Pattern Recognition Letters 62 (2015) 41–48

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A new Chebyshev polynomials descriptor applicable to open curves[☆]

Gang Wu[a,*], Yanchun Zhang[b]

[a] Institute of Multimedia Computing Technology, Nanjing University of Finance and Economics, Nanjing 210003, China
[b] Centre for Applied Informatics, Victoria University, Melbourne 8001, Australia

ARTICLE INFO

Article history:
Received 14 January 2015
Available online 27 May 2015

Keywords:
Chebyshev polynomial
Fourier descriptor
Shape analysis
Open curves
Shape representation

ABSTRACT

Effective and efficient representation of open curves is a challenging problem in statistical shape analysis. In this paper, we propose a novel shape descriptor, called Chebyshev polynomial descriptor (CPD) for representing open curves. Firstly, a general formula for the computation of CPDs and parametric equations of reconstructed curves are given; secondly, we investigate properties of CPDs, including its stability, similarity-invariant and invariance under different starting points. Finally, the reconstructed curve from CPDs is used to compute the curvature of an original open curve. Experimental results demonstrate the effectiveness of both representation of handwriting and computation of curvatures.

1. Introduction

The edge information extracted from real-world image is a plane contour shape, which is usually represented as a closed curve or an open curve. During the past decades, many methods were proposed to represent closed curves, such as Fourier descriptors [15], curvature scale space [7], implicit polynomial curves [11] and so on. However, the representation of open curves has received much less attention than that of closed curves. Actually, many shapes extracted from images are open, including the strokes of digital signatures [8], the shapes of plant organs containing worm-eaten defects [12], the edges of hand postures and the edges of face profiles. In addition, we sometimes need to focus on a certain part of an object outline and the part may be also regarded as an open curve, for example curve matching for open curves [3] and so on. Consequently, the use of efficient methods to represent open curves is important task when analyzing objects in images. Although most of methods for representing closed curves can be applied to open curves, these methods do not always work well because they usually suffer from a failure to process the points near endpoints of open curves. For instance, in elliptic Fourier descriptor [4], the reconstructed curve tends to become a closed curve and to vibrate near the endpoints. In order to efficiently represent open curves, the P-type Fourier descriptor [13] was proposed and has been successfully applied to recognition of shape of human face profiles [1], statistical shape analysis of rice leaf [17] and tracing open curves [14]. However, this method is sensitive to noise and hence fails to analyze the subtle and fine structure of a shape. In contrast to P-type Fourier descriptor, the protrusion Fourier descriptor [12] is more robust against noise, but this method suffers from high computational cost and difficulty to reconstruct the original shape from its descriptors. The above methods belong to the type of statistical analysis. Another type of methods are of approximate representations. Implicit polynomial representations and B-spline representations fall into this category. Implicit polynomial representations are unstable and only suitable for simple shapes [16]. B-spline representations need to determine the control points before representing shapes. These disadvantages limit their application in shape representation. In addition, based on Chebyshev polynomials theory, the Chebyshev moment descriptors were proposed in [9] to represent shapes of objects. However, the descriptors belong to a kind of region-based descriptors. Generally, they are not appropriate for describing curves, especially open curves. Chebyshev moment descriptors can be applied to a closed curve by changing the curve into a binary image, for example setting to 1 the pixels located inside the closed curve and to 0 the pixels located outside. Unlike close curves, there are no interior and exterior for open curves, and hence, we cannot use region-based shape descriptors, such as Chebyshev moment descriptors, to represent open curves.

As a matter of fact, to represent open curves, a key problem is how to handle their extremities, but most of existing algorithms cannot work well at those points nearby endpoints [18]. In general, an original open curve needs to be padded with some data, such as padding the curve with zero data, replicated data, symmetric data and so on. In addition, in [2], the open curves are padded through minimizing the spectral energy. However, these methods need much more computational cost and suffer from lack of theoretical basis.

☆ This paper has been recommended for acceptance by A. Koleshnikov.

* Corresponding author. Tel.: +86 13770590389.
E-mail address: wugang69@gmail.com (G. Wu).

http://dx.doi.org/10.1016/j.patrec.2015.05.004
0167-8655/

In this work, we propose a novel Chebyshev polynomial descriptor (CPD) to represent open curves. In contrast to the existing methods, the CPDs have the following advantages.

1. Powerful ability to represent open curves. Like Fourier descriptor, the first few low order CPDs can capture the overall shape while the high order ones capture its finer details. Besides, CPDs are easy to normalize (rotation, scale and translation) and preserve the overall shape information. The CPDs represent not only the closed curves but also open curves, especially the self-overlapping curves.
2. Accurate, stable and fast representing. The use of Chebyshev polynomial to represent digital open curves is actually Chebyshev interpolation at Chebyshev polynomial zeros. This means the reconstructed curve can completely pass through all points on the digital open curves. Hence, CPDs can accurately and stably represent open curves. On the other hand, due to the orthogonality of Chebyshev polynomials, the mathematical formulation for the computation of CPDs is very simple and accordingly, CPDs can be obtained with low computational cost.
3. Easy to be applied. The formula for CPDs is a real valued function and the equations of the reconstructed curve are expressed by a finite sum of Chebyshev polynomials. Then, many theorems and methods for Chebyshev polynomial can be used to analyze the represented open curves. For instance, we can compute the derivatives and curvatures of an original open curve through analyzing its reconstructed curve from CPDs.

This paper is organized as follows. We discuss the background in the next section. Specifically, we review the mathematical formulation for Chebyshev polynomials, and focus on how to use Chebyshev polynomial to represent one dimensional data sets. In Section 3, we define the Chebyshev polynomial descriptor and investigate how to reconstruct the original curve from its CPDs. Furthermore, we explore the properties of the CPDs, including the stability of CPDs and the similarity-invariant based on CPDs. In addition, CPDs independent on starting points are also discussed. Section 4 gives an algorithm for computation of the derivative and curvature of shapes based on CPDs. Section 5 presents simulation results obtained in representing handwriting and recognition of shapes. In addition, the estimation of curvatures of curves is also performed. Section 6 summarizes and concludes the paper.

2. Background: representation of one dimensional data sets with Chebyshev polynomials

In this section, we provide a brief overview of Chebyshev polynomials and their key properties, including definition and orthogonality of Chebyshev polynomials, especially focusing on how to fit a chebyshev polynomial to one given dimensional data set.

2.1. Mathematical formulation for Chebyshev polynomials

There are several kinds of Chebyshev polynomials [6]. However, in this paper, we only discuss Chebyshev polynomials of the first kind. Actually, Chebyshev polynomial of the first kind is one of the most important Chebyshev polynomials. In the following sections, we refer to the Chebyshev polynomial of the first kind as Chebyshev polynomial.

A Chebyshev polynomial $P_m(t)$ is a polynomial in t of degree m, defined as

$$P_m(t) = \cos(m\ \arccos(t)) \tag{1}$$

Clearly, the range of variable t is the interval $[-1, 1]$. That is, Chebyshev polynomials are polynomials defined on the interval $[-1, 1]$. Let $t = \cos(\theta)$, and then we can obtain $P_m(t) = \cos(m\theta)$. According to trigonometric identity $\cos m\theta + \cos(m-2)\theta = 2\cos\theta\cos(m-1)\theta$, the definition of Chebyshev polynomial (1) can be rewritten with recurrence relation as follows

$$\begin{aligned} &P_0(t) = 1,\ P_1(t) = t \\ &P_m(t) = 2tP_{m-1}(t) - P_{m-2}(t),\quad m = 2, 3, \ldots \end{aligned} \tag{2}$$

Chebyshev polynomials exhibit many good properties. For example, Chebyshev polynomials can overcome Runge's phenomenon. If we use polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points, then a problem of oscillation at the edges of an interval will appear. This is known as Runge's phenomenon. However, the roots of the Chebyshev polynomials are used as nodes in polynomial interpolation and the resulting interpolation polynomial minimizes the problem of Runge's phenomenon. Furthermore, the interpolation polynomial is of best approximation to a continuous function under the maximum norm. These good properties motivate us to use Chebyshev polynomials to represent open curves.

2.2. Chebyshev polynomials approximation

The orthogonality is a key property of Chebyshev polynomials. Specifically, the Chebyshev polynomials $P_0(t), P_1(t), \ldots, P_m(t)$ form an orthogonal system on the interval $[-1, 1]$ with respect to the weight $(1-t^2)^{-\frac{1}{2}}$. That is, the system of Chebyshev polynomials satisfies the following identity [6]

$$\int_{-1}^{1} \frac{P_i(t)P_j(t)}{\sqrt{1-t^2}} dt = \begin{cases} 0\,, & i \neq j \\ \dfrac{\pi}{2}\,, & i = j \neq 0 \\ \pi\,, & i = j = 0 \end{cases} \tag{3}$$

where $(1-t^2)^{-\frac{1}{2}}$ is a function of t on the interval $[-1, 1]$, called the Chebyshev weight function and denoted by $w(t)$.

Due to the orthogonality of the Chebyshev polynomials, a given function $f(t)$ can be expanded in a series based on the system of Chebyshev polynomials. That is

$$f(t) = \sum_{i=0}^{\infty} c_i P_i(t) \tag{4}$$

It is clear from the above series expansion that the function $f(t)$ can be approximated by the first m items in the series. That is, the given function $f(t)$ may be approximated as

$$f(t) \approx \sum_{i=0}^{m} c_i P_i(t) \tag{5}$$

Particularly, if the function $f(t)$ is a polynomial of degree less than or equal to m, then the first m items in the Chebyshev series expansion in (4) will exactly approximate the function $f(t)$. Clearly, Chebyshev polynomials approximation of the function $f(t)$ depends on how to compute the coefficients c_i in (5). According to [6], the coefficients c_i are given by the explicit formula

$$c_i = \begin{cases} \dfrac{1}{m}\displaystyle\sum_{j=1}^{m} f(t_j)P_i(t_j) = \dfrac{1}{m}\displaystyle\sum_{j=1}^{m} f(t_j), & i = 0 \\ \dfrac{2}{m}\displaystyle\sum_{j=1}^{m} f(t_j)P_i(t_j), & 1 \le i \le m \end{cases} \tag{6}$$

where $t_j = \cos\frac{(j-0.5)\pi}{m}$. Obviously, the values of $P_m(t_j), j = 1, 2, \ldots, m$ are zeros. Hence we call $t_j, j = 1, 2, \ldots, m$ the Chebyshev ploynomial zeros [6]. It follows from above formulae that the Chebyshev polynomials $P_i(t), i = 1, 2, \ldots, m$ are orthogonal over the Chebyshev polynomial zeros.

For one dimensional data sets, e.g., $y_j, j = 1, 2, \ldots, m,$ we may construct a continuous function $f(t)$ so that $f(t_j) = y_j, (j = 1, 2, \ldots, m)$. In this case, the Chebyshev polynomials can be used

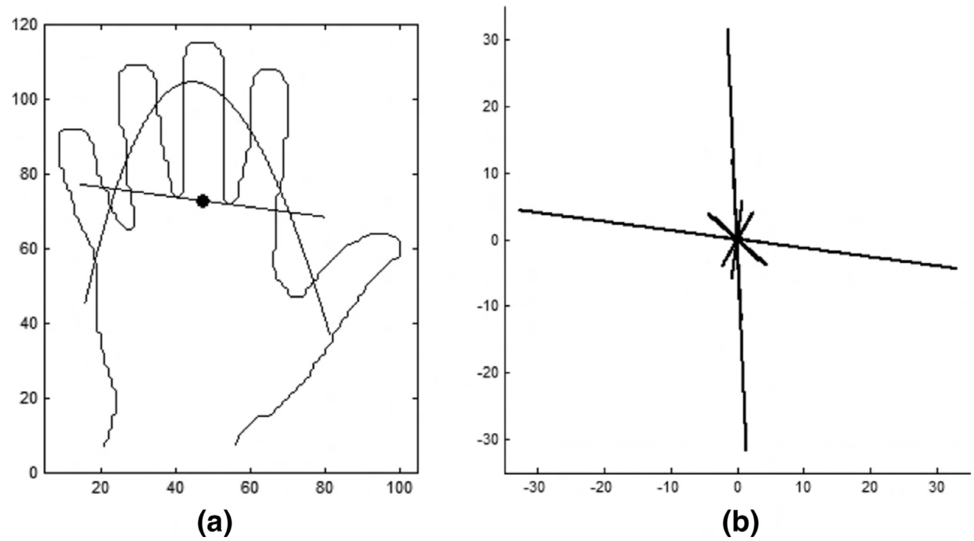


Fig. 1. An example of the locus of 10 lines. (a) Contour of the hand. (b) The lines corresponding to the first 10 items of CPDs for hands.

to approximate the function $f(t)$ according to (5), where the coefficients are computed by (6). Actually, this approximation method is an interpolation one. That is, we can use the Chebyshev polynomials $P(t) = c_0P_0(t) + c_1P_1(1) + \cdots + c_mP_m(t)$ to perform interpolation at the Chebyshev polynomial zeros t_j, $(j = 1, 2, \ldots, m)$. As noted in [6], the Chebyshev approximation is a minimax approximation and far more efficient and stable than other approximation methods. Indeed, Chebyshev approximation has been successfully applied to representation of time series [10].

3. Chebyshev polynomial descriptors of open curves

In this section, we propose a new Chebyshev polynomial descriptor (CPD) to represent a given open curve based on the Chebyshev polynomial approximation discussed in Section 2.2. Then we further discuss how to reconstruct a represented curve from its CPDs.

3.1. The definition of CPDs

An open plane curve C can be expressed by parametric equation $Z(t) = (x(t), y(t)), t \in [0, L]$, where $x(t)$ and $y(t)$ are the functions of the length t of the curve from an end point on C, and L denotes the total length of the open curve C. Clearly, each value of t determines a point ($x(t)$, $y(t)$). Then the point ($x(t)$, $y(t)$) varies and traces out a curve C in a coordinate plane as t varies. In addition, the parameter t does not necessarily mean the length of the curve from end point on C, but rather the parameter of time.

In the discrete case, A digitized curve having N points can be expressed by a data set $C = \{Z_i | (x_i, y_i), i = 1, 2, \ldots, N\}$. Then, both $\{x_i\}$ and $\{y_i\}$ may be regarded as one dimensional data set respectively. Hence, we can respectively apply Chebyshev approximation to these two data sets and obtain their coefficients of Chebyshev approximation according to (6). Specifically, let $\{a_i\}$ denote the coefficients of Chebyshev approximation applied to the data set $\{x_i\}$, and $\{b_i\}$ denote the coefficients corresponding to $\{y_i\}$. Then ,we have

$$\begin{cases} a_0 = \dfrac{1}{N}\sum_{j=1}^{N} x_j, \ a_i = \dfrac{2}{N}\sum_{j=1}^{N} x_j P_i(t_j), \ i = 1, 2, \ldots, N \\ b_0 = \dfrac{1}{N}\sum_{j=1}^{N} y_j, \ b_i = \dfrac{2}{N}\sum_{j=1}^{N} y_j P_i(t_j), \ i = 1, 2, \ldots, N \end{cases} \quad (7)$$

The coefficients of Chebyshev approximation, a_i and b_i, $i = 0, 1, 2, \ldots, N$, are named as Chebyshev polynomial descriptors (CPDs). In addition, coefficients a_i and b_i are usually noted $CPD_i, i = 0, 1, 2, \ldots, N$. The value N represents the order of CPDs. Like Fourier descriptors, the CPDs represent the curve in spectral domain and control the amount of each frequency that contributes to make up the curve. As a result, CPDs do not have the same values for different curves, which enable CPDs to describe the curves.

3.2. The reconstruction of curves from CPDs

After obtaining the CPDs (a_i, b_i, $i = 0, 1, \ldots, N$) of a certain curve, according to (5), we can easily reconstruct the curve and then formulate it in the form of parametric equations:

$$\begin{cases} x_l(t) = \sum_{i=0}^{l} a_i P_i(t) \\ y_l(t) = \sum_{i=0}^{l} b_i P_i(t) \end{cases} \quad (8)$$

where $0 \le l \le N$

Each term in the above equation has a geometric interpretation as a line. That is, for a fixed value of i, the corresponding parametric equation $x = a_iP_i(t), \ y = b_iP_i(t)$ defines the locus of a line segment in the coordinate plane. Clearly, from the parametric equation, we have $y/x = b_i/a_i$. This means the equation of the line with value i is $y = (b_i/a_i)x$. On the other hand, from (1), we have $-1 \le P_i(t) \le 1$. Hence, the length of the line segment defined by $x = a_iP_i(t), y = b_iP_i(t)$ is $\sqrt{4a_i^2 + 4b_i^2}$

Fig. 1 (b) illustrates the 10 line segments corresponding to the first 10 items of CPDs obtained by representing a hand shape shown in Fig. 1(a). In this interpretation, the values of a_0 and b_0 define the location of the hand, which is shown by the dot in Fig. 1(a) and the two longest line segments in Fig. 1(b) are given by the two parametric equations $x = a_1P_1(t), \ y = b_1P_1(t)$ and $x = a_2P_2(t), \ y = b_2P_2(t)$ respectively. Also, we can see that the line segment becomes shorter as the index of each item i in (8) increases. This implies the first several items of CPDs contain a large amount of information on the features of a represented curve. Figs. 1(a) and 2 illustrate this idea. When $l = 1$ in (8), the first order reconstructed curve is a line. However when the second item is considered, namely, $l = 2$, the line changes into a conic (the line and the conic are shown in Fig. 1(a)). When adding more items, the reconstructed curve intends to represent an accurate approximation of the original shape of the hand. As shown in Fig. 2, when $k = 10$, the 10th order reconstructed curve describes

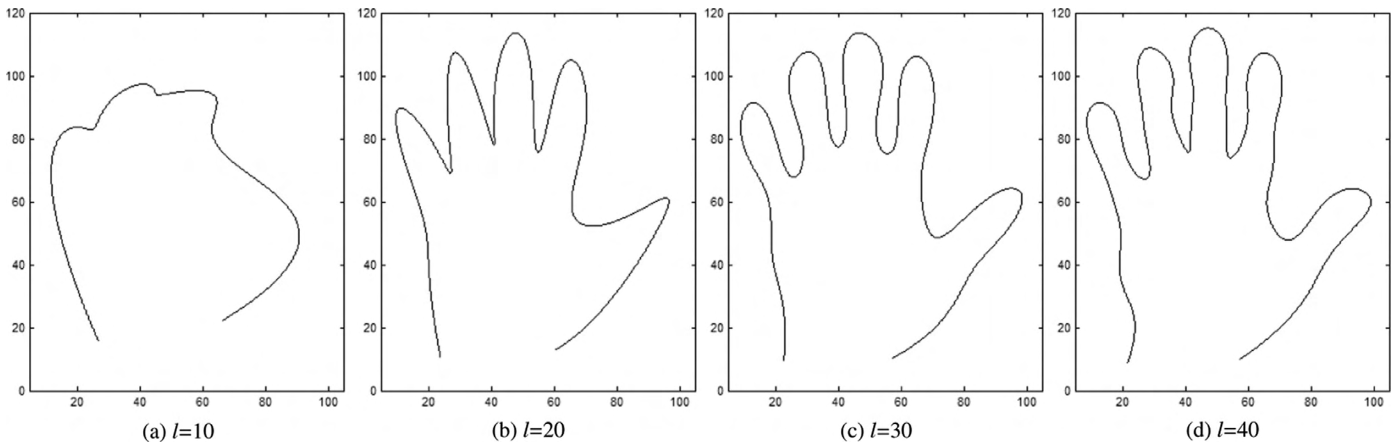


Fig. 2. Reconstruction of the hand shape from the first N items of its CPDs.

the main features of the hand, but when $l = 30$, the 30*th* order reconstructed curve accurately represents the shape of the hand. In general, like fourier descriptor, the CPDs of lower index contain information about general features of curves, and ones of high index contain finer details. Furthermore, The CPDs can describe a curve by only a few values. It is noted that each Fourier descriptor (e.g. Ellipse Fourier Descriptor) contains four parameters, but each CPD contains two parameters. This is because Fourier descriptors are complex numbers each of which has two parameters.

In addition, we define a representation error of CPDs to evaluate the accuracy of representation according to (8) as follows

$$RE = \sqrt{\sum_{i=0}^{N}[(x_i - x_l(t_i))^2 + (y_i - y_l(t_i))^2]} \tag{9}$$

3.3. The stability of reconstructed curves

In this section, we discuss the stability of representing curves using CPDs. The stability means as the number of CPDs increases, the representation errors of reconstructed curves from these CPDs decrease. As shown in Fig. 2, the reconstructed curve will approximate the original curve more and more accurately as its order increases without suffering from vicious oscillation.

Theorem 1. *Supposed that a given digitized curve consisting of N points,* $(x_i, y_i), i = 1, 2, \ldots, N$*, is represented by Chebyshev polynomial descriptors* $a_j, b_j, j = 0, 1, 2, \ldots, N$*, then the Nth order reconstructed curve passes through all these points* (x_i, y_i)*.*

Proof. Let $f(t_j) = x_j, g(t_j) = y_j$, where $t_j = \cos\frac{(j-0.5)\pi}{N}$ and then, we can obtain the reconstructed curve according to (8) and (7). Centering the curve center of mass at the origin of the coordinate system and substituting a_i and b_i in (7) into (8), we have the equations of the reconstructed curve as follows

$$\begin{cases} x_N(t) = \sum_{i=1}^{N} a_i P_i(t) = \sum_{i=1}^{N} \frac{2}{N} \sum_{j=1}^{N} f(t_j) P_i(t_j) P_i(t) \\ y_N(t) = \sum_{i=1}^{N} b_i P_i(t) = \sum_{i=1}^{N} \frac{2}{N} \sum_{j=1}^{N} g(t_j) P_i(t_j) P_i(t) \end{cases}$$

It should noted that $a_0 = \frac{2}{N}\sum_{j=1}^{N} f(t_j)P_0(t_j) = 0$ and $b_0 = \frac{2}{N}\sum_{j=1}^{N} g(t_j)P_0(t_j) = 0$. For an arbitrary value $t_k = \cos((k-0.5)\pi)/N$, substituting t_k into the above equations and rearranging the order of summation, we have

$$\begin{cases} x_N(t_k) = \frac{2}{N} \sum_{j=1}^{N} f(t_j)\{\sum_{i=1}^{N} P_i(t_j)P_i(t_k)\} \\ y_N(t_k) = \frac{2}{N} \sum_{j=1}^{N} g(t_j)\{\sum_{i=1}^{N} P_i(t_j)P_i(t_k)\} \end{cases} \tag{10}$$

Denote the sum $\sum_{i=1}^{N} P_i(t_j)P_i(t_k)$ by $S(j, k)$ and let $t_j = \cos(\theta_j)$ and $t_k = \cos(\theta_k)$. Then, we have

$$\begin{aligned} S(j, k) &= \sum_{i=1}^{N} P_i(t_j)P_i(t_k) = \sum_{i=1}^{N} \cos(i\theta_j)\cos(i\theta_k) \\ &= \frac{1}{2}\sum_{i=1}^{N}[\cos(i(\theta_j + \theta_k)) + \cos(i(\theta_j - \theta_k))] \end{aligned} \tag{11}$$

Considering $\theta_j = (j-0.5)\pi/N$, $\theta_k = (k-0.5)\pi/N$, $\sum_{i=1}^{N} \cos(i\theta) = (\sin(N\theta/2)\cos(N+1)\theta/2)/\sin(\theta/2)$. Then we may deduce from (11) that $S(j, k) = 0$ when $j \neq k$ and $S(j, k) = N/2$ when $j = k$. Furthermore, from (10), we have

$$\begin{cases} x_N(t_k) = f(t_k) = x_k \\ y_N(t_k) = g(t_k) = y_k \end{cases}$$

The above equations mean that the point (x_k, y_k) lies on the reconstructed curve. That is, the *Nth* order reconstructed curve passes through all points $(x_i, y_i), i = 1, 2, \ldots, N$ □

The above theorem shows that the *Nth* order reconstructed curve can accurately represent the curve consisting of N points. The property of *lth* $(l \leq N)$ order reconstructed curve is given by the following theorem.

Theorem 2. *The representation of a digitized curve consisting of N points using lth* $(l \leq N)$ *order reconstructed Chebyshev polynomial curve is stable.*

Proof. Denoting parametric equation of the *lth* order reconstructed curve by $(x_l(t), y_l(t))$, *Nth* order reconstructed curve by $(x_N(t), y_N(t))$, according to (8), we have

$$\begin{cases} |x_l(t) - x_N(t)| = |\sum_{i=l}^{N} a_i P_i(t)| \\ |y_l(t) - y_N(t)| = |\sum_{i=l}^{N} b_i P_i(t)| \end{cases}$$

Considering $|P_i(t)| \le 1$ and the coefficients $|a_i|$ and $|b_i|$ decrease rapidly as the index i increases [6], we have the following inequality

$$\begin{cases} |x_l(t) - x_N(t)| \le \sum_{i=l}^{N} |a_i| \le (N-l)a_l \\ |y_l(t) - y_N(t)| \le \sum_{i=l}^{N} |b_i| \le (N-l)b_l \end{cases}$$

It is clear from the above inequality and Theorem 1 that the curve $(x_l(t), y_l(t))$ approximates the original curve $(x_N(t), y_N(t))$ as l goes to N. That is, the representation of a digitized curve using the lth order reconstructed Chebyshev polynomial curve is stable. □

In addition, because the first few low order CPDs can capture the overall shape while the high order ones capture its finer details, we can choose enough CPDs to preserve the useful information of a curve and filter out the noise introduced by sampling and quantizing. Our experiments show CPDs remain stable when we resample a given curve with different methods.

3.4. The similarity-invariant based on CPDs

Invariant descriptors can be used to recognize an object from its image without detailed prior knowledge of the actual transformation such as similarity or other transformations. Like Fourier descriptor, there exist similarity invariants based on CPDs. We also call them similarity invariant descriptors based on CPDs, which can be used to recognize a specific curve undergoing a similarity transformation. Specifically, the similarity invariants of a certain curve mean that they remain invariant when the curve is translated, scaled or rotated. Accordingly, we have the following theorem.

Theorem 3. *Supposed that a given curve is represented by Chebyshev polynomial descriptors* $a_i, b_i, i = 0, 1, 2, \ldots, N$, *then* $\sqrt{a_i^2 + b_i^2}/\sqrt{a_1^2 + b_1^2}$, $i = 2, 3, \ldots, N$ *are similarity invariant descriptors*

Proof. In order to obtain similarity invariant descriptors based on CPDs, we rewrite (8) in vector form as

$$\begin{bmatrix} x_N(t) \\ y_N(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \sum_{i=1}^{N} \begin{bmatrix} a_i \\ b_i \end{bmatrix} P_i(t) \tag{12}$$

Supposed that the curve(C) given by the above equation is translated to a point (t_x, t_y), scaled by a scale factor s and rotated around origin by an angle α counterclockwise, then the equation of the transformed curve(C') can be given by

$$\begin{bmatrix} x'_N(t) \\ y'_N(t) \end{bmatrix} = \begin{bmatrix} a_0 + t_x \\ b_0 + t_y \end{bmatrix} + s \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \sum_{i=1}^{N} \begin{bmatrix} a_i \\ b_i \end{bmatrix} P_i(t)$$

The above equation can be further written as follows

$$\begin{bmatrix} x'_N(t) \\ y'_N(t) \end{bmatrix} = \begin{bmatrix} a_0 + t_x \\ b_0 + t_y \end{bmatrix} + s \sum_{i=1}^{N} \begin{bmatrix} a_i \cos(\alpha) + b_i \sin(\alpha) \\ -a_i \sin(\alpha) + b_i \cos(\alpha) \end{bmatrix} P_i(t) \tag{13}$$

On the other hand, we may define the transformed curve(C') as

$$\begin{bmatrix} x'_N(t) \\ y'_N(t) \end{bmatrix} = \begin{bmatrix} a'_0 \\ b'_0 \end{bmatrix} + \sum_{i=1}^{N} \begin{bmatrix} a'_i \\ b'_i \end{bmatrix} P_i(t) \tag{14}$$

By comparing (13) and (14), we have

$$a'_0 = a_0 + t_x \quad b'_0 = b_0 + t_y$$

$$a'_i = s(a_i \cos(\alpha) + b_i \sin(\alpha))$$
$$b'_i = s(-a_i \sin(\alpha) + b_i \cos(\alpha))$$

According to the above identities, we have

$$a_i'^2 + b_i'^2 = s^2(a_i^2 + b_i^2)$$

Hence, $\sqrt{a_i^2 + b_i^2}/\sqrt{a_1^2 + b_1^2}$, $i = 2, 3, \ldots, N$ are the similarity invariant descriptors based on CPDs □

From the above formulae, we can obtain $N - 1$ similarity invariant descriptors based on CPDs. Clearly, these descriptors contain neither scale factor s, nor the rotation α. Then they can be used to recognize the objects undergoing similarity transformation.

3.5. CPDs independent on starting points

Generally, there are two end points on an open curve. Either of the two end points may be chosen as a starting point when the CPDs are used to represent the curve. In this section, we discuss the CPDs are independent of the starting points.

In (7), The CPDs a_i and b_i are obtained by choosing the point (x_1, y_1) as the starting point. Considering $t_j = \cos\frac{(j-0.5)\pi}{N}$, $j = 1, 2, \ldots, N$, we have $P_i(t_j) = cos(i \cos\frac{(N-j+1-0.5)\pi}{N}) = P_i(t_{N-j+1})$, namely, $P_i(t_j) = P_i(t_{N+1-j})$, $j = 1, 2, \ldots, N$. If we choose the point (x_N, y_N) as a starting point and denote the corresponding obtained CPDs by a'_i and b'_i, then we have

$$\begin{aligned} a'_0 &= \frac{1}{N}(x_N + x_{N-1} + \cdots + x_1) = \frac{1}{N}\sum_{j=1}^{N} x_j = a_0 \\ a'_i &= \frac{2}{N}(x_N P_i(t_1) + x_{N-1}P_i(t_2) + \cdots + x_1 P_i(t_N)) \\ &= \frac{2}{N}(x_N P_i(t_N) + x_{N-1}P_i(t_{N-1}) + \cdots + x_1 P_i(t_1)) \\ &= \frac{2}{N}\sum_{j=1}^{N} x_j P_i(t_j) \\ &= a_i \end{aligned}$$

In a similar way, we have $b'_i = b_i$ $i = 0, 1, 2, \ldots, N$. Consequently, it follows that the CPDs remain invariant under the different starting points.

4. Derivatives and curvatures of curves based on CPDs

For a digitized open curve, it is difficult to estimate its derivative and curvature at a given point, especially at those points nearby the endpoint. In this section, we focus on how to use the CPDs to estimate the curvatures and derivatives of curves.

According to (1), we can easily obtain the formula for the first derivative of the curve described by (8)

$$\begin{cases} x'(t) = \sum_{i=0}^{N-1} \bar{a}_i P_i(t) - \frac{1}{2}\bar{a}_0 \\ y'(t) = \sum_{i=0}^{N-1} \bar{b}_i P_i(t) - \frac{1}{2}\bar{b}_0 \end{cases} \tag{15}$$

where $\bar{a}_i = \sum_{k=i+1,\ k-i\ odd}^{N} 2ka_k$ and $\bar{b}_i = \sum_{k=i+1,\ k-i\ odd}^{N} 2kb_k$.

Similarly, the formula for the second derivative is as follows

$$\begin{cases} x''(t) = \sum_{i=0}^{N-2} \hat{a}_i P_i(t) - \frac{1}{2}\hat{a}_0 \\ y''(t) = \sum_{i=0}^{N-2} \hat{b}_i P_i(t) - \frac{1}{2}\hat{b}_0 \end{cases} \tag{16}$$

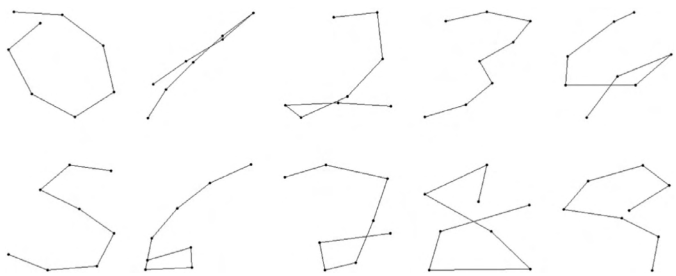


Fig. 3. Ten original handwritten digits.

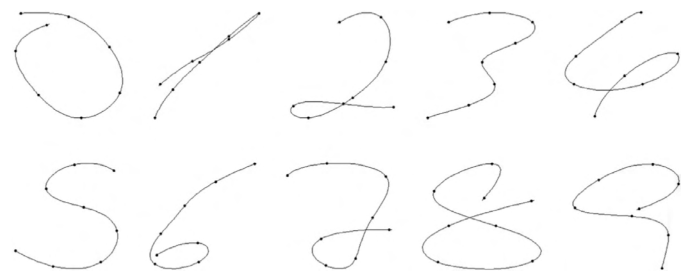


Fig. 4. The reconstructed curves of 10 handwritten digits from their CPDs.

where $\hat{a}_i = \sum_{k=i+2,\ k-i\ even}^{N} (k^2 - i^2) k a_k$ and $\hat{b}_i = \sum_{k=i+2,\ k-i\ even}^{N} (k^2 - i^2) k b_k$.

For every point on curves, we may use (15) and (16) to compute the curvature at the point. For example, for a given point (x_k, y_k) on a certain curve, from Theorem 1, we have $x_k = x(t_k)$ and $y_k = y(t_k)$. Hence, the first and second derivatives of functions $x(t)$ and $y(t)$ at this point are $x'(t_k), y'(t_k), x''(t_k)$ and $y''(t_k)$ respectively. Then the curvature (C_k) of the curve at this point can be obtained by the following formula.

$$C_k = \frac{x'(t_k)y''(t_k) - y'(t_k)x''(t_k)}{(x'(t_k)^2 + y'(t_k)^2)^{(3/2)}} \tag{17}$$

As discussed in Section 3.2, only the first few CPDs in (8) may be used to estimate the curvatures because these CPDs are efficient enough to describe the curve. Consequently, the estimation of curvatures using CPDs has lower computational complexity. On the other hand, the curvatures of the curve can be computed at varying levels of detail from the corresponding varying order reconstructed curve. This means the curvature scale space representation can be obtained by CPDs. From (15) and (16), it follows that the use of CPDs to estimate curvatures of curves at varying levels of detail is efficient because the calculation results at the previous level of detail can be reused. For instance, if the curvatures of *nth* order reconstructed curve have been estimated, then the curvatures of $(n+1)th$ order reconstructed curve can be obtained by only computing the 4 coefficients: $\bar{a}_{n+1}$, $\bar{b}_{n+1}$, $\hat{a}_{n+1}$ and $\hat{b}_{n+1}$. In addition, the curvatures of the open curve at those points nearby the endpoint can be directly computed without padding this curve.

5. Experimental results

In this section, we illustrated the effectiveness of CPDs through using it to represent and recognize handwritten digits, and further presented the comparison of CPDs and FDS in representing five shapes. In addition, due to the exact mathematical expression of CPDs, we performed a test on how to estimate the curvatures of digitized curves using CPD representation.

5.1. Recognition of handwriting and comparison of representation stability

The strokes of handwritten digits are typically open curves. Hence, we conducted the experimental evaluation on handwritten digits from UCI pendigits data set which can be found at https://archive.ics.uci.edu/ml/datasets. There are 10992 handwritten digits of the 10 numeral classes ("0"-"9") in this data set and each of handwritten digits consists of 8 points. In addition, the number of handwritten digits of each numeral class is approximately 1000. Fig. 3 shows 10 different handwritten digits obtained by selecting randomly one from handwritten digits of each numeral class. In this figure, we can see every handwritten digit contains 8 points and we connected them with the straight line segments in order. Clearly, the handwritten digits are coarsely represented due to the large and equal distance between every two adjacent points. We respectively used CPDs to represent the 10 handwritten digits and reconstructed curves of these original handwritten digits from their corresponding CPDs. The resulting curves are showed in Fig. 4. From this figure, we see that the reconstructed curves are able to accurately represent the handwritten digits and perfectly fill the gaps between the every two adjacent points. Particularly, there are no vibration near the endpoints. In addition, the CPDs can well represent the self-overlapping curves (e.g. "2","4","7" and "8").

In order to examine the classification performance of CPDs representation, we used two third of the handwritten digits in the data set for training, the remaining ones for testing, and then computed the CPDs and Fourier descriptor of these handwritten digits respectively. Afterward, Euclidean distance was used as the metric to perform *k*-nearest neighbors classification of the test set from the training set, where *k* ranges from 1 to 10. The resulting classification error rates are shown in Fig. 5.

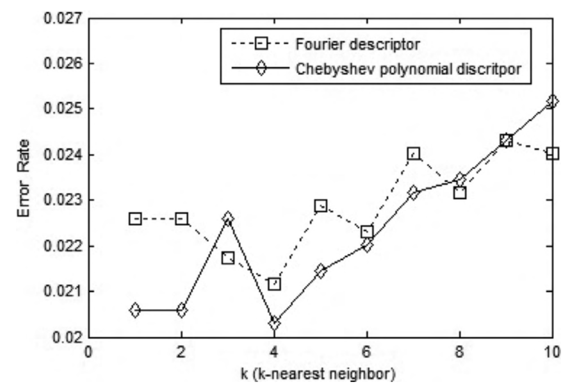
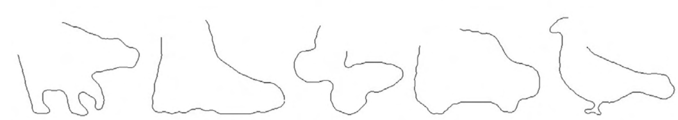


Fig. 5. Classification error rate for CPDs versus Fourier descriptor.

(a) the five original shapes (bear,boot,butterfly,car,pegion)



(b) the reconstructed curves from CPDs

(c) the reconstructed curves from FDs

Fig. 6. The comparison of CPDs and FDs.

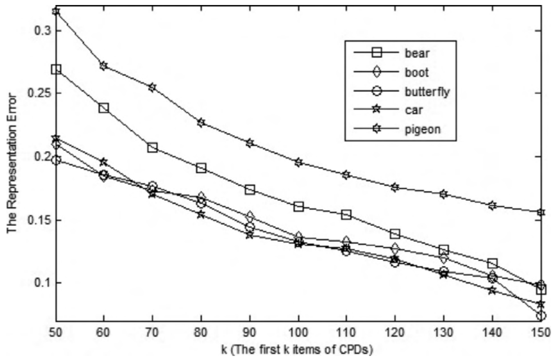


Fig. 7. The representation errors from the first k (from 50 to 150 with step size 10) items of CPDs.

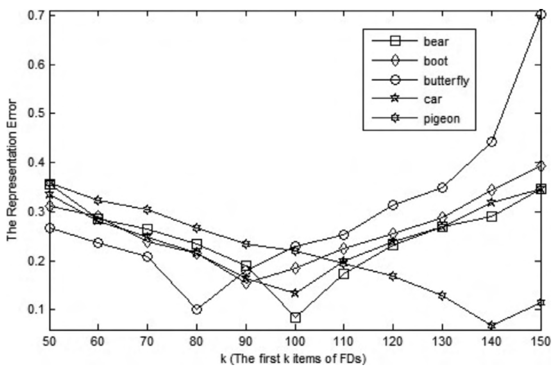


Fig. 8. The representation errors from the first k (from 50 to 150 with step 10) items of FDs.

It is clear from this figure that the CPDs perform consistently better than Fourier descriptors except when k takes values 3 and 10. This is mainly because CPDs can not only represent the open curve accurately and stably but also capture its main features well. In particular, CPDs can stably represent those points near both the ends of open curves. We illustrated this further by the following experimental evaluation on five shapes which were selected randomly from MPEG-7 Shape 1 Part B data set [5]. In order to have open curves, a part of every shape was cut away as shown in Fig. 6(a). We used CPDs and FDs to represent these shapes, and then reconstructed them using the first 50 items of their descriptors respectively (see Fig. 6(b) and (c)). It is clear from Fig. 6(b) that CPDs accurately represented all the five shapes. Conversely, from Fig. 6 (c), we can see that FDs failed to work well for them. The reason is that both the ends of the represented curves suffered from severe oscillation, especially for bears and boots.

In order to further explore the stability of CPDs and FDs, we increased the number of items of CPDs and FDs from 50 to 150 with step size 10 to respectively reconstruct the five shapes, followed by using the representation error (9) to evaluate the accuracy of representations. Noting that the representation errors of FDs can be defined in a similar way as (9). Fig. 7 shows the representation errors of CPDs for the five shapes and Fig. 8 shows the ones of FDs. From Fig. 7, we can see that all the representation errors for the five shapes strictly decrease as the number of the items increases. This is consistent with the Theorem 2. In addition, we can also find that representation errors for pigeons and bears are larger than ones for the other three shapes. This is because the two shapes are not far smoother than other shapes, leading to their CPDs having slower convergence rate [6]. On the other hand, from Fig. 8, we find that the representation errors for butterflies and boots start to increase rapidly when the

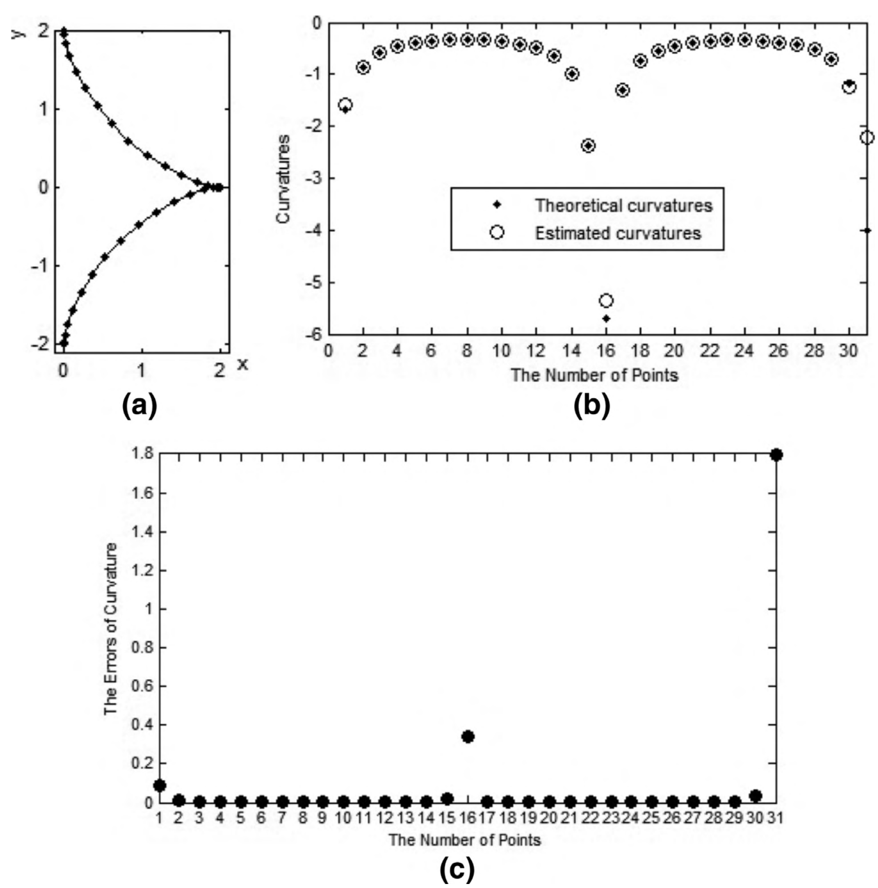


Fig. 9. Estimating curvatures with CPDs. (a) Sampling some points from an astroid. (b) Estimating curvatures of an astroid from its CPDs. (c) Errors of estimated curvatures.

numbers of the items are 80 and 90 respectively. For bears and cars, the representation errors increase when the numbers are 100. This means the represented curves for these shapes suffer from more and more severe oscillation as the number of items of FDs increase. Consequently, In contrast to FDs, CPDs are suitable for representing open shapes.

5.2. Estimating curvatures of digitalized curves

The CPDs have a powerful ability to represent open curves. Hence, the curvature of an open curve can be accurately estimated from its CPDs. In order to illustrate the effectiveness of estimating curvatures with CPDs, we consider an analytical open curve, namely astroid, the parametric equation of which is given by $x = 2\cos^3 t, y = 2\sin^3 t$. Although this defines a closed curve for $t \in [0, 2\pi)$, we consider $t \in [-\pi/2, \pi/2]$ in order to have an open curve. Specifically, let the variable t take values from $-\pi/2$ to $\pi/2$ using step size 0.1, then, we can obtain the digitized curve consisting of 32 points, which is presented in Fig. 9(a). According to the parametric equation of the astroid, we computed its curvatures at each point and their resulting values, denoted by dots, are shown in Fig. 9(b). On the other hand, we used the CPDs to represent the digitized curve and then computed its curvatures at each point. The resulting curvatures, denoted by circle, are also shown in Fig. 9(b). In Fig. 9(c), the errors between the estimated curvatures and theoretical curvatures were presented. From these two figures, we can see that the estimated curvatures with CPDs almost are equivalent to the theoretical curvatures except at the last point and the cusp point on the digitized curve. As shown in Fig. 9(b), the 16th point is a cusp point where the error of the curvature is 0.4 (see Fig. 9(c)). Clearly, this cusp point is a discontinuous one, which leads to large errors of curvatures. In order to reduce the error, we can further sample more points nearby the cusp point. In fact, the use of more sampling points means more CPDs are used to represent the digitized curves [6]. The large error of the curvature also appears at the last point (i.e. end point). In a similar way, we can also use more sampling points to reduce it. It should noted that use of CPDs to represent open curves can perform well on the analysis of curves (e.g. the estimation of curvatures) without needing to pad the curves with some data.

6. Conclusion

In this paper, we present the concept of Chebyshev polynomial descriptor (CPD) and introduce it to represent open curves. Based on analysis of properties of Chebyshev polynomial, we present the general formula for the computation of CPDs and the parametric equations of the reconstructed curve. In addition, we investigate the properties of CPDs, including its stability, the similarity-invariant and invariance under different starting points. Due to the exact mathematical expression of CPDs, we discuss how to use CPDs to compute the curvature of an original open curve.

In our experiments, we show the powerful ability of CPDs to represent open curves. For example, the representation of handwritten digits is very stable, especially at the those points nearby endpoints. Furthermore, the CPD representations are easy to implement due to the simple formula for CPDs and less computational cost. We show this good characteristic by computing the curvatures of digitized curves.

Future work should study affine invariant CPDs and how to apply the invariants to recognition of objects. In addition, we would like to extend Chebyshev polynomial descriptors to three dimension surfaces representation.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants 71372188, National Center for International Joint Research on E-Business Information Processing under grant 2013B01035 and Natural Science Foundation of Jiangsu Province China under grant BK2011832.

References

[1] T. Aibara, K. Ohue, Y. Matsuoka, Human face recognition by p-type fourier descriptor, in: Proceedings of the SPIE, 1606, 1991, pp. 198–203.
[2] L. Costa, Estimating derivatives and curvature of open curves, Pattern Recognit. 35 (11) (2002) 2445–2451.
[3] M. Cui, J. Femiani, J. Hu, P. Wonka, A. Razdan, Curve matching for open 2D curves, Pattern Recognit. Lett. 30 (1) (2009) 1–10.
[4] G.H. Granlund, Fourier preprocessing for hand print character recognition, IEEE Trans. Comput. C-21 (2) (1972) 195–201.
[5] L. Latecki, R. Lakamper, T. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, 1, 2000, pp. 424–429, doi:10.1109/CVPR.2000.855850.
[6] J.C. Mason, D. Handscomb, Chebyshev Polynomials, Chapman & Hall/CRC, 2003.
[7] F. Mokhtarian, A.K. Mackworth, A theory of multiscale curvature-based shape representation for planar curves, IEEE Trans. Pattern Anal. Mach. Intell. 14 (8) (1992) 789–805.
[8] M. Mori, S. Uchida, H. Sakano, Global feature for online character recognition, Pattern Recognit. Lett. 35 (1) (2014) 142–148. (Frontiers in Handwriting Processing)
[9] R. Mukundan, S. Ong, P. Lee, Image analysis by Tchebichef moments, IEEE Trans. Image Process. 10 (9) (2001) 1357–1364, doi:10.1109/83.941859.
[10] J. Ni, C.V. Ravishankar, Indexing spatio-temporal trajectories with efficient polynomial approximations, IEEE Trans. Knowl. Data Eng. 19 (5) (2007) 663–678.
[11] M. Rouhani, A. Sappa, Implicit polynomial representation through a fast fitting error estimation, IEEE Trans. Image Process. 21 (4) (2012) 2089–2098.
[12] E. Tanaka, Y. Tamura, M. Hosoya, T. Shiroishi, Protrusion fourier descriptor: skeleton-based representation of open curves, Forma 23 (1) (2008) 9–18.
[13] Y. Uesaka, A new fourier descriptor applicable to open curves, Electron. Commun. Jpn. [Part I: Communications] 67 (8) (1984) 166–173.
[14] C. Yue, H.-J. Su, Q.J. Ge, A hybrid computer-aided linkage design system for tracing open and closed planar curves, Comput. Aided Des. 44 (11) (2012) 1141–1150.
[15] C.T. Zahn, R.Z. Roskies, Fourier descriptors for plane closed curves, IEEE Trans. Comput. 21 (3) (1972) 269–281.
[16] B. Zheng, J. Takamatsu, K. Ikeuchi, An adaptive and stable method for fitting implicit polynomial curves and surfaces, IEEE Trans. Pattern Anal. Mach. Intell. 32 (3) (2010) 561–567.
[17] Z. Zheng, H. Iwata, Y. Hirata, Y. Tamura, Quantitative evaluation of the degree of sprout leaf bending of rice cultivars using p-type fourier descriptors and principal component analysis, Euphytica 163 (2) (2008) 259–266.
[18] B. Zhong, K.-K. Ma, J. Yang, Curvature scale-space of open curves: theory and shape representation, in: 2013 20th IEEE International Conference on Image Processing (ICIP), 2013, pp. 3671–3675.