alexa M. Sketch based image deformation. In: Proceeding of Vision, Modeling and Visualization. Saarbrücken: MPI and AKA GmbH, 2007
6 Gain J E, Dougson N A. Preventing self-intersection under free-form deformation. IEEE Trans Visual Comput Graphics, 2001, 7: 289–298
7 Grant M, Boyd S. Cvx: Matlab software for disciplined convex programming. Version 2.0 beta, 2013
8 Pasko A, Adzhiev V, Sourin A, et al. Function representation in geometric modeling: Concepts, implementation and applications. Visual Comput, 1995, 11: 429–446
9 Praun E, Hoppe H. Spherical parameterization and remeshing. ACM Trans Graphics, 2003, 22: 340–349
10 Qu Y M, Su J Z, Chen F L. Approximate parametric representation of algebraic curves (in Chinese). J Univ Sci Tech China, 1997, 12: 382–388
11 Reider I. Nonabelian Jacobian of smooth projective surfaces – a survey. Sci China Math, 2013, 56: 1–42

12 Reinsch C H. Smoothing by spline functions. Numer Math, 1967, 10: 177–183
13 Sampson P D. Fitting conic sections to very scattered data: An iterative refinement of the bookstein algorithm. Comput Graphics Image Process, 1982, 18: 97–108
14 Sclaroff S, Pentland A. Generalized implicit functions for computer graphics. Comput Graphics, 1991, 25: 247–250
15 Sederberg T W, Zhao J W, Zundel A K. Approximate Parameterization of Algebraic Curves. In: Theory and Practice of Geometric Modeling. Berlin-Heidelberg: Springer-Verlag, 1989
16 Sederberg T W, Zundel A K. Scan line display of algebraic surfaces. Comput Graphics, 1989, 23: 147–156
17 Surazhsky V, Gotsman C. High quality compatible triangulations. Engingeering Comput, 2004, 20: 147–156
18 Waggenspack W N, Anderson D C. Piecewise parametric approximation for algebraic curves. Comput Aided Geom Design, 1989, 6: 33–53
19 Wahba G. A comparison of gcv and gml for choosing the smoothing parameter in the generalized spline smoothing problem. Ann Stat, 1985, 13: 378–402
20 Wyvill B, McPheeters C, Wyvill G. Animating soft objects. Visual Comput, 1986, 2: 235–242
21 Xu G, Mourrain B, Duvigneau R, et al. Parameterization of computational domain in isogeomatric analysis: Methods and comparision. Comput Methods Appl Mech Engrg, 2011, 200: 2021–2031
22 Yamamoto D, Ozeki S, Takahashi N. Focus+glue+context: An improved fisheye approach for web map services. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: ACM, 2009, 101–110
23 Zhang W H, Li Y, Deng J S. The differential geometry of implicitly parametric curve and surface (in Chinese). J Univ Sci Tech China, 2012, 42: 482–487