J Math Imaging Vis (2016) 55:89–104
DOI 10.1007/s10851-015-0615-7

A Novel Fractional Implicit Polynomial Approach for Stable Representation of Complex Shapes

Gang Wu[1] · **Yanchun Zhang**[2]

Received: 17 April 2015 / Accepted: 26 October 2015 / Published online: 2 November 2015

Abstract Implicit polynomials (IPs) are applied to represent 2D object shapes in image processing and computer vision. However, it is difficult for IPs to represent complex object shapes due to high computational cost and high instability. In this work, we present a new representation model based on IPs, which is called fractional implicit polynomial (FIP). Firstly, the general formula for FIP and a definition of base are given; secondly, we investigate the properties of FIPs and conclude that the FIP representation exhibits higher stability and higher power than IP representation due to the presence of the base. Thirdly, we develop an algorithm for determination of a moderate degree for an FIP to represent a given shape, which can be obtained by only computing the number of stationary points on the shapes. We compare FIPs with IPs by test on various object shapes and the results show that the FIP is indeed sufficiently powerful to represent the complex object shapes.

Keywords Algebraic curve · Curve and surface fitting · Implicit polynomial · Shape representation

1 Introduction

Implicit polynomials (IPs) have long been applied to represent 2D curves and 3D surfaces specified by discrete data in various fields, such as object recognition [3,9,16,20,21,24], pose estimation [27,28], coding [8], boundary estimation from intensity or color images [22], symmetry detection [14,15], image database indexing [11], representation of time series [26], etc. In contrast to other representations such as Implicit B-Splines (IBSs) [18], Fourier descriptors, radial basis functions [6], and Poisson equations [10], IPs have the main advantages attracting application in various fields, including fast fitting, few parameters, algebraic/geometric invariants, robustness against noise and occlusion, the ease of containment computations (by computing the sign of the polynomial), etc. [12,29]. Specifically, IP fitting to noisy data has low computational cost and the coefficients of IP appear to be relatively insensitive to noise or modest changes in the subset of boundary used, which can help in recognizing objects rapidly.

Despite the great advantages mentioned above, IP representation mainly suffers from three major issues as follows:

1. Difficult to accurately describe complex objects. Objects obtained by vision modalities are often complex but IP can only coarsely represent their global features [29]. In order to capture the local features, we must choose a higher degree IP to represent them, but this would reduce the feasibility and efficiency of this representation due to its high computational cost and numerical instability. As an example, Fig. 1a shows original dataset of the bear, and Fig. 1b shows the corresponding result of fitting 12th degree IP curve to this bear by using the state of the art fitting algorithms in [29]. Obviously, although the 12th degree IP is chosen to represent the shape of the bear, the fitted IP curve cannot capture the enough local information yet, i.e., the head and the foot. Therefore, further improving the fitting result needs IP of degree greater than 12. That is, more than 91 coefficients of IP should

✉ Gang Wu
wugang69@gmail.com

Yanchun Zhang
yanchun.zhang@vu.edu.au

[1] Institute of Multimedia Computing Technology, Nanjing University of Finance and Economics, Nanjing 210003, China

[2] Centre for Applied Informatics, Victoria University, Melbourne, VIC 8001, Australia

be used to represent the bear shape, which implies that the IP has poor representations for complex objects.

2. Numerical instability and high computational cost, especially for high-degree IP representations. Numerical instability means that even tiny errors in the coefficients values of IPs may result in large fitting errors. This problem is due to extremely high sensitivity of high-degree IP to small changes of dataset points [2,13]. On the other hand, high-degree IP fitting dataset points leads to high computational complexity because the computational complexity of the IP fitting algorithm grows exponentially. As shown in Fig. 1b, the silhouette of bear is represented by 12th degree IP curve. Then, for the leading monomials of this IP, the fitting processes need computation of the coordinate of every point in the dataset to the power of 12, which results in high computational cost and high computational error. For the above reason, generally, only low degree IP is used to represent objects, which greatly limits the application of IP representations.
3. Great sensitivity to preprocessing methods. The stability of IP representations partially depends on which preprocessing methods are chosen to scale the dataset points [2,13]. If the coordinates of the dataset points are much smaller than 1, then higher degree monomials of IP are very sensitive to small changes of the dataset. On the other hand, if these coordinates are much larger than 1, then lower degree monomials of IP are very sensitive to small changes of the dataset. Consequently, the aim of preprocessing dataset points is to scale all of dataset points so that their coordinates will be close to value 1 or -1 [13]. That is, dataset is first translated to have the center of the mass at origin and then data points are scaled to be as close as possible to the unit square. However, none of any scaling methods can achieve this aim. The reason is that the shapes of objects are usually complex. As shown in Fig. 1a, whatever preprocessing methods are used, some points (i.e., those in the head, foot, neck, and chest) are always far away from value 1 or -1, but the others are close to value 1 or -1, and vice versa.

Fractional polynomials were introduced and applied to regression analysis, exhibiting good properties especially in representing multivariable regression models [4,19]. Motivated by fractional polynomials, in this work, we propose a novel fractional implicit polynomial (FIP) as a representation method for overcoming the above three weaknesses. We first present the concept of FIP and formulate it, and then, explore the properties of the FIP, including its higher stability and lower computational cost. In addition, we develop an algorithm for the determination of a moderate degree for an FIP before fitting a given object. Simulation results show the FIP is more suitable for representing complex objects than the IP.

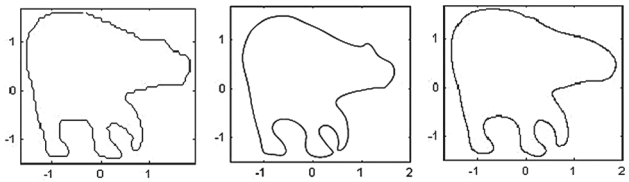

Fig. 1 IP representation for complex objects: **a** original object, **b** 12th degree IP fitting result, **c** FIP degree 5 and base 3 fitting result

Table 1 The number of coefficients

Base (right) Degree (down)	3	5	7	9	11
2	10	14	18	22	26
3	25	44	67	94	125
4	49	99	165	247	345
5	82	179	312	481	686
6	124	284	508	796	1148
7	175	414	753	1192	1731
8	235	569	1047	1669	2435
9	304	749	1390	2227	3260
10	382	954	1782	2866	4206
11	469	1184	2223	3586	5273

The values 3, 5, 7, 9, and 11 in the first row are bases. The values 2 through 11 in the first column are degrees. The other values are the numbers of coefficient corresponding to bases and degrees

The advantages of FIP representation over IP representation are threefold. Firstly, FIPs can accurately describe the complicated objects. The lower degree FIP works as well as the higher degree IP in representing the same objects. As an example, 12th degree IP and the fifth-degree FIP are used to fit the same bear shape (see Fig. 1a), but their fitting results, shown in Fig. 1b, c, respectively, are nearly equivalent, which shows that the FIP has powerful capability to describe complex objects. In fact, the IP of degree less than 12 cannot represent the bear shape efficiently [29]. On the other hand, the number of coefficients of the 12th degree IP is 91. However, the one of fifth degree FIP is 82 (see Table 1). This means that in contrast to IPs, FIPs can use fewer coefficients to represent objects. Secondly, Due to the fractional exponent of the degree, FIP representation has much lower computational complexity and higher numerical stability, which will be further discussed in the Sect. 4. Thirdly, the stabilities of FIP representations are less affected by various preprocessing methods. FIP representations may be viewed as a fitting process consisting of three stages.

1. Transforming coordinates of dataset points to be close to a unit square.
2. Fitting an IP to these transformed dataset points.
3. Transforming the IP to an FIP.

The first item above implies that the FIP representation itself has the ability to scale the dataset points to be close to a unit square. Consequently, various preprocessing methods have less effect on the stabilities of FIP representations.

This paper is organized as follows. Below we discuss the background. Particularly, we review the IP representation, and focus on the Ridge Regression (RR) fitting algorithm. In Sect. 3, we define the FIP and investigate the relation among the number of coefficients, the base, and the degree of FIPs. In Sect. 4, we explore the properties of the FIP, including its high stability and powerful representation and present the comparison of FIPs and IPs in representing complex objects. We develop an algorithm for determination of a moderate degree for an FIP to represent a given object. Section 5 presents simulation results obtained from comparison of FIPs and IPs in representing complex objects. In addition, the experiment on the comparison of contribution of the base and the degree to the fitted FIP is also performed. Section 6 summarizes and concludes the paper.

2 Background

In this section, we provide a brief overview of IP representation, including definition of IPs, IP fitting algorithms, and preprocessing methods.

2.1 Mathematical Formulation

Implicit polynomial in variable two and three can be used to represent 2D curves and 3D surfaces, respectively. In this paper, we only focus on IP curve, because the general properties of IP curve can be also easily extended to those of IP surface.

An IP curve is a planar curve, and it can be specified by the zero set of a 2D IP of degree n given by

$$\begin{aligned} f(x,y) &= \sum_{i+j\le n,\ 0\le i,j} a_{ij}x^i y^j \\ &= a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 \\ &\quad + \dots \\ &\quad + a_{n0}x^n + a_{(n-1)1}x^{n-1}y + \cdots + a_{0n}y^n = 0. \end{aligned} \tag{1}$$

The homogeneous polynomial of degree r in the variables x and y is a form as follows:

$$a_{r0}x^r + a_{(r-1)1}x^{r-1}y + \cdots + a_{0r}y^r$$

particularly, the homogeneous polynomial of degree n is called leading form of IPs.

The IP $f(x, y)$ can also be expressed through the form of the coefficient vector as follows:

$$f(x,y) = X^{\mathrm{T}}A, \tag{2}$$

where

$$A^{\mathrm{T}} = \begin{bmatrix} a_{00} & a_{10} & \cdots & a_{n0} & a_{(n-1)1} & \cdots & a_{0n} \end{bmatrix}$$

and

$$X^{\mathrm{T}} = \begin{bmatrix} 1 & x & y & \cdots & x^n & x^{n-1}y & \cdots & y^n. \end{bmatrix}$$

Denoting the number of coefficients of $f(x, y)$, as well as the dimension of the coefficient vector A, by p, we have $p = (n+1)(n+2)/2$. In addition, the set of all points (x, y) at which the IP $f(x, y)$ is zero is called the zero set of IP $f(x, y)$.

2.2 Fitting Methods

In general, IP representations are obtained through a fitting process, which is to find a coefficient vector that leads to an IP that best fits the dataset points under a criterion to be specified. In the past decades, various methods have been developed, such as 3L method [5], gradient-one method [23], Min-Max method [7], etc. Among them, the Min-Max method is the most stable, which further improves the gradient-one and other algorithms by constraining the gradient vector along the zero set of IPs to have a norm for each point of the dataset. The ridge regression (RR) algorithm is also presented in [23] which applies regularization to the gradient-one algorithm to alleviate the problem of spurious zero sets of the fitted IP. In this work, we choose RR as the fitting algorithm to investigate the properties of FIPs. The RR algorithm is reviewed as follows:

Let Z_k be the $p \times 2$ gradient monomial matrix, N_k and T_k be the local normal unit vector and tangent unit vector at (x_k, y_k), respectively, where $k = 1, 2, \dots, N$ and N is the number of points in the dataset. Then, the monomial matrix can be given as

$$M_0 = \begin{bmatrix} X(x_1,y_1) & X(x_2,y_2) & \cdots & X(x_N,y_N) \end{bmatrix}.$$

And the normal monomial matrix and the tangent monomial matrix are given, respectively, as follows:

$$\begin{aligned} M_1 &= \begin{bmatrix} Z_1N_1 & Z_2N_2 & \cdots & Z_NN_N \end{bmatrix}, \\ M_2 &= \begin{bmatrix} Z_1T_1 & Z_2T_2 & \cdots & Z_NT_N \end{bmatrix}. \end{aligned}$$

Let $b = [\,0_{N\times 1}\ 1_{N\times 1}\ 0_{N\times 1}\,]^{\mathrm{T}}$ and define a new overall monomial matrix $M = [\,M_0\ M_1\ M_2\,]^{\mathrm{T}}$. Then we can formu-

late the fitting criteria as

$$MA = b.$$

By using a linear least squares algorithm to solve the above equation, we can obtain the coefficient vector of the fitted IP curve:

$$A = (M^{\mathrm{T}}M)^{-1}M^{\mathrm{T}}b. \tag{3}$$

Generally, the matrix $M^{\mathrm{T}}M$ is nearly singular [23]. In order to overcome this problem, we usually add a term kD to the matrix $M^{\mathrm{T}}M$, where k is a small positive value called the RR parameter, and D is usually a diagonal matrix, e.g., unit matrix. Then, (3) can be modified as

$$A = (M^{\mathrm{T}}M + kD)^{-1}M^{\mathrm{T}}b. \tag{4}$$

In addition, the detail for how to build D to obtain the fitting invariance refers to [23].

2.3 Preprocessing Methods

In order to obtain stable IP representations, we have to preprocess dataset prior to IP fitting. In general, the preprocessing data consists of two steps. Firstly, locate the dataset center of mass at the origin of the coordinate system. Specifically, for dataset points (x_i, y_i), $i = 1, 2, \ldots, N$, we convert them to new coordinates $(x_i{}', y_i{}')$ using the following formulae.

$$x_i{}' = x_i - \frac{1}{N}\sum_{j=1}^{N} x_j,$$
$$y_i{}' = y_i - \frac{1}{N}\sum_{j=1}^{N} y_j.$$

And secondly, scale each point (x_i', y_i') to be close to value 1 or -1. Specifically, choose a good scale factor S and transfer (x_i', y_i') to $(\bar{x}_i, \bar{y}_i)$ by using the following function.

$$\bar{x}_i = x_i'/S,$$
$$\bar{y}_i = y_i'/S.$$

In recent years, various scale factors have been developed, such as S_{av}, $S_{\max}$, and $S_{75\%}$. S_{av} and $S_{\max}$ are obtained by computing the average Euclidian distance and maximum Euclidian distance of the data points from the origin, respectively [23]. $S_{75\%}$ is the 75th percentile of the distances of the data points from the origin [3].

3 Definition of FIP

In this section, we define the FIP and investigate the relation among the number of coefficients, bases, and degrees of FIPs.

3.1 Formulation

Fractional implicit polynomial (FIP), which is extended from the IP, is an implicit function defined in a multivariate fractional polynomial. The major difference from IPs is that the degree of FIP is not an integer but a fraction. In this paper, we focus on 2D FIP, which is defined as

$$f_{nm}(x, y) = \sum_{\substack{m \le i+j \le nm \\ i,j=0 \text{ or } i,j \ge m}} a_{i/m\ j/m} x^{i/m} y^{j/m}, \tag{5}$$

where both n and m are integers greater than value 1. Furthermore, m is odd. We call $f_{nm}(x, y)$ FIP of degree n and base m. Besides, the items of FIPs are called fractional monomials. When m is known, we also call $f_{nm}(x, y)$ FIP of degree n. Obviously, 3D FIP can be easily extended according to (5).

Moreover, the FIP $f_{nm}(x, y)$ can also be rewritten in a vector form as follows

$$f_{nm}(x, y) = X^{\mathrm{T}}A, \tag{6}$$

where

$$\begin{aligned} A^{\mathrm{T}} = \big[& a_{0/m\ 0/m}\ a_{m/m\ 0/m}\ a_{0/m\ m/m}\ a_{(m+1)/m\ 0/m} \\ & a_{0/m\ (m+1)/m}\ \cdots\ a_{2m/m\ 0/m}\ a_{m/m\ m/m} \\ & a_{0/m\ 2m/m}\ \cdots\ a_{nm/m\ 0/m}\ a_{(nm-m)/m\ m/m} \\ & a_{(nm-m-1)/m\ (m+1)/m}\ \cdots \\ & a_{(m+1)/m\ (nm-m-1)/m}\ a_{m/m\ (nm-m)/m} \\ & a_{0/m\ nm/m}\big] \end{aligned}$$

and

$$\begin{aligned} X^{\mathrm{T}} = \big[& 1, x^{m/m}, y^{m/m}, x^{(m+1)/m}, y^{(m+1)/m}, \ldots, \\ & x^{2m/m}, x^{m/m}y^{m/m}, y^{2m/m}, \ldots, x^{nm/m}, \\ & x^{(nm-m)/m}y^{m/m}, x^{(nm-m-1)/m}y^{(m+1)/m}, \ldots, \\ & x^{(m+1)/m}y^{(nm-m-1)/m}, x^{m/m}y^{(nm-m)/m}, \\ & y^{nm/m}\big]. \end{aligned}$$

As an example, we write the formula for $f_{n3}(x, y)$ according to (5), which can also be written in the above vector form.

$$
\begin{aligned}
f_{n3}(x,y) = {} & a_{0/3\,0/3} + a_{3/3\,0/3}x^{3/3} + a_{0/3\,3/3}y^{3/3} \\
& + a_{4/3\,0/3}x^{4/3} + a_{0/3\,4/3}y^{4/3} \\
& + a_{5/3\,0/3}x^{5/3} + a_{0/3\,5/3}y^{5/3} \\
& + a_{6/3\,0/3}x^{6/3} + a_{3/3\,3/3}x^{3/3}y^{3/3} \\
& + a_{0/3\,6/3}y^{6/3} + \cdots \\
& + a_{3n/3\,0/3}x^{3n/3} + a_{(3n-3)/3\,3/3}x^{(3n-3)/3}y^{3/3} \\
& + \cdots \\
& + a_{3/3\,(3n-3)/3}x^{3/3}y^{(3n-3)/3} + a_{0/3\,3n/3}y^{3n/3}.
\end{aligned}
$$

In order to further reveal the structure of each fractional monomial of $f_{n3}(x, y)$, we do not simplify their degrees. For example, $x^{3/3}$, $y^{3/3}$, and $x^{6/3}$ do not simplify to x, y, and x^2, respectively. Observing the above formula for the FIP, we can find that FIPs have three features:

1. The degree of each variable in FIPs is an improper fraction. Observing the FIP $f_{n3}(x, y)$, we can find that its degrees of x and y in each fractional monomial are greater than value 1. This implies that the first-order partial derivatives of $f_{n3}(x, y)$ with respect to x or y are polynomials rather than rational polynomials, which ensures most of algorithms for IPs, such as fitting algorithm, can be used for FIPs without any changes. For example, for two given functions of two variables $f(x, y) = x^{4/3}y$ and $g(x, y) = x^{2/3}y$, their first-order partial derivatives with respect to x should be $f_x(x, y) = (4/3)x^{1/3}y$ and $g_x(x, y) = (2/3)x^{-1/3}y$, respectively. Clearly, $g_x(x, y)$ is a rational polynomial, which implies that the variable x can not take value zero. However, many fitting algorithms, such as Min-Max method, require the first-order partial derivative of every fractional monomial of FIPs is continuous on every point in the coordinate plane. Consequently, the first-order partial derivatives of fractional monomials of FIPs must not be rational polynomials. That is, the degree of each variable in FIPs is required to be an improper fraction.
2. For FIPs and IPs of the same degrees, the number of coefficients of the FIP is more than that of the IP, which means FIPs can be provided with more information. For example, an IP curve of degree 2 is a conic, but an FIP curve of degree 2 is a more complex curve than a conic.
3. The base of FIPs is an odd number. If the base of FIPs were an even number, then we could not apply FIPs in real domain. For example, supposed that there exists an FIP $f_{n4}(x, y)$, the base of which is 4, then the two variables x and y of the FIP only take positive real values or zeros, which limits FIPs to represent most of objects.

Note that many fitting methods for IPs, such as 3L, RR, and Min-Max, can also be directly used to fit FIPs without any changes.

3.2 The Number of Coefficients

In order to obtain the general formula for the number of coefficients of the FIP $f_{nm}(x, y)$, we first investigate the coefficients of the FIP $f_{n3}(x, y)$ of degree n and base 3, particularly, the regulatory variation of their indexes of coefficients. Then, we can compute the number of coefficients of the FIP $f_{n3}(x, y)$ as follows:

$$p_3 = 2(3-1) + (1+2+3+\cdots+3n-3).$$

In the similar way, the number of coefficients of $f_{n5}(x, y)$ and $f_{n7}(x, y)$ can also be given, respectively, by

$$p_5 = 2(5-1) + (1+2+3+\cdots+5n-7)$$

and

$$p_7 = 2(7-1) + (1+2+3+\cdots+7n-11).$$

Similarly, the general formula for the number of coefficients of the FIP $f_{nm}(x, y)$, as well as the dimension of vector A in (6) can be written as follows:

$$p = 2(m-1) + (1+2+3+\cdots+m(n-2)+3)$$

or, equivalently

$$p = \frac{1}{2}m^2(n-2)^2 + \frac{7}{2}mn - 5m + 4, \tag{7}$$

where m and n are base and degree of $f_{nm}(x, y)$, respectively. Note that both n and m are integers greater than value 1.

Before representing a dataset using an FIP, both its degree and its base must be determined. It is clear from (7) that the number of coefficients depends on these two parameters. Note that too many coefficients will limit application of FIPs. Consequently, it is of importance to choose a moderate degree and base for an FIP so that it has both high representation power and as few coefficients as possible.

Table 1 lists the number of coefficients p corresponding to degrees n when bases m are 3, 5, 7, 9, and 11, respectively. Clearly, for base 3, base 5, and base 7, the numbers of their corresponding coefficients all tend to increase slowly as degrees increase, but for base 9 and 11, the opposite is true. This implies that we choose high degrees to improve the representation power of FIPs only when bases are 3, 5, or 7. On the other hand, when the degrees are 2, 3, or 4, the numbers of coefficients do not increase rapidly as bases increase. For example, the numbers of coefficients of second

and third degree FIPs are only 26 and 125, respectively, when both their bases are 11.

In general, if further improving FIP representations, we can increase either bases or degrees of FIPs according to the following principle: For low degrees, such as degree 2 and 3, we may increase bases, but for high degrees, only base 3 and 5 are suggested to choose. In other words, high degrees and high bases should not be simultaneously chosen for FIPs to represent objects under the requirement of few coefficients. However, regardless of the number of coefficients, we can greatly improve representation power of FIPs by increasing both its degree and its base.

4 Properties of FIP Representation

In this section, we investigate the properties of the FIP, including its high stability and powerful representation and give the comparison of FIPs and IPs in representing the objects. In addition, we develop an algorithm for determination of a moderate degree for an FIP to represent a given object, which can be obtained by only computing the number of stationary points on the objects. Lastly, the computational cost of FIPs is analyzed.

4.1 The Stability of FIPs

As discussed in introduction, the stability of IPs mainly depends on the degree of IPs and preprocessing method for the same fitting algorithm. In general, the higher the degree is, the more instable the IP representation becomes. Consequently, in order to reduce instability of representation, one should choose as low degree for an IP as possible. However, the lower degree IP cannot represent complex objects. Therefore, we extend IPs to FIPs to overcome these problems.

The stability problem of FIP representation means the high sensitivity of zero set to small changes in the values of the FIP coefficients. In other words, if FIP representation is stable, then, small changes (errors) in its coefficients only result in small changes in the location of its zero set points. Conversely, if FIP representation is unstable, then small changes (errors) in its coefficients result in large changes in the location of its zero set points. Specifically, for a given point (x_i, y_i) on zero set of $f(x, y)$, its small change implies that the point moves small distance in the direction that is perpendicular (normal) to the zero set. Denoting this distance and the variation of coordinate of the point (x_i, y_i) by Δd and $(\Delta x, \Delta y)$, respectively, we have

$$\Delta x = \Delta d \cos\theta_i, \quad \Delta y = \Delta d \sin\theta_i, \tag{8}$$

where

$$\cos\theta_i = f_x(x_i, y_i) \Big/ \sqrt{f_x^2(x_i, y_i) + f_y^2(x_i, y_i)},$$
$$\sin\theta_i = f_y(x_i, y_i) \Big/ \sqrt{f_x^2(x_i, y_i) + f_y^2(x_i, y_i)},$$

and the vector $(\cos\theta_i, \sin\theta_i)$ is the gradient vector of the function $f(x, y)$ at the point (x_i, y_i).

In order to analyze the stability of FIP representation, we need to examine how small changes in the coefficient values affect the location of a point on the zero set. Then, we have the following theorem.

Theorem 1 *Supposed that $f(x, y) = 0$ is an FIP curve of degree n and base m which is accurately fitted to a given shape, then the fitted FIP would become more stable as m increases.*

Proof According to (6), we can rewrite the coefficient vector A^{T} in (6) as $[a_1\ a_2\ a_3\ \cdots\ a_{p-1}\ a_p]$, where a_i is the ith element from A^{T} and p is the number of coefficients of the FIP curve $f(x, y) = 0$, as well as the dimension of A^{T}. Regarding x, y, and a as variables of $f(x, y)$ and expanding $f(x, y, a)$ in the point (x_i, y_i, a_j) by a first-order Taylor series approximation, it follows that

$$\begin{aligned} &f(x_i + \Delta x, y_i + \Delta y,\ a_j + \Delta a) \\ &\quad = f(x_i, y_i, a_j) + f_x(x_i, y_i, a_j)\Delta x \\ &\qquad + f_y(x_i, y_i, a_j)\Delta y + f_a(x_i, y_i, a_j)\Delta a. \end{aligned}$$

Considering $f(x_i, y_i, a_j) = 0$, $f_a(x_i, y_i, a_j) = x_i^{u/m} y_i^{v/m}$, where $x_i^{u/m} y_i^{v/m}$ is the fractional monomial corresponding to the coefficient a_j. Hence, we can obtain

$$f_x(x_i, y_i, a_j)\Delta x + f_y(x_i, y_i, a_j)\Delta y + x_i^{u/m} y_i^{v/m}\Delta a = 0.$$

Substituting Δx and Δy in (8) into the preceding equation, then we have

$$\begin{aligned} &f_x(x_i, y_i, a_j)\Delta d\cos\theta_i + f_y(x_i, y_i, a_j)\Delta d\sin\theta_i \\ &\quad + x_i^{u/m} y_i^{v/m}\Delta a = 0. \end{aligned}$$

Considering that $(\cos\theta_i, \sin\theta_i)$ is the gradient of the function $f(x, y)$ at the point (x_i, y_i), simplifying the above equation, then we have

$$\frac{\Delta d}{\Delta a} = \frac{-(x_i^u y_i^v)^{1/m}}{\sqrt{f_x^2(x_i, y_i) + f_y^2(x_i, y_i)}}. \tag{9}$$

Noting that m is less than u and v. Let $u = d_1 m + r_1$ and $v = d_2 m + r_2$, where r_1 and r_2 are remainders, d_1 and d_2 are quotients.

It is clear from (5) that $d_1+d_2 \leq n$, both r_1 and r_2 are less than m. Then, we have $x_i^{u/m} y_i^{v/m} = (x_i^{d_1} y_i^{d_2})(x_i^{r_1} y_i^{r_2})^{1/m}$, and hence, the fractional monomials $x_i^{u/m} y_i^{v/m}$ would approach $x_i^{d_1} y_i^{d_2}$ as m goes to infinity.

On the other hand, since the FIP curve $f(x, y) = 0$ accurately fits the given shape, then, the gradient norm of $f(x, y) = 0$ at each point (x_i, y_i) on this curve, namely $\sqrt{f_x^2(x_i, y_i) + f_y^2(x_i, y_i)}$, is a constant value. The reason is that the given shape is fixed and the gradient norms of the fitted FIP $f(x, y)$ are constrained to have the same values as the estimated gradient norms from this shape [23]. This implies that the gradient norms $\sqrt{f_x^2(x, y) + f_y^2(x, y)}$ are independent of base m. That is, when m goes to infinity, we have

$$\frac{\Delta d}{\Delta a} = \frac{-x_i^{d_1} y_i^{d_2}}{\sqrt{f_x^2(x_i, y_i) + f_y^2(x_i, y_i)}}. \tag{10}$$

Since both x_i and y_i have been preprocessed to be close to values 1.0 or -1.0 before fitting FIP to a given shape, we can see that $x_i^{d_1} y_i^{d_2}$ is closer to values 1.0 or -1.0 than $x_i^{u/m} y_i^{v/m}$. This means that the effect of a small coefficient perturbation Δa on the location of the points on zero set will decrease as the base m increases. That is, an FIP would become more stable as its base m increases. □

From the above theorem, we also find that in contrast to IP representation, the change in the coefficient values of FIPs is less affected by the change in the location of points on the zero set due to the presence of bases of FIPs. The main reason is that FIPs can use less degree than IPs to represent a given shape.

To gain further insight into the stability of an FIP of degree n and base m, we give an example to demonstrate how the coordinates of dataset points are transformed to be closer to value 1 or -1 by computing their mth root in fitting FIP process. Both Fig. 2a, b show bear shape (solid line) normalized by scale factor S_{av} and the unit square (dotted line). Obviously, the normalization only transforms the coordinates of the points on the original bear shape to be close to the unit square without changing original shape of the bear. However, the transformed shape by computing mth root of coordinates is just the opposite. As shown in Fig. 2a, b, the deformed bear shapes with bold solid line are obtained by computing third root and fifth root of coordinates of the bear shape, respectively. Obviously, the latter with bold solid line shown in Fig. 2b is more similar to unit square than the former with bold solid line shown in Fig. 2a. In order to estimate the similarity of the transformed bear shapes with mth root ($m = 1, 3, 5, 7, 9$) against the unit squares, we compute the distances between them. As illustrated in Fig. 2c, the distance decreases rapidly as the root m increases. Specifically,

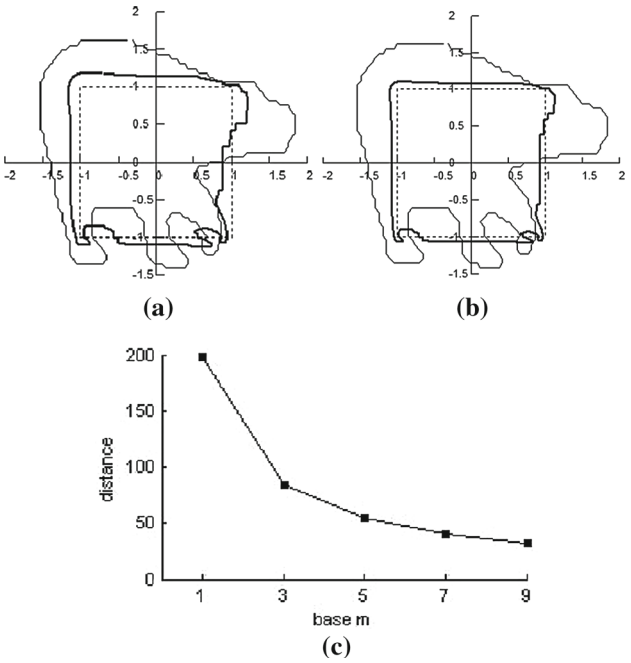

Fig. 2 **a** Bear shape transformed with third root (*bold solid line*). **b** Bear shape transformed with fifth root (*bold solid line*). Both **a** and **b** share the same bear shapes (*solid line*) normalized with Sav and unit square (*dotted line*). **c** The distances from unit square to the bear shape transformed with first, third, fifth, seventh and ninth roots, respectively

the distances are 197.63, 84.71, 54.9, 40.71, and 32.36 when the bases m are 1, 3, 5, 7, and 9, respectively. This implies that as the base increases, the coordinates of transformed dataset points are more and more close to value 1 or -1, which makes the fitted FIP more and more stable.

4.2 The Representation Power of FIPs for Objects

It is necessary to review the properties of IPs in order to explore the representation power of FIPs. The reason is that an FIP is generalized from an IP. The following theorem demonstrates the representation power of IPs mainly depends on its degree.

Theorem 2 *If C_1 and C_2 are zero sets of the IPs of degree n_1 and n_2 that do not share a common component, they can intersect in at most $n_1 n_2$ points* [1].

This is the famous Bezout theorem in algebraic geometry. As an example, it is noted in [11] that a pentagon cannot be represented by a fourth-degree IP curve. The reason is the following: If the pentagon can be fitted by a fourth-degree IP curve, then there always exists a conic that intersects two sides of each angle of the pentagon. This implies the conic intersects the fitted fourth-degree IP curve in 10 points. However, according to Theorem 2, a conic has to intersect a fourth-degree IP curve in at most 8 points. Consequently, a pentagon cannot be represented by a fourth-degree IP curve,

but can be done by at least a fifth-degree one. Thus, Theorem 2 illustrates the relation between degrees of the two IP curves and the number of their intersection points. Similarly, for an FIP curve, we have the following theorem.

Theorem 3 *Supposed that C_1 and C_2 are, respectively, zero sets of the FIPs of degree n_1 and base m, degree n_2 and base m which do not share a common component, then the two zero sets intersect in at most $n_1 n_2 m^2$ points.*

Proof Let C_1 be the zero set of the FIP $f(x, y)$ and C_2 be the zero set of the FIP $g(x, y)$. We can write the formula for $f(x, y)$ according to (5) as follows

$$f(x, y) = \sum_{\substack{m \le i+j \le n_1 m \\ i,j=0 \text{ or } i,j \ge m}} a_{i/m\ j/m} x^{i/m} y^{j/m}. \tag{11}$$

Similarly, we can also obtain the formula for $g(x, y)$ according to (5)

$$g(x, y) = \sum_{\substack{m \le i+j \le n_2 m \\ i,j=0 \text{ or } i,j \ge m}} b_{i/m\ j/m} x^{i/m} y^{j/m}. \tag{12}$$

Defining $s = x^{1/m}$ and $t = y^{1/m}$, substituting s and t for x and y, respectively, then (11) and (12) are converted to implicit polynomials, which are denoted by $p(x, y)$ and $q(x, y)$, respectively.

$$p(x, y) = \sum_{\substack{m \le i+j \le n_1 m \\ i,j=0 \text{ or } i,j \ge m}} a_{i/m\ j/m} s^i t^j,$$

$$q(x, y) = \sum_{\substack{m \le i+j \le n_2 m \\ i,j=0 \text{ or } i,j \ge m}} b_{i/m\ j/m} s^i t^j.$$

We can see that $p(x, y)$ and $q(x, y)$ are implicit polynomials of degree $n_1 m$ and $n_2 m$, respectively. According to Theorem 2, their zero sets can intersect in at most $n_1 n_2 m^2$ points. That is, the zero sets of $f(x, y)$ and $g(x, y)$ also intersect in at most $n_1 n_2 m^2$ points. □

It is clear from the two above theorems that the number of intersection points of the two FIP curves is square their base times that of the two corresponding IP curves when their degrees are the same.

4.3 The Determination of Degree of FIPs

As noted in [11], if IPs of arbitrary degree are allowed, then every object can be described, but it is better to work with degrees as low as possible. Clearly, the same goes for FIPs. Thus, we need to choose a moderate degree for an FIP so that it can represent a given object efficiently. The general methods are to find a moderate degree by trying different degrees several times and selecting the best one from the results [29]. In this section, we present a method to determine the degree range only based on the shape of the given dataset.

In order to determine a moderate degree for an FIP, we need to explore its properties. Obviously, some of these properties can be extended from those of IPs, such as boundedness, connectivity, etc.

Boundedness implies that the zero set of FIPs is bounded, and connectivity means the zero set of the FIP does not include disconnected components or does not intersect itself. In general, FIPs are used to represent closed and bounded curves, rather than open curves. Therefore, the fitted FIP curve is restricted to be bounded and connective.

As noted in [11,25], if zero set of an IP is bounded, then, degree of the IP must be even. In other words, zero set of the IP of odd degree must be unbounded. Besides, if the zero set of an IP does not intersect itself, then the first derivatives of the IP with respect to x and y both are not zeros simultaneously. Obviously, comparing (1) with (5), we can arrive at the conclusion: the FIP also has these two properties.

For a given dataset, e.g., an object's boundary, determination of the moderate degree of the fitted FIP can only depend on the structure of the dataset, especially on the features of the boundary, such as stationary points and corner points. An stationary point is an input to a function where the derivative is zero (equivalently, the slope is zero): where the function stops increasing or decreasing. The following theorem shows how to use the number of stationary points to determine the degree range for an FIP.

Theorem 4 *Supposed that the number of stationary points in the boundary of a given object B is s, and the FIP of degree n and base m fits B best. Then relation between n and s is given by*

$$n \ge \left(1 + \sqrt{1 + 4s/m^2}\right)/2. \tag{13}$$

Proof For the boundary of object B, we need to use a bounded and connective FIP to represent it because B is closed and connective. Let the fitted FIP curve be denoted by $f(x, y) = 0$ and then we can obtain the slope of the tangent line to the curve at each point according to the slope function $-f_x(x, y)/f_y(x, y)$. On the other hand, due to the connectivity of the curve, $f_x(x, y)$ and $f_y(x, y)$ both are not zeros simultaneously. Without loss of generality, let $f_y(x, y) \neq 0$, we can obtain stationary points on the curve through solving the following system of equations.

$$\begin{cases} f(x, y) = 0 \\ f_x(x, y) = 0. \end{cases} \tag{14}$$

It is clear from (5) that $f_x(x, y)$ is the FIP of degree $n-1$ and base m. According to Theorem 3, there exist at most $n(n-1)m^2$ solutions to (14). In other words, there exist at most $n(n-1)m^2$ stationary points on the FIP curve $f(x, y) = 0$. Hence, we have the following inequality:

$$n(n-1)m^2 \geq s.$$

Solving the above inequality, we have (13). □

The above theorem provides a way to determine the degree of an FIP in representing a given object. In general, the base m takes the smaller value, such as 3 or 5. For example, $n \geq (1+\sqrt{1+4s/9})/2$ when $m = 3$. In order to force the fitted FIP curve to be bounded, the degree is required to be even number. Thus, the lower bound for the degree is the smallest even number greater than $(1+\sqrt{1+4s/9})/2$. Moreover, to further ensure that the fitted FIP is of even degree and efficient to represent the given dataset, we can modify (13) as follows:

$$n = \begin{cases} 2 & s < 7 \\ \text{even } (2+(1+\sqrt{1+4s/m^2})/2) & s \geq 7, \end{cases} \tag{15}$$

where even is a function and its returned value is the even number nearest to $2+(1+\sqrt{1+4s/m^2})/2$.

It is clear from (15) that we can find if $7 \leq s < 34$, then the expression $2+(1+\sqrt{1+4s/9})/2$ ranges from 3.51 to 4.48. Hence, the lower bound for the degree should be 4. That is, if the number of stationary points on the boundary of a given object is less than 34, then we should choose an FIP of degree greater than or equal to 4 to fit the object. Similarly, the degree greater than or equal to 6 should be chosen for fitted FIP when $34 \leq s < 78$ and degree 8 when $79 \leq s < 141$.

Before applying (15) to determine the moderate degree for the fitted FIP, we need to obtain the number of stationary points in the given dataset. Hence, estimation of the slope is required from the scattered data, which is a basic problem in curve and surface approximation. Generally, we can fit a regression line to the points in the neighborhood of the given point and regard the slope of the line as that of the dataset at this point [7]. Afterward, we can obtain all stationary points in the dataset by finding the points where the slopes take value 0.

Note that if $m = 1$, then (13) can be modified as $n \geq (1+\sqrt{1+4s})/2$. This inequality provides a way to determine degree for the IP curve.

In summary, the algorithm for determination of degree of FIPs can be simply described as follows:

1. Fitting a regression line to the points in the neighborhood of each point in the given dataset and calculating the slope of this line.
2. Counting the number of the points where the slopes take value 0. Then viewing it as the number of stationary points.
3. According to (15), calculating the degree and choosing it as the candidate degree of the fitted FIP.

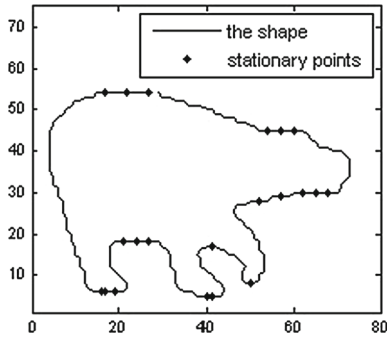

Fig. 3 Detection of stationary points from the shape of a bear

We named the above algorithm shape-degree algorithm.

As an example, Fig. 3 shows the shape of a bear and its stationary points obtained with above algorithm. It can be seen that the number of the stationary points is 21. According to Theorem 4, if setting the value of base to be 3, then the degree of the fitted FIP should be greater than or equal to 4. Figure 1c shows the fitted zero set of the FIP of degree 5, which is very accurate and stable for fitting the bear shape. Note that the algorithm is insensitive to the errors of the number of the detected stationary points. For example, the lower bound of the degree is 6 if only the number of the detected stationary points belongs to the interval [34,78).

Although the above algorithm only provides a lower bound for degree for the fitted FIP curve, we can find a moderate degree by starting to fit an FIP curve from this lower bound rather than from degree 2. In most case, the moderate degree for the fitted FIP curve usually is equal to this lower bound, which is further illustrated in detail in Sect. 5.

4.4 Computational Cost

The complexity of evaluating FIP representation mainly depends on the total number of multiplications required to evaluate all its fractional monomials. In order to clearly show lower computational complexity of evaluating an FIP, we only analyze the complexity of the FIP of base 3.

Let $q = x^{1/3}$, $r = x^{2/3}$, $s = y^{1/3}$, and $t = y^{2/3}$. The formula for $f_{n3}(x, y)$ can be rewritten as

$$
\begin{aligned}
f_{n3}(x, y) = {} & a_{0/3\ 0/3} + a_{3/3\ 0/3}x + a_{0/3\ 3/3}y \\
& + a_{4/3\ 0/3}xq + a_{0/3\ 4/3}ys \\
& + a_{5/3\ 0/3}xr + a_{0/3\ 5/3}yt \\
& + a_{6/3\ 0/3}x^2 + a_{3/3\ 3/3}xy + a_{0/3\ 6/3}y^2 \\
& + \cdots .
\end{aligned} \tag{16}
$$

Observing each of the homogeneous fractional polynomials of $f_{n3}(x, y)$, defining $b_{i/3}$ to denote the total number of multiplications required to evaluate the homogeneous fractional polynomial of degree $i/3$, then we have

$$
\begin{aligned}
& b_{3/3} = 0 + 0,\ b_{4/3} = 1 + 1,\ b_{5/3} = 1 + 1, \\
& b_{6/3} = 1 + 1 + 1, \\
& b_{7/3} = 2 + 2 + 2 + 2 = 2 \times 4 + 0, \\
& b_{8/3} = 2 + 2 + 3 + 2 + 2 = 2 \times 5 + 1, \\
& b_{9/3} = 2 + 2 + 3 + 3 + 2 + 2 = 2 \times 6 + 1 \times 2.
\end{aligned}
$$

In the similar way, we can obtain the total number of multiplications required to evaluate the homogeneous fractional polynomials of degree $10/3$, $11/3$, and $12/3$, respectively, as follows:

$$
\begin{aligned}
& b_{10/3} = 3 \times 7 + 0, \\
& b_{11/3} = 3 \times 8 + 2 \times 1, \\
& b_{12/3} = 3 \times 9 + 2 \times 2.
\end{aligned}
$$

In general, for degree $(3i-2)/3$, $(3i-1)/3$, $(3i-0)/3$, we have

$$
\begin{aligned}
& b_{(3i-2)/3} = (i-1) \times (3i-5), \\
& b_{(3i-1)/3} = (i-1) \times (3i-4) + (i-2), \\
& b_{(3i-0)/3} = (i-1) \times (3i-3) + (i-2) \times 2.
\end{aligned}
$$

Defining c_i to denote the sum of the above three terms, namely, $c_i = b_{(3i-2)/3} + b_{(3i-1)/3} + b_{(3i-0)/3}$, then we have

$$
c_i = (i-1) \times (9i-12) + 3(i-2)
$$

or, equivalently,

$$
c_i = 9i^2 - 18i + 6.
$$

Sum all c_i, $i = 3, 4, \ldots, n$ to obtain

$$
\sum_{i=3}^{n} c_i = 3n^3 - 9n^2/2 - 3n/2 - 3.
$$

Adding $b_{3/3} + b_{4/3} + b_{5/3} + b_{6/3}$ (equal to 7) to above formula, we can obtain the formula for the total number of multiplications required to evaluate (16) as follows:

$$
3n^3 - 9n^2/2 - 3n/2 + 4, \tag{17}
$$

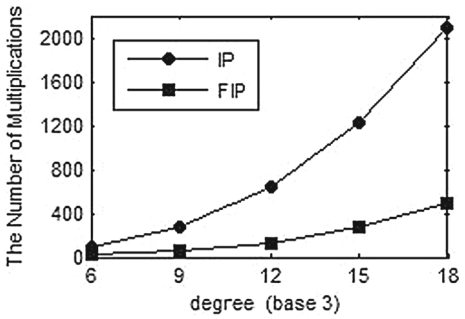

Fig. 4 A comparison between FIPs and IPs with respect to the number of multiplications

where $n \geq 2$.

In addition, evaluation of $x^{1/3}$ can be implemented by expanding it at $x = 1$ or $x = -1$ by a third-order Taylor series approximation because the dataset points (x, y) have been scaled to be close to value 1 or -1. Accordingly, the evaluation of $x^{1/3}$ requires 3 multiplications. In the same way, evaluation of $y^{1/3}$ requires 3 multiplications as well, and both $x^{2/3}$ and $y^{2/3}$ require 4, respectively. Then, the evaluations of q, r, s, and t in (16) require 14 multiplications in all.

Adding 14 to (17), it follows that the formula for the total number of multiplications (denoted by S_{FIP}) required to evaluate FIP $f_{n3}(x, y)$ is

$$
S_{\mathrm{FIP}} = 3n^3 - 9n^2/2 - 3n/2 + 18.
$$

On the other hand, for complexity of evaluating an IP $f_n(x, y)$, it is easy to obtain the required total number of multiplications (denoted by S_{IP}):

$$
S_{\mathrm{IP}} = n^3/3 - n^2/2 - 5n/6.
$$

As described in Sect. 4.2, an FIP of degree n and base 3 shares approximately the same power as an IP of degree $3n$ in representing a given object. In order to show the lower computational cost of the FIP, we make a comparison of evaluation complexity between the FIP and the IP in the case of the same representation power. Figure 4 illustrates this comparison result.

It is clear from the comparison result that the FIP fitting needs much lower computational cost than the IP in the case of the same representation power. For example, evaluation of the IP of degree 18 requires more than 2000 multiplications. However, the corresponding FIP only requires around 400 multiplications.

5 Experimental Results

In this section, we illustrate the effectiveness of FIP representation through competing it with IP representation under

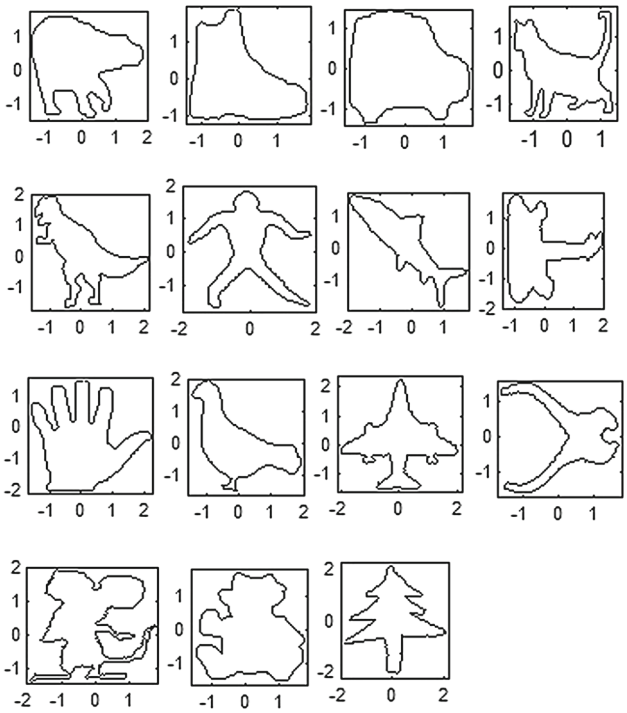

Fig. 5 Objects used in the experiments

the same fitting algorithm. To make comparative evaluation possible, we set our experiments in some precondition.

1. All of datasets of objects are normalized by S_{av} methods, namely, centering the dataset center of mass at the origin of the coordinate system and scaling it by dividing each point by the average length from point to origin.
2. The RR algorithm is used as fitting method and fitting results are evaluated using the fitting error (FE) computed for every single point [17]. In the following experiments, we use accumulated FE(AFE) where $\text{AFE} = \sum_{i=1}^{N} \text{FE}_i$ as a quantitative criterion for comparison.
3. The bases of FIPs used in the experiments take value 3 unless otherwise specified.

5.1 Comparison of Fitting Stability

In order to access the fitting stability of the FIP representation, we conducted the experimental evaluation on various objects (Fig. 5) including real-world objects and artificial free-form shapes ranging from simple to complex. Particularly, the powerful representation of FIPs can be shown by accurately representing the complex objects, such as hands, rats, and cats. Note that in the following experiments, we choose RR algorithm to fit FIPs and IPs.

These objects shown in Fig. 5 were obtained from the Laboratory for Engineering Man/Machine System (LEMS), Brown University. Before representing these objects using FIPs, we used the shape-degree algorithm to determine the candidate degree for each of them. The degrees obtained by using the algorithm are shown in the second column in Table 2.

We fitted the FIPs with these candidate degrees to the objects, and found that only degree 4 failed to work well for bears and hands; degree 6 failed to do for rats. For example, use of the degree 4 to fit the bear would lead to undesired inaccuracy as showed in Fig. 11a. Accordingly, we increased the degrees to 6 and 8 shown in the parentheses in Table 2, respectively. Note that the 5th degree FIP can represent the bears, hands, and planes well, but to ensure the FIP to be of even degree, the degree 6 is chosen. The Fitting results are shown in Fig. 6. It is clear that most of the degrees are moderate ones for FIPs to fit these objects, which indicates the shape-degree algorithm is efficient, and its most advantage is that the degrees of FIPs can be determined only based on the shapes of the given objects before fitting.

Figure 6 shows the fitted FIP curves for all objects in Figs. 5, and 7 shows the fitted IP curves with the corresponding degrees. Their AFEs are shown in the third column and the fifth column in Table 2, respectively. It is clear from the fitting results and corresponding AFEs that for all the shapes, the FIP representation performs consistently better than IP representation, both visually and in terms of the error measures used. Especially for the complex shapes, the FIP exhibits higher representation power. For example, although bears, cats, and hands are all complex shapes, only sixth-degree FIP can accurately represent them. The AFEs for the three shapes were only 0.0616, 0.4527, and 0.1745, respectively. Furthermore, few extraneous components appear in the zero set of the three fitted FIPs, which illustrates the high stability of FIPs. However, as shown in Fig. 7, the same degree IPs failed to represent these three shapes. The works in [28] and [5] note that the bears require at least 12th degree IP for fitting and hands require 18th degree one. In fact, high-degree IP will lead to not only high computational cost but also high numerical instability. That is, high-degree IP is not suitable to represent complex objects.

Because FIP of degree n and base 3 shares the approximately same power as IP of degree $3n$ in representing a given object in theory, for the sake of fair comparison, we chose IPs with their degrees three times the ones of the fitted FIPs shown in Fig. 6 to represent the corresponding shapes. That is, we fitted the IPs of degree 18 to bears, cats, dinos, hands, planes, and rats, and degree 12 to other shapes. The fitting results are shown in Fig. 8 and the corresponding AFEs are shown in the fourth column in Table 2. Observing the AFEs in the fourth column, although the AFEs obtained from the fitting IPs to cars, cats, pliers, and teddies are smaller than those from the corresponding FIPs, however, from the resulting zero sets of these IPs, we can see that IPs unstably represented bears, cats, dudes, planes, pliers, teddies, and

Table 2 Comparison of fitting error for FIP and IP representations of the same degree

Shape name	Degree	AFE for IP	AFE for IP (3*degree)	AFE for FIP
Bear	4(6)	2.2629	0.1108	0.0616
Boot	4	1.2917	0.0406	0.1513
Car	4	6.2977	0.0294	0.1078
Cat	6	13.5692	0.3179	0.4527
Dino	6	27.6609	4.2048	1.3908
Dude	4	13.9523	0.0833	0.0735
Fish	4	3.9886	0.2189	0.9172
Guitar	4	22.4822	0.1555	0.4526
Hand	4(6)	78.5990	5.4993	0.1745
Pigeon	4	0.7907	0.1221	0.0628
Plane	6	10.7351	11.2386	0.4176
Pliers	4	65.4549	0.0942	0.3315
Rat	6(8)	22.0368	6.3061	2.9503
Teddy	4	17.5258	0.1616	1.2277
Tree	6	12.9109	2.6381	0.5348

The degree of the fitted FIP for each shape is obtained from (15), except the numbers in the parentheses which are actual degrees of fitted FIPs. IP and FIP are fitted to each shape by the same RR method. Their corresponding degrees and AFEs are shown in second, third, fourth, and last column, respectively

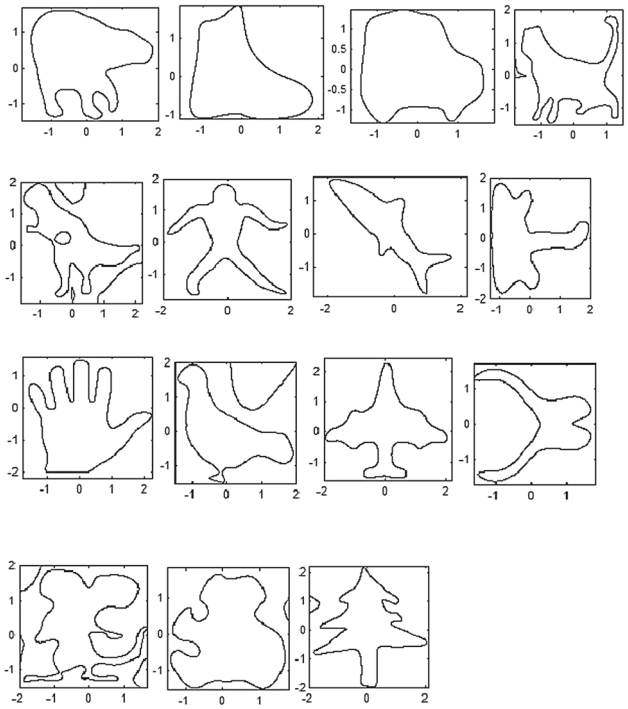

Fig. 6 Using RR fitting algorithm, the results of fitting FIPs to shapes shown in Fig. 5 and their degrees corresponding to the shapes are shown in Table 2

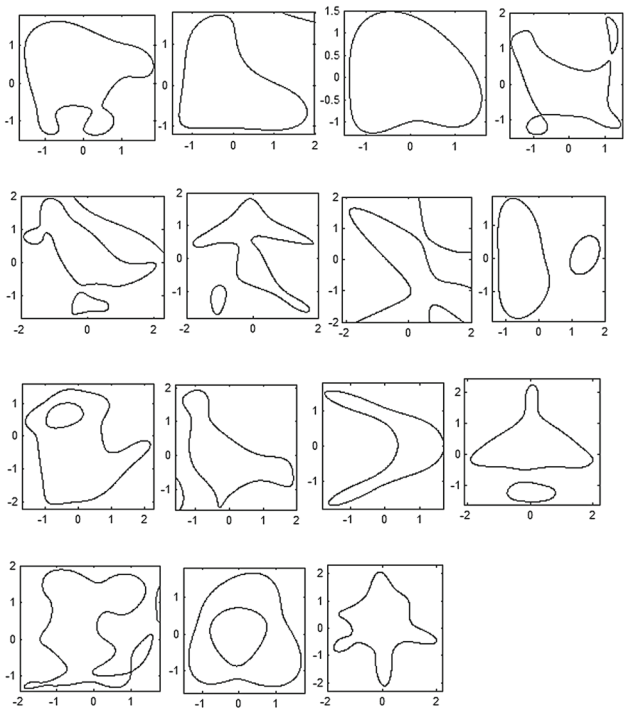

Fig. 7 Using RR fitting algorithm, the results of fitting IPs to shapes shown in Fig. 5, and their degrees corresponding to the shapes are shown in Table 2

trees, particularly, failed to represent dinos, hands, and rats, and only accurately represented the simple objects, such as boots, cars, fishes, and guitars. Comparing the fitting results in Fig. 7 and in Fig. 8, it appears that the use of higher degree IPs has improved the accuracy of representation, but leads to much more instability, even failure in representing complex shapes. On the other hand, comparing the fitting results in Fig. 6 and in Fig. 8, we can see that both accuracy and stability of FIP representation are significantly better than the ones of IP representation especially for complex shapes.

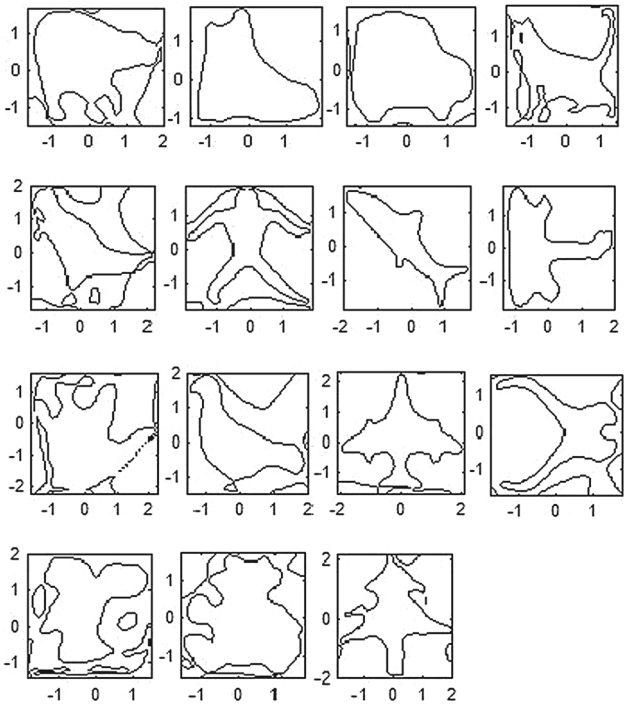

Fig. 8 Using RR fitting algorithm, the fitting results of IPs with their degrees three times the ones of the corresponding fitted FIPs in Fig. 6

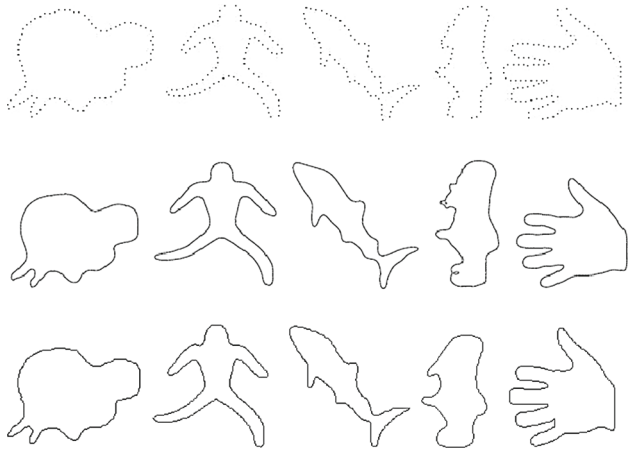

Fig. 9 Comparison FIP representations with IBS representations. Orignial 2D datasets (*first row*). Resulting zero sets of IBS (*second row*) and FIP (*third row*)

Table 3 Comparison of performance for FIPs and IBSs

	IBS		FIP	
Shape name	Size	AFE	Size	AFE
Oni	190	0.2788	124	0.0777
Dude	305	0.6183	124	0.0687
Fish	178	0.7475	124	0.0849
Homer	303	0.2546	124	0.3001
Hand	865	0.4000	124	0.1162

For example, among the objects in Fig. 5, the shape of rat is the most complex, but only eighth-degree FIP shown in Fig. 6 can accurately represent it (the AFE of this fitted FIP is 2.9503). Although there exist extraneous components in the zero set of the FIP, these extraneous components do not intersect the shape of rat. Moreover, we can remove it by further improve fitting algorithm. On the other hand, the fitting results of FIPs shown in Fig. 6 and IPs shown in Fig. 8 are in good accordance with the stability analysis from Theorem 1. That is, these fitting results provide an experimental validation of the theorem.

Implicit B-splines (IBSs) [18] can also accurately and flexibly describe objects with complex topologies. However, in the case of the same accuracy of representation, IBSs require to use much more parameters (e.g., coefficients) than FIPs. Figure 9 illustrates five datasets and we fitted IBSs and FIPs to them, respectively. The resulting zero sets of IBSs and FIPs are shown in the second row and third row. The corresponding coefficient sizes and AFEs are provided in Table 3. From Fig. 9, we can see that the resulting zero sets of FIPs and IBSs are approximately equivalent, but the IBSs need much more parameters than FIPs (see Table 3). For example, for hands, IBSs methods (AFE = 0.4) and FIPs methods (AFE = 0.1162) have approximately the same fitting accuracy. However, IBSs need 865 coefficients and FIPs only need 124 coefficients. Clearly, the use of the fewer coefficients is the key problem in computer vision application.

It is noted that in this work, we only focus on the properties of FIP representation rather than the fitting algorithm. Therefore, for easy comparison, we only use the same RR fitting algorithm to perform all tests. In fact, the fitted FIP shown in Fig. 6 can be further improved by choosing the better fitting algorithm.

In order to illustrate empirically computational cost of FIPs and IPs, we performed an experiment to compare the CPU costs of fitting FIPs and IPs to the fifteen shapes. The experiment was run on an Intel quad CPU 2.33 GHz processor, with 4GB bytes of main memory. The CPU cost of fitting FIPs and fitting IPs are shown in Fig. 10 and the resulting zero sets of FIPs and IPs are shown in Figs. 6 and 8, respectively. From Fig. 10, we can see the CPU cost of the FIP fitting is consistently lower than that of IP fitting, which is highly consistent with the conclusion drawn in Sect. 4.4 that the FIP fitting needs much lower computational cost than the IP in the case of approximately the same representation power.

5.2 Comparison of Contribution of the Base and the Degree to the Fitted FIP

The shapes of bears used for comparing of the contribution of the base and the degree to the fitted FIPs are shown in

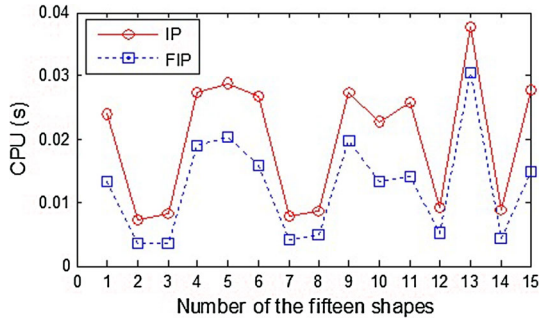

Fig. 10 The comparison of CPU costs of fitting FIPs and IPs

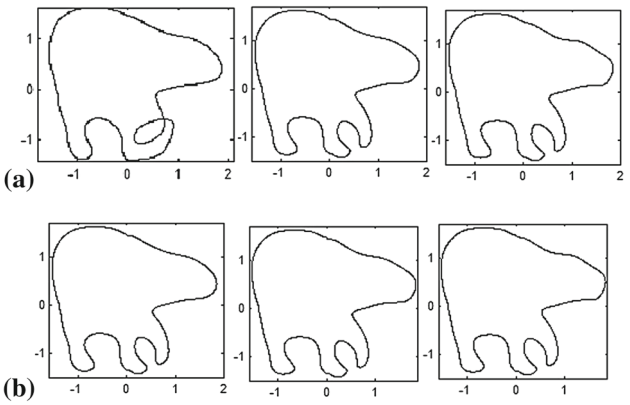

Fig. 11 Comparison of contribution of the base and the degree to FIPs. **a** Fitting result of FIPs of degree 4, 5, 6 with the same base 3, respectively; **b** Fitting result of FIPs of degree 4, 5, 6 with the same base 5, respectively

Fig. 11. Figure 11a shows the results of fitting the fourth-, fifth-, and sixth-degree FIPs of the same bases with its value 3 to bears, respectively, and the AFEs of the three fitted FIPs are shown with solid lines in Fig. 12, while Fig. 11b shows corresponding fitting results in the case of the same bases with their value 5 and their AFEs are shown with dashed line in Fig. 12. It is clear from the two figures that the FIP of the degree 4 and base 3 cannot represent the shapes of bears, but the FIP of degree 4 and base 5 can do this. In addition, the resulting zero set of the FIP of the degree 6 and base 3 is similar to that of the FIP of the degree 4 and base 5. Hence, the fitting result can be improved by increasing either the degree or the base of the FIP. However, under the same condition, we prefer the latter. The reason is that the greater the bases are, the more stable the fitted FIPs become as noted in Sect. 4.1. We give an example to further illustrate this in the following experiments.

By adding uniformly distributed random noise to the coefficients of the FIP to simulate the small coefficient changes, we can find the effect of the different bases on the stability of FIPs. The bear shown in Fig. 5 was used to test in this experiment. The random noise, having a uniform distribution in the range $[-0.01,\ 0.01]$, was generated. The same noise vector

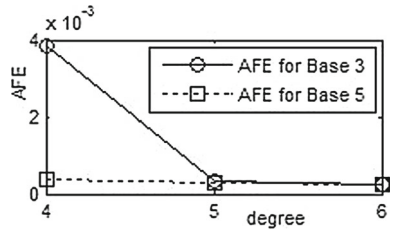

Fig. 12 AFE of the fitted FIP of base 3 and 5, respectively

Table 4 Comparison of error statistic for FIPs of difference base and IPs of the corresponding degree

The type of curves	Mean	Variance
FIP of Base 3	0.2211	0.7480
FIP of Base 5	0.0155	0.0011
IP of degree 18	0.0476	0.0416

The results shown are a uniform random noise added to the coefficients of the two fitted FIPs and one fitted IP. The statistics is based on 200 independent error vectors

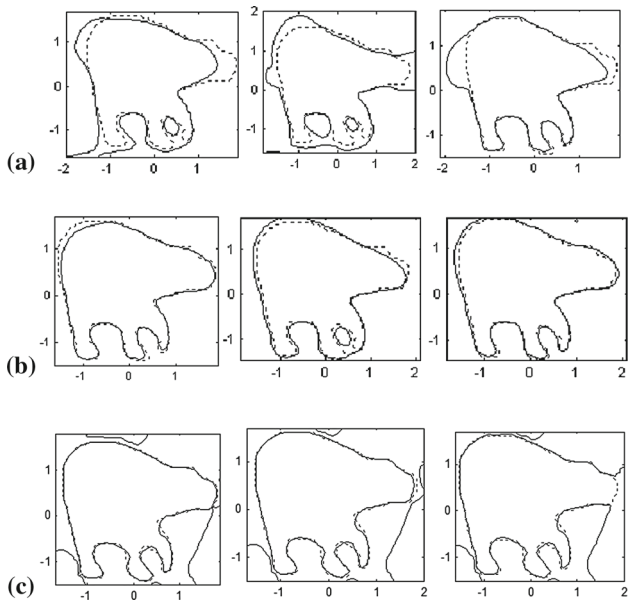

Fig. 13 Sensitivity for FIPs of different bases. **a** the fitted FIP (*solid line*) of degree 6 and base 3 with three noise vectors added to its coefficients, respectively, **b** the fitted FIP (*solid line*) of degree 4 and base 5 with the three noise vectors added to its coefficients, respectively, **c** the fitted IP (*solid line*) of degree 18 with the three noise vectors added to its coefficients, respectively

was added to the coefficients of the fitted FIP and the fitted IP for the bear. Figure 13a shows that the results obtained by adding three independent noise vectors to the same fitted FIP of degree 6 and base 3 for the bear; (b) shows the corresponding results from the FIP of degree 4 and base 5; and (c) shows the corresponding results from the fitted IP of the degree 18. It is clear from the Fig. 13a that the FIP of base 3 appears to be very instable. This means that the smaller changes in the

coefficient of the fitted FIP of base 3 (solid line in Fig. 13a) lead to the failure in representing the bear (dotted line). We also find from Fig. 13c that the three IP curves are instable, especially for the third fitted IP curve (solid line), which was changed into an open curve. However, after the same noise vectors were added to the coefficients of the fitted FIP of base 5, we can find from Fig. 13b that the fitted FIPs were less affected by the added noise vectors. These implied that the FIP of the higher base were more stable. In order to further illustrate this, by adding 200 independent random noise vectors to the coefficients of the fitted FIP of base 3, base 5, and the fitted IP of degree 18 for the bear, respectively, we obtained 200 AFEs for the FIPs of base 3 and base 5 and 200 AFEs for the IP of degree 18, respectively, whose statistics including mean and variance are shown in Table 4. Obviously, the mean and variance for FIP of base 5 were 0.0155 and 0.0011, respectively, which were far smaller than 0.2211, 0.7480 for the FIP of base 3 and 0.0476, 0.0416 for the IP of degree 18. This also indicates that the higher base FIP is more stable than the lower base one in representing complex objects.

6 Conclusion

In this paper, we present the concept of the FIP and introduce it to represent complex objects. Based on analysis of properties of IPs, we give the general formula for FIPs and extend the connectivity and boundedness of IPs to those of FIPs. Furthermore, we investigate the properties of the base of FIPs and find that FIPs have higher stability and powerful representation than IPs due to the presence of the base. In addition, we develop an algorithm for determination of a moderate degree for an FIP to represent a given object, which can be obtained by only computing the number of stationary points on the object.

In simulations, we compare FIPs with IPs in fitting 15 different object shapes. We demonstrate that the advantage of the FIP representation over IP representation is more significant when the shapes are much more complex. In this case, IPs usually fail to represent the complex shapes, but FIPs can accurately represent them with lower computational cost and higher stability. Furthermore, the higher base FIP is more stable than the lower base one.

It is noted that FIPs, extended from IPs, contain many good characteristics of IPs, such as fast fitting, interpretable coefficients, and robustness against noise and occlusion. Particularly, the geometry variants of FIPs can be obtained easily by evaluating the local minimum values of the FIP function because these local minimum values and their ratios are invariant under the similarity transform and affine transform, respectively.

Future work should study the algebraic or geometry variants based on FIPs and extend FIP curves to FIP surfaces.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 71372188 and National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035.

References

1. Abjyankar, S.: Algebraic Geometry for Science and Engineers. American Mathematical Society, Providence (1990)
2. Ben, H.: 3d objects description and classification by implicit polynomials. M.Sc. thesis, The Technion Israel Institute of Technology, Haifa (2008)
3. Ben, H., Malah, D., Barzohar, M.: Recognition of 3d objects based on implicit polynomials. IEEE Trans. Pattern Anal. Mach. Intell. **22**(7), 663–674 (2010)
4. Binder, H., Sauerbrei, W., Royston, P.: Comparison between splines and fractional polynomials for multivariable model building with continuous covariates: a simulation study with continuous response. Stat. Med. **32**(13), 2262–2277 (2013)
5. Blane, M., Lei, Z.: The 3L algorithm for fitting implicit polynomial curves and surfaces to data. IEEE Trans. Pattern Anal. Mach. Intell. **22**(3), 298–313 (2000)
6. Carr, J., Beatson, R., Cherrie, J., Mitchell, T., Fright, W., McCallum, B.: Reconstruction and Representation of 3d Objects with Radial Basis Functions. In: SIGGRAPH, pp. 67–76 (2001)
7. Heizer, A., Barzohar, M., Malah, D.: Stable fitting of 2d curves and 3d surfaces by implicit polynomials. IEEE Trans. Pattern Anal. Mach. Intell. **26**(10), 1283–1294 (2004)
8. Helzer, A., Zohar, M.B., Malah, D.: Using Implicit Polynomials for Image Compression. In: Proceedings of IEEE Conference on Convention of the Electrical and Electronic Engineers, pp. 384–388 (2000)
9. Jaroslav, K., Flusser, J.: Implicit Invariants and Object Recognition. In: Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, pp. 462–469 (2007)
10. Kazhdan, M., Hoppe, H.: Screened Poisson surface reconstruction. ACM Trans. Graph. (TOG) **32**(3), 29 (2013)
11. Keren, D., Cooper, D.: Describing complicated objects by implicit polynomial. IEEE Trans. Pattern Anal. Mach. Intell. **16**(1), 38–53 (1994)
12. Keren, D., Gotsman, C.: Fitting curves and surfaces with constrained implicit polynomials. IEEE Trans. Pattern Anal. Mach. Intell. **21**(1), 31–41 (1999)
13. Landa, Z.: 2d object description and classification based on contour matching by implicit polynomials. M.Sc. thesis, The Technion Israel Institute of Technology, Haifa (2006)
14. Lebmeir, P., Gebert, J.: Rotations translations and symmetry detection for complexified curves. Comput. Aided Geom. Des. **25**(9), 707–719 (2008)
15. Marola, G.: A technique for finding the symmetry axes of implicit polynomial curves under perspective projection. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3), 465–470 (2005)
16. Oden, C., Ercil, A., Yildiz, V., Kirmiztia, H., Buke, B.: Hand Recognition Using Implicit Polynomials and Geometric Features. Springer Lecture Notes in Computer Science, vol. 2091, pp. 336–341. Springer, Berlin (2001)
17. Rouhani, M., Sappa, A.: Implicit polynomial representation through a fast fitting error estimation. IEEE Trans. Image Process. **21**(4), 2089–2098 (2012)

18. Rouhani, M., Sappa, A., Boyer, E.: Implicit b-spline surface reconstruction. IEEE Trans. Image Process. **24**(1), 22–32 (2015)
19. Royston, P., Sauerbrei, W.: Improving the robustness of fractional polynomial models by preliminary covariate transformation: a pragmatic approach. Comput. Stat. Data Anal. **51**(9), 4240–4253 (2007)
20. Subrahmonia, J., Cooper, D., Keren, D.: Practical reliable Bayesian recognition of 2d and 3d objects using implicit polynomials and algebraic invariants. IEEE Trans. Pattern Anal. Mach. Intell. **18**(5), 505–519 (1996)
21. Tarel, J., Cooper, D.: The complex representation of algebraic curves and its simple exploitation for pose estimation and invariant recognition. IEEE Trans. Pattern Anal. Mach. Intell. **22**(7), 663–674 (2000)
22. Tasdizen, T., Cooper, D.: Boundary estimation from intensity color images with algebraic curve models. In: Proceedings of 15th International Conference on Pattern Recognition, pp. 225–228 (2000)
23. Tasdizen, T., Tarel, J.: Improving the stability of algebraic curves for application. IEEE Trans. Image Process. **9**(3), 405–416 (2000)
24. Taubin, G., Cukirman, F., Sullivan, S.: Parameterized families of polynomials for bounded algebraic curve and surface fitting. IEEE Trans. Pattern Anal. Mach. Intell. **16**(3), 286–303 (1994)
25. Wu, G., Li, D.: Object recognition based on affined invariants in implicit polynomial curves. Acta Electron. Sin. **32**(12), 1987–1991 (2004)
26. Wu, G., Yang, J.: A representation of time series based on implicit polynomial curve. Pattern Recogn. Lett. **34**(4), 361–371 (2013)
27. Yasin, A., Calli, B., Unel, M.: Image based visual servoing using algebraic curves applied to shape alignment. In: Proceedings of IEEE International Conference On Intelligent Robots and Systems, pp. 5444–5449 (2009)
28. Zheng, B., Ishikawa, R., Oishi, T., Takamatsu, J., Ikeuchi, K.: A fast registration method using IP and its application to ultrasound image registration. IPSJ Trans. Comput. Vis. Appl. **1**, 209–219 (2009)
29. Zheng, B., Takamatsu, J., lkeuchi, K.: An adaptive and stable method for fitting implicit polynomial curves and surfaces. IEEE Trans. Pattern Anal. Mach. Intell. **32**(3), 561–567 (2010)

Gang Wu received his BS degree in applied mathematics in 1992 from Chengdu University of Technology in China. He received his MS and PhD degrees in computer science from Hefei University of Technology in 1998 and 2001, respectively. His research interests include image processing, computer vision, and pattern recognition. He has published more than 30 research papers.

Yanchun Zhang received his PhD degree in computer science from the University of Queensland in 1991. He is a full professor of computer science and the director of the Centre for Applied Informatics, Victoria University, Melbourne. His research interests include databases, cooperative transactions management, Web information systems, Web mining, and Web services. He has published more than 150 research papers in international journals, such as the ACM Transactions on Computer and Human Interaction (TOCHI), the IEEE Transactions on Knowledge and Data Engineering (TKDE), and IEEE Internet Computing, and in conference proceedings. He authored/edited more than a dozen books/proceedings and journal special issues in the related areas. He is an editor in chief of the World Wide Web: Internet and Web Information Systems (WWW Journal).