



# Shape description and recognition by implicit Chebyshev moments<sup>☆</sup>

Gang Wu<sup>a,\*</sup>, Linmin Xu<sup>a</sup>

*Institute of Multimedia Computation Technology, Nanjing University of Finance and Economics, Nanjing 210003, China*



## ARTICLE INFO

### Article history:

Received 17 February 2018

Revised 15 February 2019

Accepted 30 August 2019

Available online 31 August 2019

### Keywords:

Chebyshev polynomial

Chebyshev moment

Implicit Chebyshev moment

Shape representation

## ABSTRACT

Chebyshev Moments(CMs) have been applied to representation and recognition of 2D object shapes in image processing and computer vision. However they still suffer from poor representation power and difficulty in computing invariants for shapes. In this work, we present Implicit Chebyshev Moments (ICMs) to overcome these issues. Firstly, we use Euclid distance transformation to generate a series of level sets based on a given shape. Secondly, we fit an implicit Chebyshev polynomial to the data set consisting of the obtained level sets together with all the boundary points on the original shape and call the obtained coefficients of the fitted implicit Chebyshev polynomial ICMs. Finally, we propose a new approach to derive geometric invariants based on ICMs. In addition, we also develop an algorithm for the determination of a suitable degree for implicit Chebyshev polynomials before representing a given shape. Experimental results show the ICMs are more efficient for representing complex shapes than CMs.

© 2019 Elsevier B.V. All rights reserved.

## 1. Introduction

Shape is one of the important perceptual features in image analysis, object recognition and content-based image retrieval paradigm. In order to extract the features of shapes, many shape representation methods have been developed during the past decades. Generally, these methods can be categorized into two types: contour-based representations and region-based representations. Compared with contour-based representations, region-based representations are applicable to more generic shapes (e.g., shapes with holes, or consisting of multiple disconnected regions) due to their ability to exploit both boundary and interior content of shapes. Among the region-based representations, moments, in particular Chebyshev moments(CMs) [13,14], are commonly used approaches. CMs exhibit superior representation capability over other moments (e.g. Legendre moments [21], Zernike moments [4]). The main reason is that their basis functions are orthogonal in the domain of image coordinate space, which can eliminate the requirement for any discrete approximation to avoid introducing quantization errors. On the other hand, the application of CMs to object recognition needs computation of their invariants. Hence, many methods for computation of invariants have been proposed. For example, the translation, scale and affine invariants of Chebyshev moments were also proposed in [10,11,19,20,23]. The results of experiments in these literatures demonstrate that the invariants of

CMs can work better than other moment invariants in terms of scale, rotation and affine invariant recognition in noisy, noise-free and smooth distortion condition.

Despite the advantages mentioned above, Chebyshev moment representation for shapes still suffers from three major issues as follows:

1. Poor feature representation capability for shapes. A shape appears to be a binary image as shown in Fig. 1(a), usually to be set to one the pixels located inside the shape and to zero the pixels located outside. The use of Chebyshev approximated function of two variables to represent a given shape means that the function is used to approximate an indicator function. Fig. 1(d) shows the graph of the indicator function reconstructed from Chebyshev moments, consisting of shape plane (red regions) and background plane (blue regions). Obviously, the representation of an indicator function needs higher order Chebyshev moments, namely, higher degree Chebyshev polynomials. The reason is that the function is discontinuous at the boundary points of the shape, and accordingly leading to weaker representation capability for shapes.
2. Inefficient Chebyshev moment invariants for object recognition. Low-order Chebyshev moments describe the global features of a shape and high-order moments capture its fine details. For a complex shape, we need to combine numerous high-order and low-order moments to describe the shape due to the poor feature representation capability of Chebyshev moments. However, high-order moments are sensitive to slightly deformed shapes (e.g. original shapes resulting from color noise [16]), which usually leads to failing to recognize complex objects.

<sup>☆</sup> Editor: Prof. S. Sarkar.

\* Corresponding author.

E-mail address: [wugang69@gmail.com](mailto:wugang69@gmail.com) (G. Wu).

3. High computational cost. Chebyshev moment invariants proposed in [11,20,23] belong to a family of the algebraic invariants. As pointed out in [15,20], computation of these algebraic invariants inherits the complexities from the Chebyshev polynomials defined in terms of hypergeometric functions and accordingly needs to use algebraic iteration and performs linear combinations of original Chebyshev moments. These two computation procedures not only lead to high computational cost but also tend to introduce accelerated errors.

In this work, we propose Implicit Chebyshev Moments (ICMs) as a representation method for eliminating the above three weaknesses. ICMs can efficiently describe the shapes, especially complicated shapes. The complicated shapes mean that the shapes can only be represented by the implicit polynomials of degree greater than six in theory, including shapes composed of a few disconnected components, shapes that intersect themselves, and shapes with holes [8]. The lower order ICMs work as well as the higher order CMs in representing the same shape. The basic reason for this is that ICMs are obtained by approximating a continuous function, but the reverse is true for CMs. As an example, we respectively used CMs and ICMs of order up to 10 to represent the same crown shape (see Fig. 1(a)). Their reconstructed shapes are shown in Fig. 1(b) and (c), respectively. From the two plots, we can see that the CMs of order up to 10 fail to represent the crown shape. Conversely, the ICMs can work well. Most importantly, both algebraic invariants and geometric invariants can be derived based on ICMs. However, only algebraic invariants for CMs. The further detailed discussion on these will be given in the following sections.

This paper is organized as follows. Below we discuss the background. Particularly, we review the definition and formula for Chebyshev polynomial in two variables, and focus on interpolation of discrete data sets with Chebyshev approximated function. In Section 3, we first define an implicit Chebyshev polynomial curve and then discuss how to fit an implicit Chebyshev polynomial curve to the generated level sets together with all the boundary points on the given shape. In addition, we develop an algorithm for determination of a moderate degree for the fitted implicit Chebyshev polynomial and a method to determine the distance between two adjacent level sets before generating level sets. Section 5 presents simulation results obtained in comparing ICMs with CMs in representing complex shapes. Section 6 summarizes and concludes the paper.

## 2. Chebyshev polynomials in two variables approximation

A Chebyshev polynomial in two variables  $P_{ij}(x, y)$  are defined as

$$P_{ij}(x, y) = T_i(x)T_j(y) \quad (1)$$

where  $T_i(x)$  and  $T_j(y)$  are Chebyshev polynomials of the first kind. They are polynomials in  $x$  of degree  $i$  and  $y$  of degree  $j$ , respectively. In general, the formula for a Chebyshev polynomial  $T_m(t)$  is given by  $T_m(t) = \cos(m \arccos(t))$ .

The Chebyshev polynomials are frequently used to approximate a continuous function or to interpolate a discrete data set. For a given 2-D data set  $\{z_{kl}\}$ ,  $k = 1, 2, \dots, N$ ;  $l = 1, 2, \dots, M$ , we can construct a linear combination of the Chebyshev polynomial function of two variables  $P_{ij}(x, y)$ , denoted by  $f(x, y)$ , to interpolate the data set  $\{z_{kl}\}$  at the associated coordinate points  $\{(x_k, y_l)\}$ , where  $x_k$  and  $y_l$  are given by  $x_k = \cos(k - 0.5)\pi/N$ , ( $k = 1, 2, \dots, N$ ) and  $y_l = \cos(l - 0.5)\pi/M$ , ( $l = 1, 2, \dots, M$ ), respectively. Then, according to [12], the formula for the function  $f(x, y)$  takes the following form

$$f(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} c_{ij} P_{ij}(x, y) \quad (2)$$

where the formula for computation of the coefficients  $c_{ij}$  can be found in [5,12,17]

The interpolation function  $f(x, y)$  in (2) is called Chebyshev approximated function. Clearly, the function  $f(x, y)$  satisfies the equations  $f(x_k, y_l) = z_{kl}$ ,  $k = 1, 2, \dots, N$ ;  $l = 1, 2, \dots, M$ . Note that the coefficients  $c_{ij}$  are 2-D discrete cosine transformation coefficients, which can be computed by using fast Fourier transform algorithm [12] or fast computation algorithm proposed in [1,2,9].  $\{c_{ij}, i = 0, 1, \dots, N-1, j = 0, 1, \dots, M-1\}$  are also called Chebyshev moments.

From the formula (2), It follows that reconstruction of the polynomial of two variables  $f(x, y)$  of degree  $n$  requires  $n^2$  multiplications of moments and Chebyshev polynomials. Implementing moments up to the  $n$ th order requires the total number of multiplications (denoted by  $S(n)$ ) as follows [18]:

$$S(n) = n^2/2 + 3n/2 + 1 \quad (3)$$

## 3. Representation of shape using implicit Chebyshev polynomial curves

Observing Fig. 1(c), we can find that a shape completely depends on all the boundary points on the shape and no any other information is contained inside and outside of the shape. However, the CMs representations introduce a great deal of extraneous information (e.g. set 1 to pixels located inside of shapes and 0 to the pixels located outside as shown in Fig. 1(a)) so as to be able to apply Chebyshev moments to the representation of the shape. This leads to the inefficiency of the CMs representations. In order to avoid the problem, we propose a method for the representation of a shape based on implicit Chebyshev polynomial curves by only using the shape boundary information instead of its associated binary image.

### 3.1. Formulation

Formally, an implicit Chebyshev polynomial curve is defined by the equation  $f(x, y) = 0$ , where the formula for  $f(x, y)$  is the same as (2). We also call the function  $f(x, y)$  the implicit Chebyshev polynomial function. For convenience, The equation  $f(x, y) = 0$  can be rewritten in vector form. Specifically, we first construct a matrix consisting of elements  $c_{ij}$  in (2). Then, we map the matrix to a vector  $A$  by stacking the rows of the matrix.

$$A = [c_{00} \ c_{01} \ \dots \ c_{0(M-1)} \ c_{10} \ c_{11} \ \dots \ c_{1(M-1)} \ \dots \\ c_{(N-1)0} \ c_{(N-1)1} \ \dots \ c_{(N-1)(M-1)}]^T$$

Clearly,  $A$  is the  $(NM \times 1)$  coefficient column vector of implicit Chebyshev polynomial function.

In a similar way, the  $(NM \times 1)$  column vector function of monomials of implicit Chebyshev polynomial function in (2), denoted by  $X(x, y)$ , can be given by

$$X(x, y) = [P_{00}(x, y) \ P_{01}(x, y) \ \dots \ P_{0(M-1)}(x, y) \ \dots \\ P_{(N-1)0}(x, y) \ \dots \ P_{(N-1)(M-1)}(x, y)]^T$$

Combining the two above generated vectors, we can rewrite the equation defining implicit Chebyshev polynomial curves in vector form as follows

$$f(x, y) = X(x, y)^T A = 0 \quad (4)$$

Denoting the number of coefficients of  $f(x, y)$ , as well as the dimension of the coefficient vector  $A$ , with  $r$ , we have  $r = NM$ . We also call  $f(x, y) = 0$  the equation for implicit Chebyshev polynomials curve of degree  $r$  and call the coefficient vector ( $A$ ) implicit Chebyshev moments (ICMs). Actually, an implicit Chebyshev polynomial curve consists of all points  $(x, y)$  at which the values of the implicit Chebyshev polynomial function  $f(x, y)$  are zeros. The set of

these points, denoted by  $Z_f$ , is called the zero set of implicit Chebyshev polynomial function  $f(x, y)$ . It should be noted that the main difference between ICMs and CMs is that ICMs are obtained by fitting an implicit Chebyshev polynomial curve to a shape, but CMs are obtained by applying the 2-D discrete cosine transformation to the associated binary image of the shape. ICMs do not belong to a kind of implicit moment invariants discussed in [6]. The reason is that implicit moment invariants have nothing common with implicit curves and implicit polynomials and they are proposed to differentiate explicit moment invariants because implicit moment invariants only provide description of a pair of images rather than a single image.

### 3.2. Fitting implicit Chebyshev polynomial curves to shapes

As mentioned in Section 1, in order to improve the feature representation capability for shapes, we only represent the shape with implicit Chebyshev polynomials, namely, fit an implicit Chebyshev polynomial curve to the shape boundary. The fitting problem is to find ICMs that lead to an implicit Chebyshev polynomial curve that best fits the given shape boundary. Specifically, for a given shape boundary, denoted by  $\Gamma_0$ , consists of  $(x_i, y_i), i = 1, 2, \dots, n_0$ , ICMS fitting requires all points  $(x_i, y_i)$  satisfy the system of equations  $f(x_i, y_i) = 0, i = 1, 2, \dots, n_0$ , or, in vector form,

$$M_0^T A = 0 \quad (5)$$

where  $M_0 = [X(x_1, y_1), X(x_2, y_2), \dots, X(x_{n_0}, y_{n_0})]$ . In general, due to  $n_0 \gg NM$ , the classical least-squares algorithm can be used to solve the above system of equations to obtain the ICMS ( $A$ ). However, the obtained ICMS usually have poor representation of shapes. The main reason is that the system of equations belongs to the type of homogeneous linear system of equations [3]. In order to overcome this problem, we use Min-Max method [7] to fit the implicit Chebyshev polynomial curve.

Defining  $X_x(x, y)$  and  $X_y(x, y)$  as the partial derivative vector function of the monomials vector function  $X(x, y)$  with respect to  $x$  and  $y$ , respectively, and denoting the local unit normal vector at the boundary point  $(x_i, y_i)$  on the shape by  $(v_i, w_i)$ , we can construct two matrices: one is denoted by  $M_x$ , each column of which is  $X_x(x_i, y_i), i = 1, 2, \dots, n_0$  and another  $M_y$ , each column is  $X_y(x_i, y_i)$ . In addition, we construct two vectors:  $v = [v_1 \ v_2 \ \dots \ v_{n_0}]$  and  $w = [w_1 \ w_2 \ \dots \ w_{n_0}]$ . Combing  $M_0, M_x$  and  $M_y$  to form a big matrix:  $M = [M_0 \ M_x \ M_y]^T$ ,  $v$  and  $w$  to form a vector:  $b = [0 \ v \ w]$ , where  $0$  represents a row vector of zeros of length  $n_0$ , then, we have the system of equations  $MA = b$ . Using the linear least squares algorithm to solve the above equation, we can obtain the coefficient vector (ICMS) of the fitted implicit Chebyshev polynomial curve:

$$A = (M^T M)^{-1} M^T b \quad (6)$$

Note that the local unit normal vector  $(v_i, w_i)$  can be estimated by calculating the normal vector to a regression line which is fitted to several points in the neighborhood of the point  $(x_i, y_i)$  on the given shape boundary. We fitted implicit Chebyshev polynomial curve to the crow shape boundary in Fig. 1(a) with Min-Max algorithm. The Fig. 2(a) shows the fitting result. From the figure, we can see that the zero set of fitted implicit Chebyshev polynomial fails to represent the crown shape due to the presence of the numerous artifacts and outliers around the shape boundaries. In order to improve the fitting accuracy, we fit gradients of the implicit Chebyshev polynomial function to the normal vectors of level sets generated from shape by using distance transform [3], with the desired result that the fitted implicit Chebyshev polynomial functions have as few extreme points as possible. We name this algorithm as gradient constraints of multiple level sets (GCMLS) fitting methods, which will be further detailed in the next section.

### 3.3. Fitting implicit Chebyshev polynomials with multi level sets

The Min-Max algorithm forces the gradients of the fitted implicit Chebyshev polynomial function to have similar orientations to the normal vectors of the shape image. This can be viewed as interpolation of normal vectors of the shape image with the second-kind Chebyshev approximated function. Clearly, the interior and exterior of the shape contain no any interpolated normal vectors, and thus leading to the failure in fitting implicit Chebyshev polynomial curves as illustrated in Fig. 2(a). In order to eliminate this issue, we need to impose some constraints on interior and exterior of the shape without introducing extraneous information, specifically, generating a series of level sets with Euclidean distance transformation and computing their corresponding normal vectors. As an example, Fig. 2(b) illustrates these level sets. The distance between any two adjacent level sets is set to 5. We compute the normal vectors at every point on these level sets, and specify their directions to point towards the inside of shape as depicted by Fig. 2(c). Our GCMLS algorithm requires to interpolate not only the normal vectors of the shape boundaries but also interpolate those of generated level sets with the gradients of the implicit Chebyshev polynomial function. Supposed that all the generated level sets consist of  $n_1$  points, denoted by  $\{(x_i, y_i), i = n_0 + 1, n_0 + 2, \dots, n_0 + n_1\}$ , where  $n_0$  is the number of points on the shape boundaries, then, we can collect all the  $n_0 + n_1$  points together to produce a new system of equations and then solve it to obtain the ICMS. Its formulation is similar to (6). This is the computational procedure for GCMLS fitting algorithm. Fig. 2(c) shows the original crown shape boundary together with all the generated level sets and their normal vectors. We fitted an implicit Chebyshev polynomial function of degree 10 to them. Fig. 1(e) shows the graph of the fitted implicit Chebyshev polynomial function, and Fig. 1(c) shows its zero set. From these two figures, we can see that the use of the proposed GCMLS fitting algorithm can significantly improve the fitting accuracy of implicit Chebyshev polynomials. Unlike the fitting result of the crown shape using the Min-Max algorithm in Fig. 2(a), the fitted zero set shown in Fig. 1(c) contains no any artifact and outlier. On the other hand, it is clear from the Fig. 1(e) that the fitted implicit Chebyshev polynomial function is a curved surface, having only six local maximal points. This means that the implicit Chebyshev polynomials of lower degree, namely, fewer parameters, can represent the crow shape accurately.

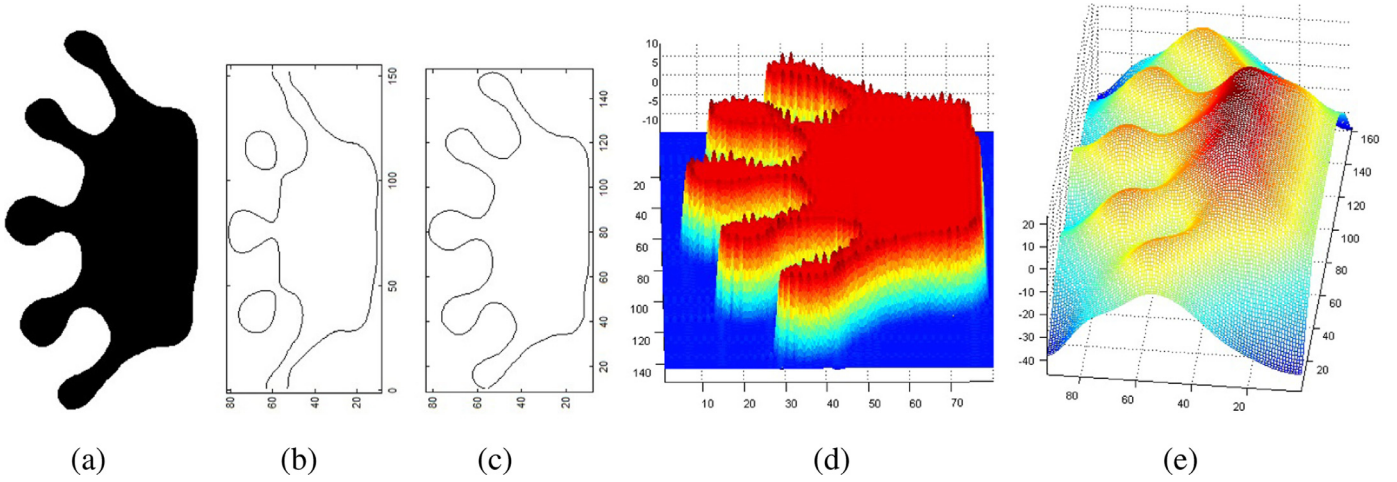
It should be noted that unlike 3L fitting algorithm [3] which forces the fitted function to reach a fixed value at all points on the same level set, the GCMLS fitting algorithm only uses the gradients of the level sets generated from the original data sets (shape), and hence, it does not introduce extraneous information.

Although the GCMLS algorithm works well, the two issues still remain unsolved. The first one is how to determine the distance between two adjacent level sets. Another is how to determine the degree of fitted implicit Chebyshev polynomial. These two problems are discussed in the following sections.

### 3.4. Determination of degrees of implicit Chebyshev polynomials

For efficient representation of object shapes, we need to choose as low a degree as possible for an implicit Chebyshev polynomial under certain criterion of fitting accuracy specified. As noted in [22], choosing a moderate degree for a fitted polynomial is difficult problem. Generally, its degree is determined by trying different degrees several times and selecting the best one from the results. Clearly, this method is low efficient due to the high computational cost. Implicit Chebyshev polynomials exhibit a good property: low-order ICMS describe the global features of a shape and high-order ICMS capture its fine details. Parseval theorem [12] also illustrates this idea. For a given continuous function of two variables  $f(x, y)$





**Fig. 1.** (a) The original crown shape. (b) The shape of the reconstructed crown with Chebyshev moments up to the tenth order. (c) The shape of the reconstructed crown with implicit Chebyshev moments up to the tenth order. (d) The graph of the Chebyshev approximated function obtained by using Chebyshev moments. (e) The graph of the implicit Chebyshev polynomials function obtained by using implicit Chebyshev moments.

and its Chebyshev series expansion  $\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} c_{ij} P_{ij}(x, y)$ , denoted by  $S_t$ , we have the following equation

$$\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} c_{ij}^2 = C \|f(x, y)\|^2$$

where  $C$  is a constant. The above equation is called Parseval's formula. It is clear from this formula that the Chebyshev series  $S_t$  is convergent, and  $c_{ij}^2$  approaches zero as  $i$  and  $j$  tend to infinity. Hence, the partial sum of the Chebyshev series  $S_t$  approach  $\|f(x, y)\|^2$  as  $i, j$  are sufficiently large. Similarly, denoting an image of size  $N \times M$  by  $\{f(i, j), i = 1, 2, \dots, N; j = 1, 2, \dots, M\}$ , applying Chebyshev transformation to the image to get Chebyshev moments, denoted by  $c_{ij}, i = 0, 1, 2, \dots, N; j = 0, 1, 2, \dots, M$ , we have the version for discrete data of Parseval's formula

$$\sum_{i=0}^N \sum_{j=0}^M c_{ij}^2 = \sum_{i=1}^N \sum_{j=1}^M f(i, j)^2 \quad (7)$$

For convenience of discussion, we make the following notation.

$$S_0 = \sum_{i=0}^N \sum_{j=0}^M c_{ij}^2, \quad S_k = \sum_{i=0}^k \sum_{j=0}^l c_{ij}^2, \quad S(k) = S_0 - S_k$$

where  $N \leq M, k = 1, 2, \dots, N; l = \lfloor k * M/N \rfloor, \lfloor l \rfloor$  denotes the integer part of  $l$ . Then, it follows from the above analysis that  $S(k)$  decreases as  $k$  increases. This means that  $S(k)$  is a decreasing function and take approximately a fixed value as  $k$  is greater than a certain value. We name  $S(k)$  a coefficient order function(COF) of Chebyshev transformation for a given image. Fig. 3 illustrates the coefficient order function of Chebyshev transformation for the crown shape shown in Fig. 1(a). In this figure, the maximum curvature point, the curvature point with curvature value being equal to 1, the last curvature point with curvature value being equal to 0.1 on the curve for coefficient order function are also shown and their magnification are further shown in Fig. 3(b). We call the three points changing point, accurate point and stable point, respectively. It is clear from the two figures that we can find at changing point, the coefficient order function value begins to decrease slowly and furthermore, gradually becomes approximately constant at the stable point. This implies that an implicit Chebyshev polynomial curves will represent the crown shape well when its degree is chosen to be greater than the abscissa of the changing point. In particular, an implicit Chebyshev polynomial curve will represent the crown shape

stably when its degree is chosen to be equal to the abscissa of the stable point. That is, the implicit Chebyshev polynomial curve can not improve its representation accuracy when its degree is chosen to be greater than the abscissa of the stable point. In order to give a tradeoff between the representation accuracy and representation efficiency, we choose the abscissa of accurate point as the moderate degree for fitted implicit Chebyshev polynomial curve if abscissa of accurate point is greater than the one of the changing point and smaller than the one of the stable point. Otherwise we choose the abscissa of the stable point as the suitable degree.

As an example, Fig. 3 shows the graph of the coefficient order function of the distance transformation image generated from the crown shape, along with the changing point, accuracy point and stable point. Clearly, their abscissas are 1216 and 28, respectively. According to above algorithm, we should choose 16 as the degree of implicit Chebyshev polynomial curve to represent the crown shape. Actually, experimental results in Section 5 demonstrate that the implicit Chebyshev polynomial of degree 16 can very accurately represent the crown shape.

After choosing the degrees of implicit Chebyshev polynomials, we need to determine the moderate distance between two adjacent level sets so that we can generate a series of level sets from the interior and exterior of the shape. Clearly, choosing the over small distance will produce too many level sets, leading to high computation cost and overfitting. Conversely, choosing the over large distance can not produce enough level sets, leading to failure in representation of shapes. Supposed that we use an implicit Chebyshev polynomial of degree  $k$  to represent a binary image of size  $N \times M \{f(i, j), i = 1, 2, \dots, N; j = 1, 2, \dots, M\}, N \geq M$ , then the Chebyshev nodes along the horizontal direction are as follows

$$x_i = \cos\left(\frac{(i - \frac{1}{2})\pi}{k}\right), \quad (i = 1, 2, \dots, k) \quad (8)$$

Computing the distance  $D_i$  between every two adjacent Chebyshev nodes, we have

$$D_i = |x_{i+1} - x_i|, \quad (i = 1, 2, \dots, k - 1) \quad (9)$$

It is easy to verify that  $D_1$  and  $D_2$  are the first and the second minimum values of the set  $\{D_i\}$ , respectively. In order to ensure the accuracy of the representation, the distance between any two interpolated data points is not larger than the minimum distance between any two Chebyshev nodes. That is, the distance is not larger than  $D_1$ . Considering that  $x_1$  is the abscissa of leftmost point in an image and usually far away from the boundary of shapes, then, the

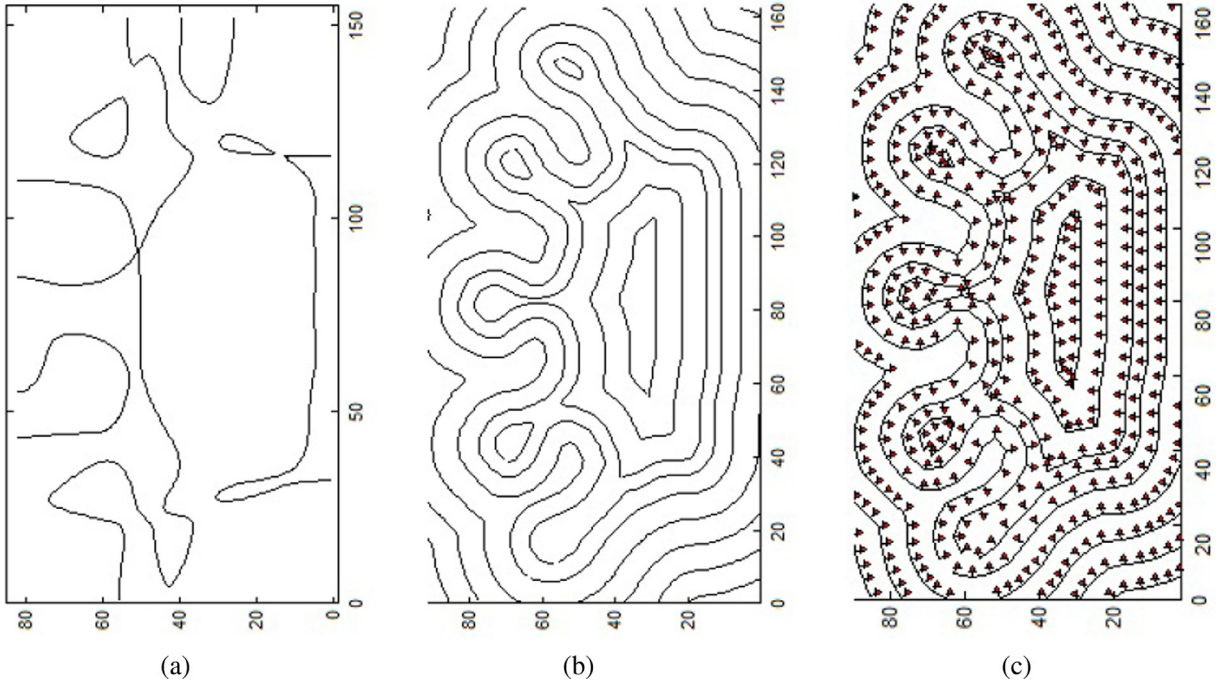


Fig. 2. (a) Fitting result using the Min-Max algorithm. (b) Generated level sets using distance image. (c) Normal vector of each level set.

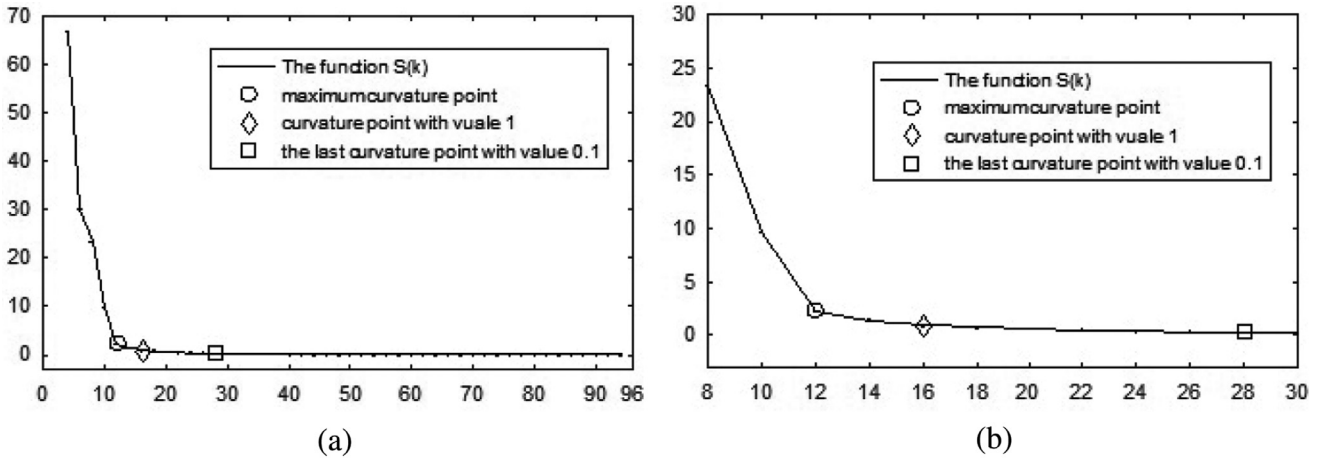


Fig. 3. (a) Coefficient order function for distance image of the crown shape shown in Fig. 1(a). (b) Magnification of (a) in the vicinity of the maximum curvature point.

distance is required to be smaller than or equal to  $D_2$ . Hence, denoting the distance between two adjacent level sets by  $G_i$ , we have

$$G_i \leq D_2 N \quad (10)$$

This implies that for a binary image of size  $N \times M (N \geq M)$ , if using an implicit Chebyshev polynomial of degree  $k$  to represent it, then, we should ensure the distance of any two adjacent level sets is not larger than  $D_2 N$ . This provides a method for the GCMLS algorithm to obtain the suitable level sets from the given binary image.

We call the above methods for determining the degree for implicit Chebyshev polynomial and distance between two adjacent level sets Parseval algorithm.

#### 4. The geometry invariants based on implicit Chebyshev polynomials

As discussed in Section 1, both algebraic invariants and geometric invariants can be derived from an implicit Chebyshev polynomial. In this section, we focus on its geometric invariants because

geometric invariants are easy to compute and insensitive to noise and slight deformation. Clearly, the ratio of the implicit Chebyshev polynomial function values at any two points is a kind of geometric invariants under Euclid or affine transformation. From this property, we can derive area invariants and perimeter invariants. Specifically, For a given implicit Chebyshev polynomial function  $f(x, y)$ , computing its maximum value (denoted by  $M$ ), letting  $v_i = iM/m, i = 0, 1, 2, \dots, m$ , where  $m$  takes integer values and smaller than  $M$ , then we can create  $m + 1$  level curves by letting  $f(x, y) = v_i, i = 0, 1, 2, \dots, m$ . Denoting the areas of  $m + 1$  level curves by  $A_i, i = 0, 1, 2, \dots, m$  and perimeters by  $P_i, i = 0, 1, 2, \dots, m$ , we respectively define area invariants  $\{\hat{A}_i\}$  and perimeter invariants  $\{\hat{P}_i\}$  as follows

$$\begin{aligned} \hat{A}_i &= A_i/A_{i-1}, i = 1, 2, \dots, m \\ \hat{P}_i &= P_i/P_{i-1}, i = 1, 2, \dots, m \end{aligned} \quad (11)$$

These  $m$  area invariants and  $m$  perimeter invariants can be used to recognize shapes undergoing Euclid or affine transformation.








| Original Image |  |      |      |      |      |      |      |      | Maximum Order of Moments |    |
|----------------|---|------|------|------|------|------|------|------|--------------------------|----|
|                | (a)   | (b)  | (c)  | (d)  | (e)  | (f)  | (g)  | (h)  |                          |    |
| Using ICMs     |  | 2643 | 3115 | 3849 | 3982 | 2264 | 1727 | 3302 | 3310                     | 8  |
|                |  | 984  | 1007 | 1695 | 1376 | 962  | 921  | 1576 | 1376                     | 16 |
|                |  | 611  | 667  | 836  | 778  | 656  | 698  | 922  | 863                      | 24 |
| Using CMs      |  | 2777 | 2785 | 4054 | 4273 | 2888 | 2789 | 4480 | 4294                     | 8  |
|                |  | 1206 | 1317 | 1841 | 1794 | 1273 | 1246 | 2084 | 1731                     | 16 |
|                |  | 744  | 799  | 1018 | 1118 | 789  | 774  | 1362 | 1117                     | 24 |

Fig. 4. Image reconstruction of a complicated crown shape under eight different affine transformations.

## 5. Experimental results

In this section, we illustrate the effectiveness of the implicit Chebyshev polynomial representation through competing it with traditional Chebyshev polynomial moments. To make comparative evaluation possible, we set our experiments in some precondition. The degrees for implicit Chebyshev polynomials and distance between two adjacent level sets are determined using Parseval algorithm in the experiments unless otherwise specified. The results of fitted implicit Chebyshev polynomials are evaluated using reconstruction error  $\epsilon$  [14]. The formula for the error  $\epsilon$  is

$$\epsilon = \sum_{i=1}^n \sum_{j=1}^n |f(i, j) - \hat{f}(i, j)| \quad (12)$$

where  $f(i, j)$  indicates the original image and  $\hat{f}(i, j)$  indicates reconstructed image.

### 5.1. Comparison of the representation power

For the comparison of the relative performance between implicit Chebyshev moments and Chebyshev moments, we respectively applied three affine transformations to a complex crown image shown in Fig. 4 (a), specifically, shearing the image by shear factor of 0.5 along horizontal direction, respectively scaling the image by the scale factor of 0.5 along the horizontal direction and vertical direction. The transformed images are shown in Fig. 4 (b)–(d), respectively. Furthermore, we rotated the four images by degree 65 in the counterclockwise direction. The rotated images are shown in Fig. 4 (e)–(h). We reconstructed a sequence of each crown image respectively using ICMs and CMs as the maximum order of moments used in the reconstruction is varied from eight to 24 with a step size of eight. The sequences of reconstructed images are shown in Fig. 4. We can see that the reconstructed images using ICMs can capture the general feature of crown images when the maximum order of moments is eight and capture the fine details when 24. However the reconstructed images using CMs cannot exhibit the same representation accuracy. The corresponding reconstruction errors shown in Fig. 4 also illustrate these.

Table 1  
Area invariant values of ICMs for crown shapes.

















| Area invariants   | A1     | A2     | A3     | A4     | A5     | A6     | A7     | A8     |
|---|--------|--------|--------|--------|--------|--------|--------|--------|
|  | 0.6441 | 0.5120 | 0.4223 | 0.3345 | 0.2625 | 0.1982 | 0.1415 | 0.0945 |
|  | 0.6186 | 0.4954 | 0.3946 | 0.2993 | 0.2274 | 0.1617 | 0.1111 | 0.0710 |
|  | 0.6224 | 0.5104 | 0.4139 | 0.3252 | 0.2544 | 0.1907 | 0.1339 | 0.0862 |
|  | 0.6239 | 0.4976 | 0.4055 | 0.3166 | 0.2391 | 0.1757 | 0.1213 | 0.0777 |
|  | 0.6422 | 0.5163 | 0.4297 | 0.3420 | 0.2711 | 0.2090 | 0.1510 | 0.0982 |
|  | 0.6090 | 0.5042 | 0.4156 | 0.3204 | 0.2453 | 0.1834 | 0.1236 | 0.0762 |
|  | 0.6311 | 0.5192 | 0.4363 | 0.3553 | 0.2775 | 0.2098 | 0.1532 | 0.1025 |
|  | 0.6153 | 0.4973 | 0.4037 | 0.3134 | 0.2353 | 0.1689 | 0.1177 | 0.0727 |
| Range   | 0.0351 | 0.0238 | 0.0417 | 0.0561 | 0.0502 | 0.0480 | 0.0421 | 0.0315 |

Table 2  
Area invariant values of DCMs for image crown.

| Area invariants   | A1     | A2     | A3     | A4     | A5     | A6     | A7     | A8     |
|---|--------|--------|--------|--------|--------|--------|--------|--------|
|    | 0.7793 | 0.5504 | 0.4452 | 0.3511 | 0.2786 | 0.2187 | 0.1571 | 0.1055 |
|    | 0.7281 | 0.5222 | 0.4057 | 0.3117 | 0.2387 | 0.1686 | 0.1142 | 0.0709 |
|    | 0.7707 | 0.5424 | 0.4357 | 0.3000 | 0.2180 | 0.1598 | 0.1104 | 0.0680 |
|    | 0.7193 | 0.5134 | 0.3775 | 0.2790 | 0.1947 | 0.1273 | 0.0792 | 0.0451 |
|    | 0.7748 | 0.5438 | 0.4414 | 0.3535 | 0.2781 | 0.2158 | 0.1563 | 0.1047 |
|    | 0.7280 | 0.5141 | 0.4012 | 0.3107 | 0.2362 | 0.1672 | 0.1132 | 0.0709 |
|   | 0.7532 | 0.5303 | 0.4312 | 0.3412 | 0.2638 | 0.1964 | 0.1368 | 0.0801 |
|  | 0.7108 | 0.4913 | 0.3736 | 0.2755 | 0.1930 | 0.1266 | 0.0783 | 0.0451 |
| Range   | 0.0685 | 0.0591 | 0.0716 | 0.0780 | 0.0856 | 0.0921 | 0.0788 | 0.0604 |

### 5.2. Comparison of ability of objection recognition

As discussed in Section 1, CMs are obtained by interpolating an indicator function with a Chebyshev approximated function of the two variables. So, only algebraic invariants can be derived from CMs. However, ICMs are obtained by fitting an implicit Chebyshev polynomial in the two variables to a curved surface. Hence, not only algebraic invariants but also geometric invariants can be derived from ICMs. Compared with algebraic invariants of CMs, the algebraic invariants of ICMs will be more efficient to recognize objects due to their strong representation power and insensitivity to noise.

In order to have geometric invariants of CMs, we interpolated the distance transformation image generated from a given shape with the Chebyshev approximated function and name the obtained CMs as DCMs. For a fair comparison, we used the same implicit Chebyshev polynomial of degree 30 to respectively compute ICMs and DCMs for the eight original images in Fig. 4 and then computed their eight area invariants ( $\hat{A}_i, i = 1, 2, \dots, 8$ ) and the results are shown in Tables 1 and 2, respectively. Observing the last row in these two tables, we can find that the range of each area invariant  $\hat{A}_i$  of ICMs are consistently smaller than those of DCMs. Specifically, all ranges of area invariants of ICMs are roughly smaller than 0.05. However the reverse is true for DCMs. This means that the area invariants of ICMs can be used to efficiently recognize the objects undergoing affine transformation.

To further validate the efficiency of invariants of ICMs for recognition of objects, we randomly selected ten complex images from an image data set created by the Laboratory for Engineering Man/Machine System, Brown University, and then respectively fitted implicit Chebyshev polynomial functions to these ten



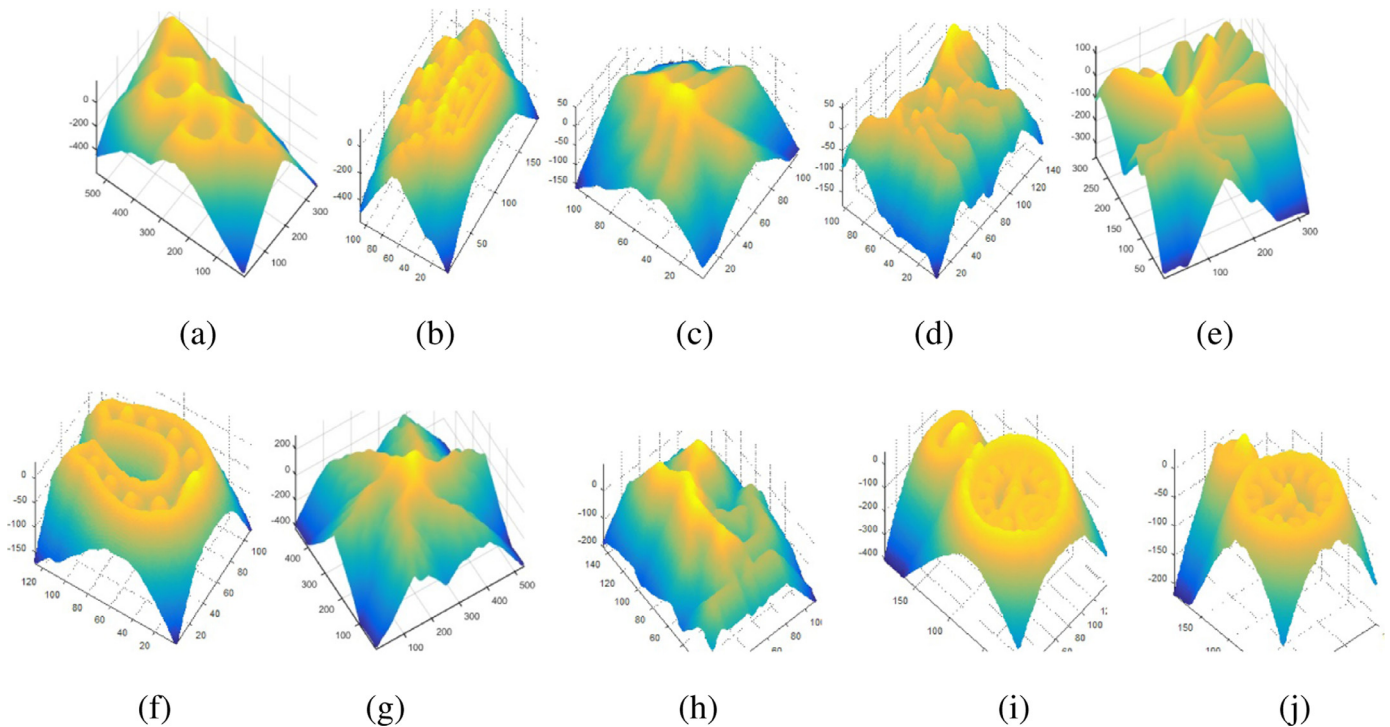












Fig. 5. The graphs of the ten fitted implicit Chebyshev polynomial functions.

Table 3  
Area invariant values of ICMs for some shapes.

| Area invariants   | A1     | A2     | A3     | A4     | A5     | A6     | A7     | A8     |
|---|--------|--------|--------|--------|--------|--------|--------|--------|
|  | 0.8298 | 0.7078 | 0.5988 | 0.4149 | 0.2143 | 0.1308 | 0.0653 | 0.0258 |
|  | 0.9153 | 0.7931 | 0.6980 | 0.6144 | 0.4405 | 0.2639 | 0.1735 | 0.1196 |
|  | 0.8064 | 0.6084 | 0.4474 | 0.3414 | 0.2607 | 0.1951 | 0.1437 | 0.0961 |
|  | 0.8161 | 0.5629 | 0.3710 | 0.2389 | 0.1610 | 0.1101 | 0.0795 | 0.0558 |
|  | 0.7669 | 0.5721 | 0.4475 | 0.3614 | 0.2858 | 0.2193 | 0.1629 | 0.1126 |
|  | 0.9798 | 0.7029 | 0.4481 | 0.2324 | 0.1246 | 0.0864 | 0.0671 | 0.0510 |
|  | 0.7701 | 0.5978 | 0.4274 | 0.2836 | 0.1605 | 0.0967 | 0.0543 | 0.0256 |
|  | 0.7717 | 0.6538 | 0.5627 | 0.4780 | 0.3964 | 0.3093 | 0.2251 | 0.1475 |
|  | 0.9704 | 0.9233 | 0.8388 | 0.3982 | 0.3342 | 0.2751 | 0.2214 | 0.1648 |
|  | 0.8907 | 0.5185 | 0.3443 | 0.1961 | 0.0932 | 0.0416 | 0.0255 | 0.0184 |

images. The graphs of these ten fitted implicit polynomial functions are shown in Fig. 5. It is clear that all the graphs are curved surfaces and Fig. 5 (b), (i) and (j) show that the three curved surfaces exhibit very complicated structure. The reason is that their corresponding object shapes (row 2, 9, 10 and first column in Table 3) are complicated. We computed the area invariants for the ten images and resulting area invariant values ( $A_i, i = 1, 2, \dots, 8$ ) are shown in Table 3. To measure the similarity between the original crown shape and each of these ten shapes, we computed Euclid distances between the area invariant vector of original crown shape (the first row in Table 1) and the area invariant vectors of the ten shapes (row 1 through 10 in Table 3), respectively. The resulting Euclid distances are 0.3575, 0.5870, 0.1906, 0.2596, 0.1478, 0.4463, 0.2427, 0.3417, 0.6895, 0.3981. Clearly, all the Euclid distances are greater than 0.1. On the other hand, we also computed the Euclid distances between the area invariant vector of the original crown shape and the ones of the transformed crown shape (row 2 through 8 in Table 1), respectively. The resulting Euclid distances are 0.0835, 0.0297, 0.0544, 0.0206, 0.0521, 0.0375, 0.0669.

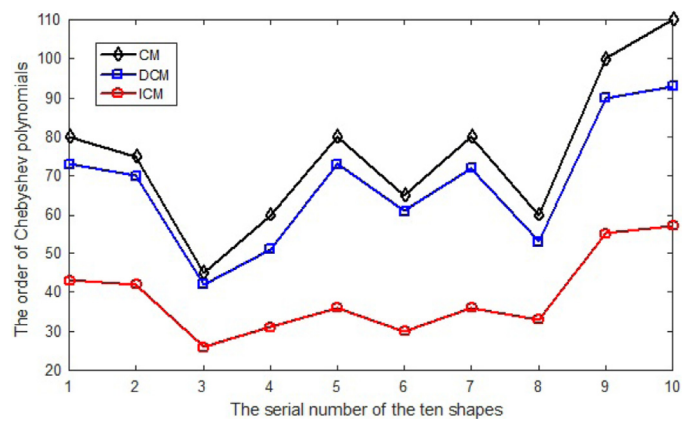


Fig. 6. The comparison of the efficiency and recognition ability.

It is easy to see that all the Euclid distances are smaller than 0.1. This means that the area invariants of ICMs enable us to identify the transformed images from other images.

To evaluate the efficiency and recognition ability of the proposed algorithm, we reconstructed the above ten shapes from CMs, DCMs and ICMs, respectively, and then compared the efficiency and recognition ability of the three moments under the same reconstruction errors. Specifically, we reconstruct a given shape image with moments of up to order  $n$  and then binarize reconstructed image, followed by computing the reconstruction error between the obtained binary image and the original shape image using the formula (12). If the reconstruction error is greater than a threshold value (e.g. 10), we reconstruct the shape image with moments of up to order  $n + 1$  and repeat the above process, otherwise the reconstruction process stops and the maximum order  $n$  is recorded. This means that the moment items from order zero up to the recorded order  $n$  can almost completely represent the given shape. Obviously, the recorded order is the required

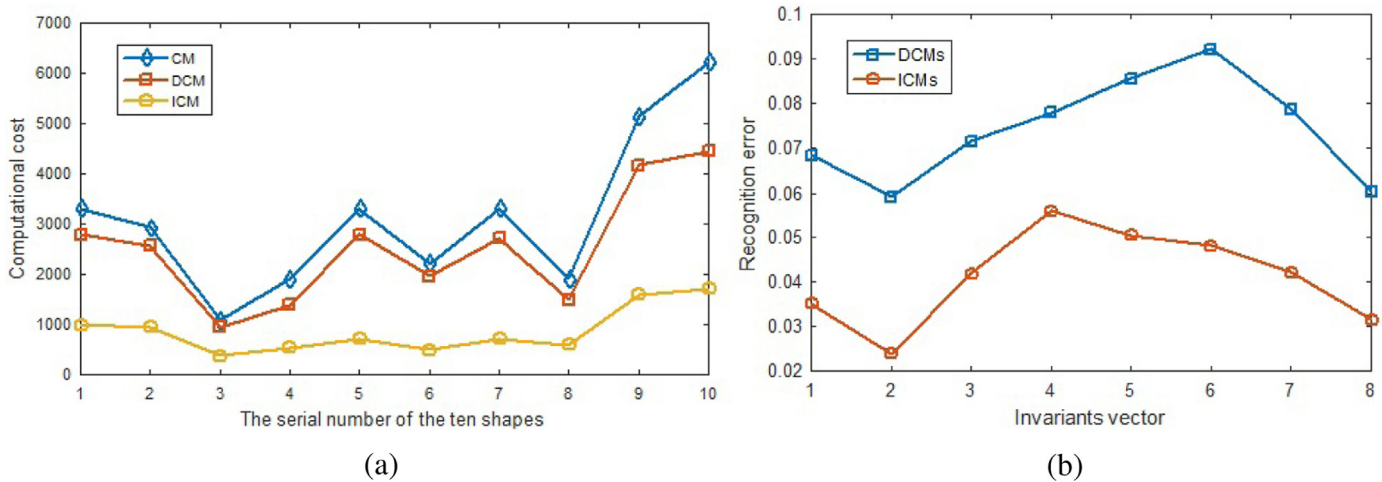


Fig. 7. (a) The comparison of computational costs. (b) The comparison of the efficiency and recognition ability.

maximum order for representing the shapes. In this way, we can obtain the maximum orders of CMs, DCMs and ICMs for representing these ten shapes.

Fig 6 shows these obtained maximum orders. From this figure, we can find that CMs and DCMs need moments of approximately the same maximum orders for representing these shapes. However, ICMs need nearly half the maximum orders of CMs and DCMs. This means that the proposed algorithm exhibits stronger representation power. On the other hand, due to the lower computational costs and the smaller computational errors, ICMs exhibit much higher recognition ability than CMs and DCMs. For instance, we use algebraic invariants proposed in [23] to recognize a shape. If the algebraic invariants obtained by using 25 moment items of CMs can be used to efficiently recognize a given shape, only 5 moment items of ICMs can reach the same recognition accuracy. This can greatly reduce the computational errors and accordingly significantly improve recognition accuracy because algebraic invariants obtained from computation of moments are sensitive to computational errors. Fig. 7(a) shows the total number of multiplications required to evaluate the moments of CMs, DCMs and ICMs for the ten shapes. Clearly, the computational costs of ICMs are far lower than the costs of CMs and DCMs. Fig. 7(b) shows the errors in recognizing the crown shape shown in Fig. 4(a) by using DCMs and ICMs, respectively. From the figure, we see that the recognition error rates of ICMs are more than half the ones of DCMs. This is consistent with the conclusion of ICMs which have stonger representation power and smaller computational errors than DCMs and CMs.

## 6. Conclusion

In this paper, we introduce Implicit Chebyshev Moments (ICMs) for representation of complex shapes and present the Gradient Constraints of Multiple Level Set(GCMLS) algorithm to fit an implicit Chebyshev polynomial curve to a shape. Based on analysis of properties of Chebyshev polynomials, we also develop an algorithm for determination of a suitable degree for fitting an implicit Chebyshev polynomial function to a given shape and a method to determine the suitable distance between two adjacent level sets when using GCMLS for fitting. In simulations, we compared ICMs with CMs in reconstructing some complex shapes using the same maximum order of moments, and the experimental results demonstrate that the representation power of ICMs performs consistently better than that of CMs. We also computed the area invariants of ICMs for some complex shapes and their transformed versions. The

resulting invariants illustrate their ability of efficient recognition of these complex shapes.

Future work should study more efficient geometry variants of ICMs and extend ICMs for 2D shapes to for 3D objects

## Declaration of Competing Interest

The authors have declared that no conflict of interest exists.

## References

- [1] S.H. Abdullhussain, A.R. Ramli, S.A.R. Al-Haddad, B.M. Mahmmod, W.A. Jassim, On computational aspects of Tchebichef polynomials for higher polynomial order, *IEEE Access* 5 (2017) 2470–2478.
- [2] B.H.S. Asli, R. Paramesran, C.-L. Lim, The fast recursive computation of Tchebichef moment and its inverse transform based on Z-transform, *Digit. Signal Process.* 23 (5) (2013) 1738–1746.
- [3] M. Blane, Z. Lei, The 3L algorithm for fitting implicit polynomial curves and surfaces to data, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (3) (2000) 298–313.
- [4] Z. Chen, S.-K. Sun, A Zernike moment phase-based descriptor for local image representation and matching, *IEEE Trans. Image Process.* 19 (1) (2010) 205–219.
- [5] E. Cheney, *Introduction to Approximation Theory*, Chelsea Publ. Company, New York, 1982.
- [6] J. Flusser, J. Kautsky, F. Šroubek, Implicit moment invariants, *Int. J. Comput. Vis.* 86 (1) (2009) 72.
- [7] A. Heizer, M. Barzohar, D. Malah, Stable fitting of 2d curves and 3d surfaces by implicit polynomials, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (10) (2004) 1283–1294.
- [8] D. Keren, D. Cooper, J. Subrahmonia, Describing complicated objects by implicit polynomials, *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (1) (1994) 38–53.
- [9] M.H. Khalid, Fast computation of accurate Zernike moments, *J. Real-Time Image Process.* 3 (1) (2008) 97–107.
- [10] M.H. Khalid, M.D. Mohamed, New set of quaternion moments for color images representation and recognition, *J. Math. Imaging Vis.* 60 (5) (2018) 717–736.
- [11] Q. Li, H. Zhu, Q. Li, Image recognition by affine Tchebichef moment invariants, in: *International Conference on Artificial Intelligence and Computational Intelligence*, Springer, 2011, pp. 472–480.
- [12] J.C. Mason, D. Handscomb, *Chebyshev Polynomials*, Chapman & Hall/CRC, 2003.
- [13] R. Mukundan, Some computational aspects of discrete orthonormal moments, *IEEE Trans. Image Process.* 13 (8) (2004) 1055–1059.
- [14] R. Mukundan, S. Ong, P. Lee, Image analysis by Tchebichef moments, *IEEE Trans. Image Process.* 10 (9) (2001) 1357–1364.
- [15] C.-Y. Pee, S. Ong, P. Raveendran, Numerically efficient algorithms for anisotropic scale and translation Tchebichef moment invariants, *Pattern Recognit. Lett.* 92 (2017) 68–74.
- [16] T. Tasdizen, J. Tarel, Improving the stability of algebraic curves for application, *IEEE Trans. Image Process.* 9 (3) (2000) 405–416.
- [17] L. Thede, *Practical Analog and Digital Filter Design*, Artech House, 2004.
- [18] G. Wang, S. Wang, Recursive computation of Tchebichef moment and its inverse transform, *Pattern Recognit.* 39 (1) (2006) 47–56.
- [19] X. Wang, G. Shi, F. Guo, A comment on "Translation and scale invariants of Tchebichef moments" by Hongqing Zhu, *Pattern Recognit.* 77 (2018) 458–463.
- [20] H. Wu, S. Yan, Computing invariants of Tchebichef moments for shape based image retrieval, *Neurocomputing* 215 (26) (2016) 110–117.
- [21] H. Zhang, H. Shu, C. Gouenou, J. Zhu, Affine Legendre moment invariants for image watermarking robust to geometric distortions, *IEEE Trans. Image Process.* 20 (8) (2011) 2189–2199.



- [22] 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.
- [23] H. Zhu, H. Shu, T. Xia, L. Luo, J.L. Coatrieux, Translation and scale invariants of Tchebichef moments, *Pattern Recognit.* 40 (9) (2007) 2530–2542.