Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 2730–2741
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Cubic Bézier approximation of a digitized curve

Sarbajit Pal[a,*], Pankaj Ganguly[b], P.K. Biswas[c]

[a]*Variable Energy Cyclotron Centre, 1/AF Bidhan Nagar, Kolkata 700064, India*
[b]*ABV-Indian Institute of Information Technology and Management, Gwalior 474 003, India*
[c]*Indian Institute of Technology, Kharagpur 721302, India*

Received 27 September 2005; received in revised form 5 January 2007; accepted 17 January 2007

Abstract

In this paper we present an efficient technique for piecewise cubic Bézier approximation of digitized curve. An adaptive breakpoint detection method divides a digital curve into a number of segments and each segment is approximated by a cubic Bézier curve so that the approximation error is minimized. Initial approximated Bézier control points for each of the segments are obtained by interpolation technique i.e. by the reverse recursion of De Castaljau's algorithm. Two methods, two-dimensional logarithmic search algorithm (TDLSA) and an evolutionary search algorithm (ESA), are introduced to find the best-fit Bézier control points from the approximate interpolated control points. ESA based refinement is proved to be better experimentally. Experimental results show that Bézier approximation of a digitized curve is much more accurate and uses less number of points compared to other approximation techniques.

Keywords: Bézier curve; Break point; Evolutionary search algorithm; Integral square error

---

1. Introduction

With the advancement of image processing algorithms, machine vision techniques are finding wide applications in object recognition, visual inspection of industrial parts, security surveillance, etc. Most of these applications work with 2-D images, which are projections of 3-D object on a 2-D plane. Boundary representation plays an important role in pattern analysis application [1]. It is a well-known fact that shape information of 2-D object (or projections) is mostly contained in the 2-D object's (projections) boundary, which is obtained as a set of discrete points of a digital curve in the 2-D image. Thus for machine vision applications dealing with object shape information, it is extremely important to extract meaningful descriptors from digital curves. Such features must follow two important properties. (i) The descriptors should be compact so that, high-level recognition algorithms can work with these descriptors efficiently. (ii) The descriptors should represent the digital curve as accurately as possible, or in other words, if we reconstruct the digital curve from the descriptors, the error between the original digital curve and the reconstructed digital curve should be minimum.

The digital curve is divided into segments and each digital segment is fitted with a piece of analytic curve, which can be a line segment, a circular arc, or a higher order curve. Polygonal approximation is the simplest approach. It fits digital curves by finding the break points and connecting them with line segments. Teh and Chin [2] proposed the concept of region of support of each point and achieved excellent polygonal approximations of digital curves. Some other attractive optimization approaches for polygonal approximations were proposed in Refs. [3–6]. Yin [7] applied the genetic algorithms to determine the optimal polygons. Unfortunately, not all of the digital curves are suitable for polygonal approximation. Smooth curves are usually hard to be fitted with polygons. To improve the performance of approximation, circular arcs and high order curves should be used.

Using circular arcs for segment representation produces better approximation at a higher level of computational complexity [8,9]. Only a few circular arcs are required to approximate a smooth curve. For a polygon like curve, a set of circular arcs

---

[*] Corresponding author. Tel.: +91 33 2337 1231; fax: +91 33 2334 6871.
E-mail addresses: sarbajit@veccal.ernet.in (S. Pal), gangulypankaj@yahoo.com (P. Ganguly), pkb@ece.iitkgp.ernet.in (P.K. Biswas).

0031-3203/$30.00 © 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.01.019

with small curvatures can produce satisfactory approximation. Pei and Horng [8] introduced a method for curve fitting using circular arcs. Later, they developed an optimal approach based on dynamic programming [9].

Many heuristics have also been developed for curve fitting using a combination of line segments and circular arcs [10–12]. The underlying concept of fitting with lines and arcs is that using line segment is perceptually better than using circular arc while their geometric-fitting errors are comparable. The method proposed by Rosin and West [10] first applies a polygonal approximation technique to find a good polygonal representation. Consecutive line segments are replaced by an arc if such replacement reduces error. This two-pass approach is a direct improvement of polygonal approximation. It suffers from a problem that a breakpoint for line segments is not necessarily suitable for the circular arc after merging. Horng and Li [11] have developed a dynamic programming approach while Sarkar et al. [12] have used genetic algorithm.

Bézier curves are piecewise polynomial functions that can provide local approximation of contours using small number of parameters. This is useful because human perception of shapes is based on curvatures of the contours [13]. The advantage of representing a digital curve by cubic Bézier curve is that it has relatively few control points. Smooth edges enhance the quality of the semi-synthetic images and hence provide more natural look than approximations with circular arcs and/or line segments.

Cinque et al. [14] describe shape using variable number of Bézier curve segments. They extracted object contour by applying border following algorithm [15]. They have sampled the shape by a set of contour points. The boundary segment between two chosen contour points (break points) has predetermined number of points. The boundary is better approximated by choosing large number of boundary segments or break points. The Bézier control points are approximated by computing the magnitude and gradient of the end point tangent vector of the Bézier segments. The error calculation was not reported to evaluate the performance of the fitness algorithm. Bhuiyan and Hama [16] presented an actor identification system by approximating the facial features by Bézier curve. The hand drawn facial feature curve is approximated by single cubic Bézier curve. So, no judgment of segmentation was required for representing a contour having multiple curvature and complex shape. They have assumed that the second and third control points are located at the tangent line made by the curve. To avoid the computation complexity $(N/5)$th and $(N-(N/5))$th points are generated from the approximated control points and compared with the corresponding curve points. This process was continued until the fitness error becomes less than the predefined minimum possible error.

None of the above mentioned work discussed on selection of the break points when a complex digital curve is to be approximated by multiple segments of Bézier curves. The break points or dominant points are considered as representative features for the object contours, because they reserve the significant features of the digitized curve of the images. Following Attneave's [1] observation, there are many approaches developed for detecting dominant points. But any efficient algorithm for fitting digital data with cubic Bézier is absent.

In this paper, we propose a simple and efficient method for detection of break points of chain-coded curves suitable for Bézier approximation. The approximation of the digitized curves is achieved by joining the successive approximated Bézier curves and straight line. To minimize the error of approximation, a two-dimensional logarithmic search algorithm (TDLSA) or an efficient evolutionary search algorithm (ESA) is used, just after the initial findings.

The paper is organized as follows. A brief overview of chain code, Bézier curve and De Casteljau's representation are presented in Section 2. The propositions for detection of break points for Bézier approximation are presented in Section 3. Section 4 describes method of segment representation of digitized curve with Bézier curve. Section 5 shows the different measures for approximation of digital curves and experimentally obtained result and Section 6 summarizes the concluding remarks.

2. Review of chain code and Bézier curve

2.1. Chain code

Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. It is a coding technique in which a string of codes represent the sequence of difference vectors between consecutive pixels. This coding convention for boundary representation of discrete curves was introduced by Freeman [17] and later elaborated by him [18,19]. The chain code assigns an integer c_i varying from 0 to 7 according to its direction to a vector, where $\frac{1}{4}\pi c_i$ is the angle between the x-axis and the vector, for $i=1,2,3,\ldots,n$ as shown in Fig. 1(a). A digital curve can be expressed as the n chain codes, and is denoted as $c_1c_2c_3\ldots c_n$, where $c_i=c_{i\pm n}$ (i.e. all indices are modulo n). The 8-directional chain code of the Fig. 1(b) would be as 10701131000776.

2.2. Bézier curve

A Bézier curve $C(t)$ of degree n can be defined in terms of a set of control points p_i, $(i=0,1,2,\ldots,n)$ and is given by Bézier [20]:

$$C(t)=\sum_{i=0}^{3} p_i Bz_{i,n}(t),$$

where each term in the sum is the product of a blending function $Bz_{i,n}(t)$ and a control point p_i. $Bz_{i,n}(t)$ are called Bernstein polynomials and are defined by

$$Bz_{i,n}(t)={}^{n}B_i t^i(1-t)^{n-i},$$

where ${}^{n}B_i$ is the binomial coefficient

$${}^{n}B_i=n!/i!(n-i)!.$$

Fig. 2 shows a curve of degree three and the associated control points p_0^0, p_1^0, p_2^0 and p_3^0. Points p_0^0 and p_3^0 (known as anchors)

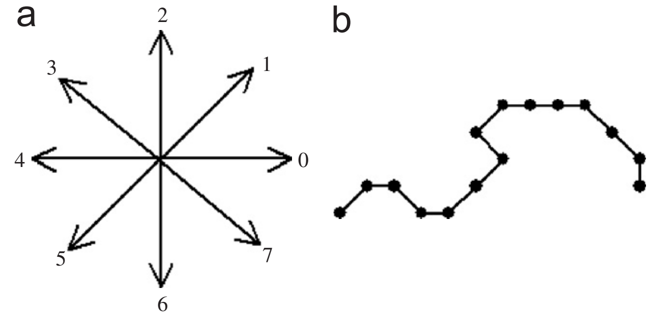

Fig. 1. (a) Chain code direction vectors and (b) curve representing 8-directional chain code 10701131000776.

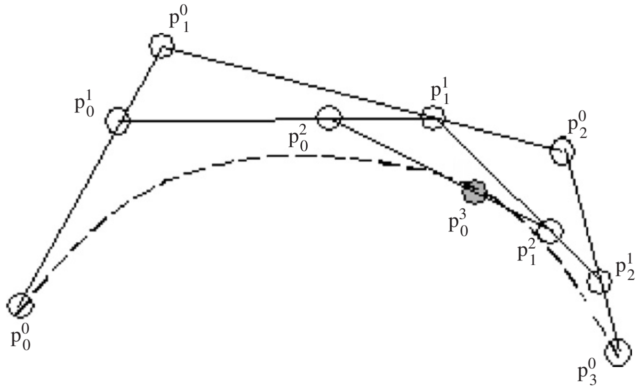

Fig. 2. The recursive interpolation to produce the point p_3^0 on the Bézier curve.

coincide with the end points of the curve but the control points p_1^0 and p_2^0 (known as handles) are not coincident with the curve. The shape of the curve is determined entirely from the position of the control points.

The advantage of representing a smooth digital curve by cubic Bézier curve is that it goes through the end points of the digital curve. So p_0^0 and p_3^0 i.e. two extreme control points of cubic Bézier could be taken as the end points of the digital curve. Two intermediate control points that determine the curvature or the bending of the curve are to be interpolated. As t in the parametric equation is increased, p_0^1 is increased towards p_1^0 and $Bz_{0,3}$ and $Bz_{1,3}$ primarily determines the curve shape. The control points p_1^0 and p_2^0 have most effect when $t = \frac{1}{3}$ and $\frac{2}{3}$, respectively, because the value of p_3^0 for a particular value of t is obtained by summing the values of four blending functions.

2.3. *De Casteljau representation of Bézier curves*

De Casteljau developed an alternative representation, which is very useful in computer graphics. The De Casteljau algorithm generates points on the curve by repeated linear interpolation [21]. Fig. 2 shows how this algorithm works for the cubic case. Starting with the control polygon p_i^0 $(i = 0, 1, 2, 3)$ the edges are subdivided in the ratio t as given above and the superscript corresponds to the level of recursion, starting at 0. These points, p_i^1, are then connected in order and the resultant edges are subdivided to get the point p_i^2. The recursion stops when only one edge remains, the subdivision of this edge producing the point on the curve $p_0^3(t)$.

For interpolating two intermediate Bézier control points, the reverse recursion of De Castaljau's algorithm is proposed. The known point $p_0^3(t)$ can be expressed in terms of cubic Bézier control points as given below,

$$p_i^r(t) = (1-t)p_i^{r-1}(t) + tp_{i+1}^{r-1}(t),$$

where $r = 0, 1, 2, 3$ and $i = 0, 1, 2, 3$ and $p_i^0(t) = p_i$, p_0, p_1, p_2, p_3 be the four control points of a cubic Bézier curve.

3. Proposed breakpoint detection method

Like the dominant point detection algorithm proposed by Sarkar [5], this algorithm is also based on manipulation with chain codes only and requires no knowledge of the co-ordinates of the digitized curve points. A digital curve C can be defined as a set of 'n' consecutive points. That is

$$C = \{p_i(x_i, y_i) | i = 1, 2, \ldots, n\},$$

where n is the number of points, p_i is the ith point with the co-ordinate (x_i, y_i). For a continuous curve, the curvature at a point is defined as the rate of change of slope as a function of arc length. However, this definition of curvature does not hold for a digital curve, since an exact mathematical definition is not available for the digital curve. For the polygonal approximation, most of the existing algorithms focus on the curvature estimation by using the information that can be extracted from the neighbours. From the previous studies it is seen that choosing the suitable length of support region is very important in determining the curvature. But approximated Bézier curve supports up to two full π-rotation of corresponding chain code representation of digital curve. So the length of support region varies widely for the nature of the curve. The initial dominant points are identified as the point with local maximum curvature. The candidates of the dominant points are called the break points. In addition, break point elimination is conducted to reduce the computations during Bézier curve approximation. An adaptive scheme is applied to obtain the break points based on the following rules.

Rule 1: Introduce a dominant point d_j if there is a crossover or junction and rearrange the chain code in a way that each segment does not contain any crossover.

Rule 2: If $|c_{i+1} - c_i| > 1$ with any rotational transformation, where, c_{i+1} and c_i are the chain codes of the $(i+1)$th and ith links of the curve, then p_{i+1} is taken as a break point.

Rule 3: We consider three consecutive dominant points as d_i, d_{i+1}, d_{i+2}. The error of fitting the Bézier curve between d_i, d_{i+1} is represented as $\mathrm{err}(d_i, d_{i+1})$. From the initial set of dominant points obtained by applying Rule 1, delete the break point d_{i+1} from the dominant point candidates, if one of the

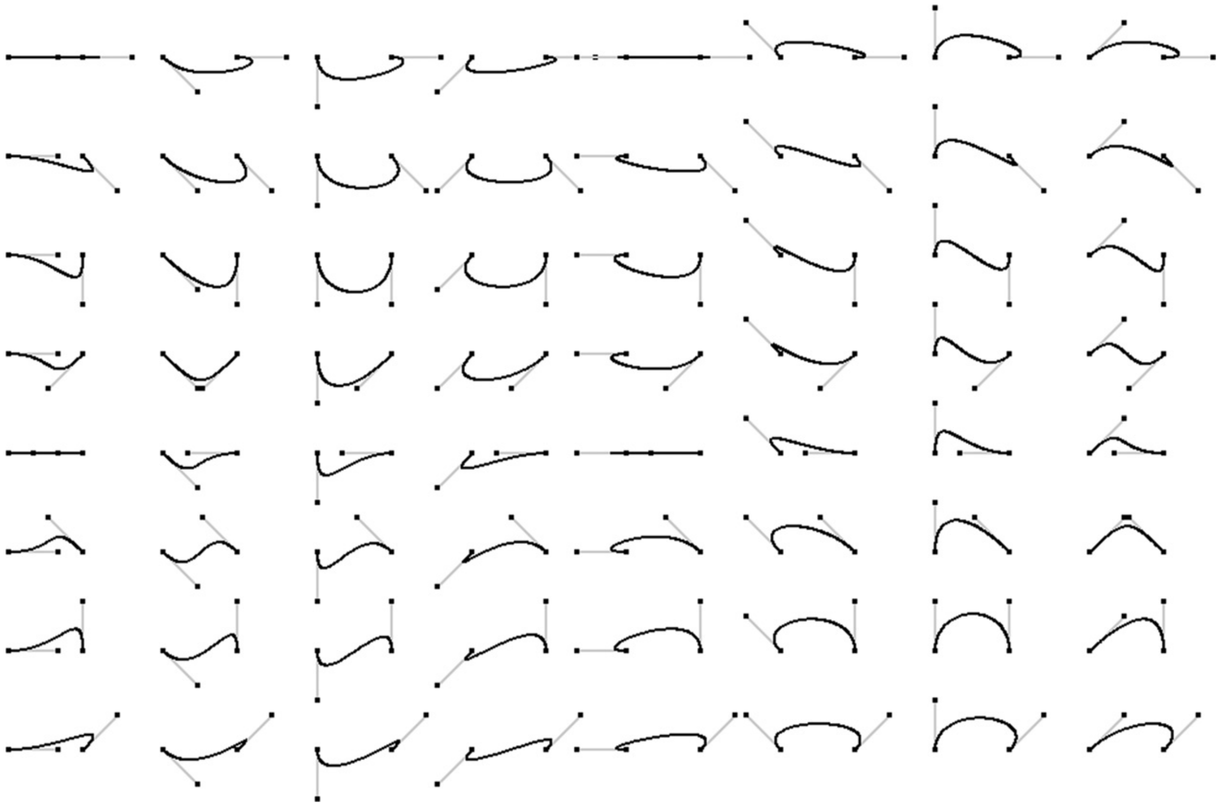

Fig. 3. Different types of non-interceptive cubic Bézier curve generated by rotating two intermediate control points.

following conditions is satisfied:

$$\mathrm{err}(d_i, d_{i+1}) + \mathrm{err}(d_{i+1}, d_{i+2}) > \mathrm{err}(d_i, d_{i+2}),$$
$$\mathrm{err}(d_i, d_{i+2}) < \varepsilon.$$

Rule 4: Introduce a dominant point using the following method if $\mathrm{err}(d_i, d_{i+1}) > \varepsilon$. From a set of chain codes between two break points, we get a modified set of chain code, after eliminating the co-linear points. While tracing from dominant point d_i to d_{i+1} in the modified set of chain code, identify a point (n) till which four $(p = 4)$ or more consecutive changes either in clockwise or anti-clockwise direction, are encountered for the second time. Then introduce a dominant point d_{new} at $(n - p)$th position where $0 \leqslant p \leqslant r$. Increase the value of p from its minimum value till we obtain $\mathrm{err}(d_i, d_{\text{new}}) < \varepsilon$.

The above method is intended to find the second local maximum curvature to introduce a dominant point. The proposed Rule application sequence to find the dominant points for Bézier approximation can be summarized as follows:

Step 1: Perform the contour tracking to find out the chain code of the curve and apply Rule 1.

Step 2: Extract the initial break points by applying Rule 2.

Step 3: Then apply Rules 3 and 4 repetitively to eliminate and introduce dominant points, respectively, wherever required.

Fig. 3 shows the different variation of Bézier curve generated by rotating each control point towards all eight possible orientations from the corresponding end points, while keeping the other internal control points at a fixed position. The absence of any dominant point is noticed for certain form of curve after applying the Rules 1 and 2 on a curve segment. These curves are mainly categorized in two distinct forms i.e. oscillating form and spiral form as shown in Figs. 4 and 5, respectively. Rules 1–3 are quite straightforward in contrast to Rule 4, which needs detailed explanation with elaboration as given in Examples 1 and 2.

Example 1. A complete 2π-rotation of the direction of progress in chain code is segmented out at alternative interception points on the curve by suitably chosen straight line. The similar segment is also obtained from the chain code of the curve by removing its co-linear points. The curve of oscillating form (Case I) is shown in Fig. 4(a). The typical chain code (0121076701…) of these types of curves after removing the co-linear points is depicted in Fig. 4(b). Four clockwise transitions (>), that corresponds to π-rotation of the direction of the progress in chain code is obtained from chain code $2 > 1 > 0 > 7 > 6$. An approximate cubic Bézier can accommodate only 2π-rotation (Fig. 4(b)) of the progress direction of its digital curve, to keep the ISE within a defined parametric limit. The segment between p_1 and p_n is the longest chain code segment, which can be approximated by cubic Bézier section as three roots of t corresponds to three interception points. Rule 2 will introduce a dominant point at p_{n+1}, though the Bézier curve can accommodate the segment from p_1 to p_n.

Example 2. Similarly the curve of spiral form (Case II) is shown in Fig. 5(a) and the chain code (2107654321…) of these

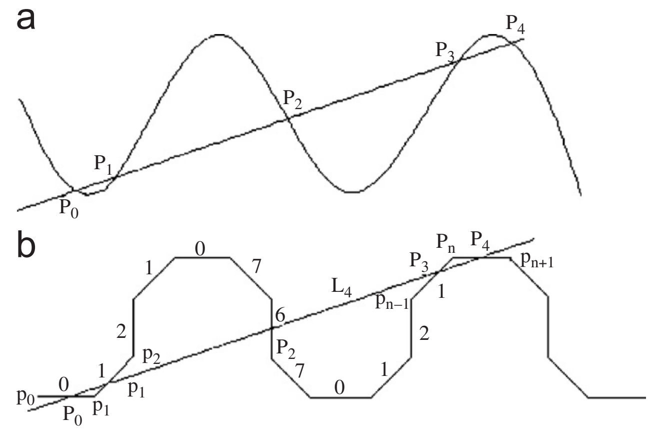

Fig. 4. Case I: (a) oscillating form of curve and (b) chain code representation after removing the co-llinear points.

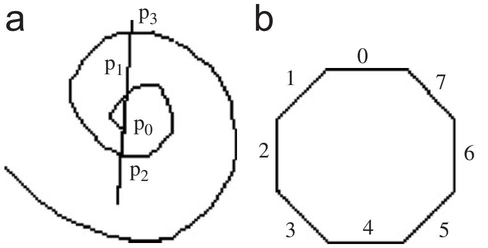

Fig. 5. Case II: (a) spiral form and (b) chain code representation after removing the co-llinear points.

types of curves after removing the co-linear points is depicted in Fig. 5(b). Four clockwise transitions, that corresponds to π-rotation of the direction of the progress of the curve is obtained from chain code $2>1>0>7>6$ as in Fig. 5(b). Similarly after $6>5>4>3$, a complete 2π-rotation is obtained and that corresponds to the p_2 point over the curve in Fig. 5(a).

4. Proposed segment representation method

After detecting the break points, the curve segments are approximated with Bézier curves. Segment representation with cubic Bézier curve is computationally very hard. The computational area (or grid) of size $M \times N$ is assumed to hold all 'n' points of the digital curve segment, the entire Bézier curve and the handles p_1 and p_2 for all cases (initial stage, intermediate processing stages and final optimal stage). The main objective is to find suitable and efficient search mechanism to find the positions of the handles p_1 and p_2 inside the computational area so that the drawn Bézier curve has least integral square error (ISE) value with respect to given 'n' sample points. This position finding computation has the complexity of $O(M^2N^2)$. If $M \approx N$, then the complexity becomes $O(M^4)$. The 625×10^8 number of computations have to be done to find the positions of the handles of the optimal Bézier curve, when $M = 500$. This computation is very much time consuming. To get the (near-) optimal solution quickly, an efficient convolution process of $O(1)$ complexity is applied and then TDLSA/ESA is used. The convolution process, after which $\frac{1}{4}$th and $\frac{3}{4}$th points on the curve are considered, does the initial approximation of the Bézier control points. These points are used for interpolating the actual Bézier control points.

4.1. *Method for Bézier control point interpolation*

The given interpolation method finds out two intermediate Bézier control points of the given curve. The starting and terminating points of the curve are known. It is required to know four points on a given curve with respective 't' to interpolate two intermediate control points. A Bézier curve is completely generated from its parametric form by varying t from 0 to 1. Therefore as t varies from 0 to $\frac{1}{4}$ and 0 to $\frac{3}{4}$, 25% pixel and 75% pixel of the curve is generated, respectively. Here we consider b_0 and b_3 as known end points (anchors) of a curve whose other two Béizer control points (handles) are to be estimated (Fig. 6). To compute other two points we find two more points located at the end of 25% and 75% of the full segment, respectively (Fig. 6). Thus the point p_1 and p_2 can be written in terms of four Bézier control points as follows:

$$p_1 = (27b_0 + 27b_1 + 9b_2 + b_3)/64,$$
$$p_2 = (b_0 + 9b_1 + 27b_2 + 27b_3)/64.$$

The values of b_1 and b_2 can be approximated by solving the De Casteljau parametric equations by taking $t = \frac{1}{4}$ and $\frac{3}{4}$, respectively,

$$b_1 = 8(3p_1 - p_2)/9 - (10b_0 - 3b_3)/9,$$
$$b_2 = 8(3p_2 - p_1)/9 + (3b_0 - 10b_3)/9.$$

This initial finding is very efficient because it takes $O(1)$ complexity. Still the ISE can be improved. Rosin [22] introduced fidelity (F) to measure how good a suboptimal solution ($E_{\min}$) is in respect to optimal solution (E)

$$F = \frac{E_{\min}}{E} * 100.$$

Assuming the initial solution as suboptimal solution and ESA based solution as the optimal one, fidelities of initial findings for the four parts of semicircle curve (see Fig. 8) are 32.8%, 51.2%, 10.6% and 51.5%. These poor fidelities suggest that further improvement is very important and is described in the next subsection.

4.2. *Methods for further improvement*

In this subsection, two improvement schemes are proposed. Both schemes are dependent on the initial finding. First method is TDLSA and another is ESA.

4.2.1. *Two-dimensional logarithmic search algorithm*

TDLSA is used to obtain control points of the best-fit Bézier curve [23]. Here the ISE for the five initial interpolated positions, one at the centre of the co-ordinate and four at co-ordinates ($\pm w/2$, $\pm w/2$) of the search window are computed

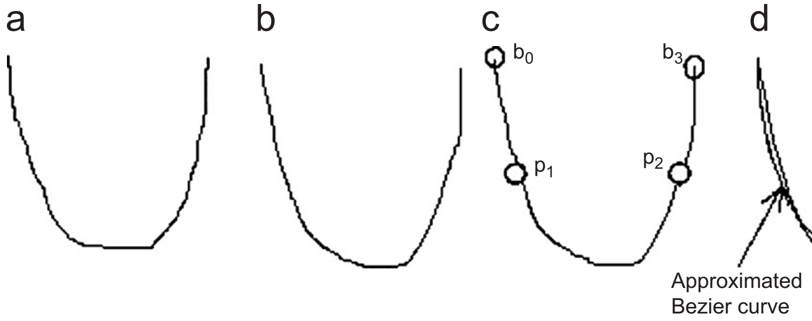

Fig. 6. (a) Face contour, (b) after smoothing operation, (c) four chosen points at $t = 0, \frac{1}{4}, \frac{3}{4}, 1$, (d) face contour and approximated Bézier curve.

first. In the next step, three more positions with the same step size in the direction of previous minimum ISE position are searched. The step size is then halved and above procedure is continued until the step size becomes unity.

4.2.2. *Evolutionary search algorithm*

In order to get near-optimal solution searching within very lesser search space and within a limited time span, ESA has been incorporated [24,25]. The various design parameters are addressed in this section. ESA must have the following components i.e. a mechanism to encode the solution as a binary string, population generation, crossover, mutation operator and a fitness function and selector.

4.2.2.1. Encoding A chromosome represents a feasible solution of the original problem. In the present context, the coordinates of handles p_1 and p_2 of cubic Bézier curve are taken as a chromosome. Bit encoding is employed and the length of the chromosome is kept fixed to $4 * \mathrm{Ceiling}(\log_2 M)$. For each Z_i, $Z = x, y$, $i = 2, 3$, $\mathrm{Ceiling}(\log_2 M)$ number of bits are needed.

4.2.2.2. Population generation Each generation 'n', there are three stages to form the valid chromosomes (see Fig. 7). In the first stage, one seed chromosome is considered. For the first generation, the algebraic solution, taking $n/4$ and $3 * n/4$th points on the smooth curve of the given digital curve segment as the two extra points needed for the solution is taken as seed. From the second generation onwards, the best chromosome found in the previous generation is taken. In the second stage, two windows of length 'W' around the positions of seed handles are considered and 'm' (even) chromosomes are randomly generated which lie within those windows. This is done to introduce some diversity into the population. The most difficulty factor here is the initial value of the windowing parameter 'W'. A poor choice may lead to several problems. A very high value of 'W' may lead to diverge from the solution and a very low value may require a large number of generations for the convergence to the near-optimal solution. Extensive experiments have been carried out to determine the initial value of 'W'. It is found that if initially the value of 'W' is kept at twice the square root of 'n' unit distance, then a very good start of search can be established. m more chromosomes are generated in the

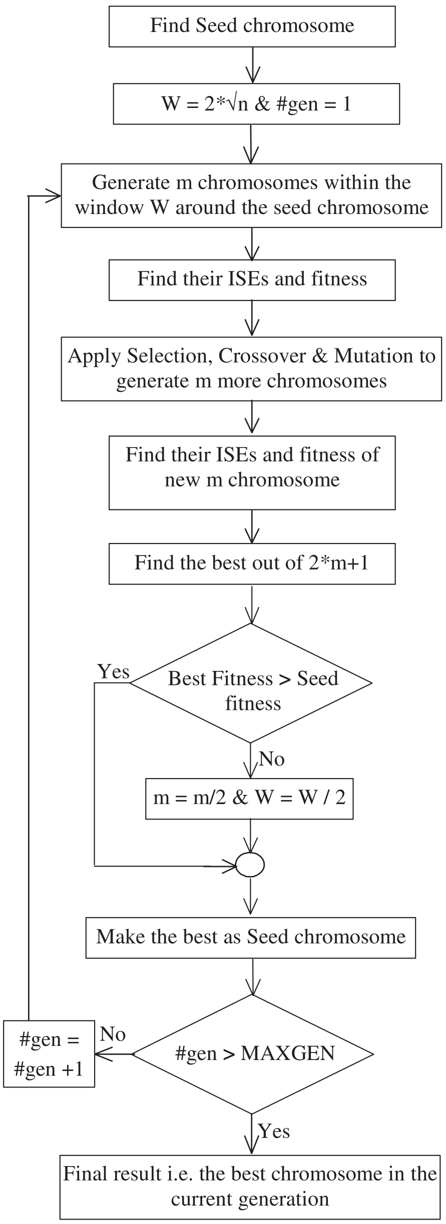
Find Seed chromosome
W = 2*√n & #gen = 1
Generate m chromosomes within the window W around the seed chromosome
Find their ISEs and fitness
Apply Selection, Crossover & Mutation to generate m more chromosomes
Find their ISEs and fitness of new m chromosome
Find the best out of 2*m+1
Best Fitness > Seed fitness
Yes
No
m = m/2 & W = W / 2
Make the best as Seed chromosome
#gen > MAXGEN
No
#gen = #gen +1
Yes
Final result i.e. the best chromosome in the current generation

Fig. 7. Structure of ESA for finding the most suitable positions of handles of cubic Bézier curve, targeting digital curve segment approximation.

last stage applying selection, crossover and mutation operator on the previously generated 'm' chromosomes. The best out of $1+m+m$ chromosomes is found out. If the present best is better than previous best or the present seed then the windows are shifted to the present best positions, keeping the size of window W fixed, otherwise W is decreased to $W/2$ keeping the positions of the windows unchanged. This is the rule for changing the windowing parameter 'W' in the successive generations. The minimum value W can attain is 1. The other difficulty factor is, choosing the value of 'm' (even). It will not be wise decision to make 'm' as a constant throughout all the generations. Decrement of the size of the windows implies the less chance of generating 'm' different chromosomes. So, after decrement of window size, there is a high chance of generating the same chromosome a number of times if 'm' is taken as a constant and this will increase the time for the searching process also. To get rid of these problems, the value 'm' is also decreased to '$m/2$' during the window size decrement. The minimum value for 'm' is 4. The initial value of 'm' is taken as 64 to get even values for 'm' throughout all the generations.

4.2.2.3. Fitness function and selection Fitness function should accurately measure the quality of the chromosome within the population. In this present context, a chromosome, which results in less ISE value, gets a high fitness. So, the fitness function should be a monotonically decreasing function of the ISE. Here, the fitness function can be defined as

$$F(c) = R - E(c),$$
$$R = \frac{(E_{\max} - E_{\min})}{(\text{Selectivity} - 1) + E_{\max}},$$

where $E(c)$ is the fitting error i.e. the ISE of chromosome 'c' and R is a constant. To calculate the suitable value of R, the method of 'selectivity' is proposed by Singh et al. [26] is adapted here. 'Selectivity' is the ratio of the fitness values of the maximally fit ($E_{\max}$) and minimally fit ($E_{\min}$) solutions.

Singh [26] have found a value of 3–5 for 'selectivity' to be optimal. In our problem R value is adaptive to keep the selectivity being a constant value of 4 in each generation. After the fitness values of all the chromosomes are scaled, a mating pool is established by the Universal Stochastic Selection scheme. Baker's stochastic universal sampling (SUS) algorithm [27], which is optimal in terms of bias and spread minimizes the stochastic errors associated with roulette-wheel selection.

4.2.2.4. Crossover It is employed to perform direct information exchange between individuals in a population. Therefore, the performance of ESA depends on the type of crossover operator implemented and a good choice of crossover rate. In the following p_1 and p_2 refer to two parents and c_1, c_2 two children. $p_1 \Theta p_2 = c_1, c_2$. We adopted m-point i.e. multi-point type operator. This operator is effected by randomly dividing each parent p_i into $(m+1)$ sections, $m \geqslant 1$. Starting with the first parent chromosome p_1, alternate sections from the two parents are copied to form the first child chromosome c_1. Similarly starting with p_2, c_2 can be generated. Consider the following example with $m = 2$:

$$p_1 = [1\,0\,0|1\,0\,1|1\,1\,1],$$
$$p_2 = [1\,1\,0|0\,0\,0|0\,0\,1]$$

then

$$c_1 = [1\,0\,0|0\,0\,0|1\,1\,1],$$
$$c_2 = [1\,1\,0|1\,0\,1|0\,0\,1].$$

The crossover operation is taken with a high probability of 0.95. A high value is taken to restrict the move of parent chromosomes into next generation without undergoing any genetic change. This is because the seed, m parent chromosomes, which are generated from the seed and m child chromosomes generated from parents are considered at a time for finding the best chromosome of current generation.

4.2.2.5. Mutation To avoid the local optima problem, mutation operator is used with very less probability. In this problem, mutation probably is kept at 0.02. Through bit inverting i.e. changing 0 to 1 and 1 to 0, mutation process has been carried out.

5. Results

In order to get into a deeper sense about the performance of the present method, we have applied TDLSA and ESA to the four digital curves, namely, a chromosome-shaped curve, an infinity-shaped curve, a leaf-shaped curve, and a curve with four semicircles. Detailed analysis of ESA on the semicircle curve is described here. This semicircle is segmented into four semicircular arcs after application of proposed break point detection method where TDLSA are used to find error of fitting. Each segment is treated as the input for the evolutionary technique. The segments are semicircle1 (point number 1–51), semicircle2 (point number 51–65), semicirle3 (point number 65–89) and semicircle4 (point number 89–102) as shown in Fig. 8. The decrements of ISE values with the generation are presented in the graphical form in Fig. 9 for all these four semicircles. A horizontal line between generation number i and $i+1$ in these plot indicates the change of W value for the $(i+2)$th generation. Near-optimal values are obtained in fourth generation for all the four segments of the semicircle curve.

Here R value is adaptive to keep the selectivity value constant (i.e. 4) in each generation. The crossover operation is taken with a probability of 0.95. These parameters play an important role in the convergence. The maximum generation value (MAXGEN) is kept at 10. The ISE values for the full digital curves using the ESA and TDLSA are presented in the Table 1. For all four test curves, ESA outperforms the other method. The ESA, coded in VB, was run on a 550 MHz Pentium-III PC under Windows operating system. Without any code optimization, the converging time (including disk access) of the proposed method obtained, corresponding to Fig. 9(a), (b) and (c), are 3 min 21 s, 2 min 58 s and 1 min 19 s, respectively.

The ISE values obtained for the four digital curves approximation using the TDLSA and ESA are presented in the

Table 1. The logarithmic search is a semi-extensive search method that searches for the optimal control points by hopping from one region to another, in the direction of lesser ISE. But, ESA reduces if the best solution inside the current population is the current seed.

The initial window size (W) in TDLSA and ESA is taken considering that high value of W may diverge the solution and low may increase the iterations to a very high number. In TDLSA, the ISE for the five initial positions, one at the centre of the interpolated co-ordinate and four at co-ordinates $(\pm W/2, \pm W/2)$ of the search window i.e. for 25 combinations are computed first. In the next step, three more positions with the same step size in the direction of previous minimum i.e. at nine locations minimum ISE is searched. The step size is then halved every time and above procedure is continued at the minimum ISE position obtained from 33 combinations until the step size becomes unity. For ESA, $m = 64$ (initially) and MAXGEN = 10 and m should be less than 81. In ESA, 80 points have been considered within the pool for generation of the child chromosome. ESA reduces if the best solution inside the current population is the current seed. Moreover, there are several conjugate pairs of control points for which integral error magnitudes are comparable. Those points are located in such distributed ways that raise several numbers of local optima which cannot be avoided in the medium window size during TDLSA by 5 points comparison.

The results of TDLSA and ESA are presented and compared with other approximation techniques in Tables 2 and 3 respectively. In order to assess the performance of the proposed,

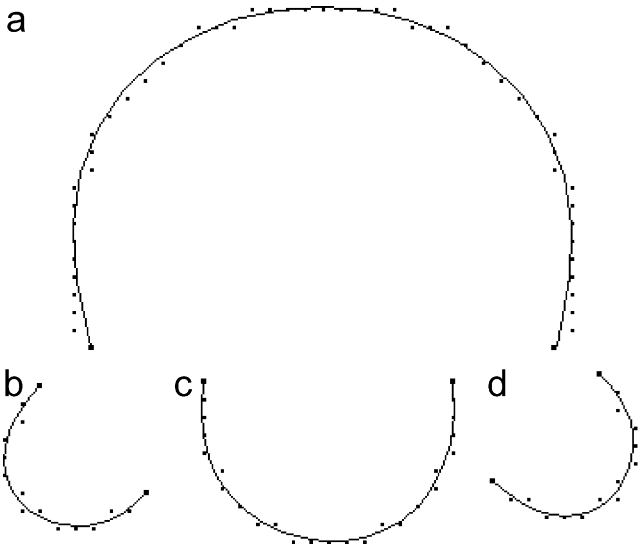

Fig. 8. Cubic Bézier approximation for four digital semicircle segments of semicircle curve with ESA: (a) approximation for semicircle1; (b) approximation for semicircle2; (c) approximation for semicircle3 and (d) approximation for semicircle4.

Table 1
ISEs of the well-known digital curves approximation using TDLSA and ESA

Curve name	Algorithm	ISE
Chromosome	2-D Logarithmic Search	3.68
	Evolutionary Search	2.22
Figure of eight	2-D Logarithmic Search	3.69
	Evolutionary Search	3.01
Semicircle	2-D Logarithmic Search	6.93
	Evolutionary Search	6.44
Leaf	2-D Logarithmic Search	4.04
	Evolutionary Search	2.34

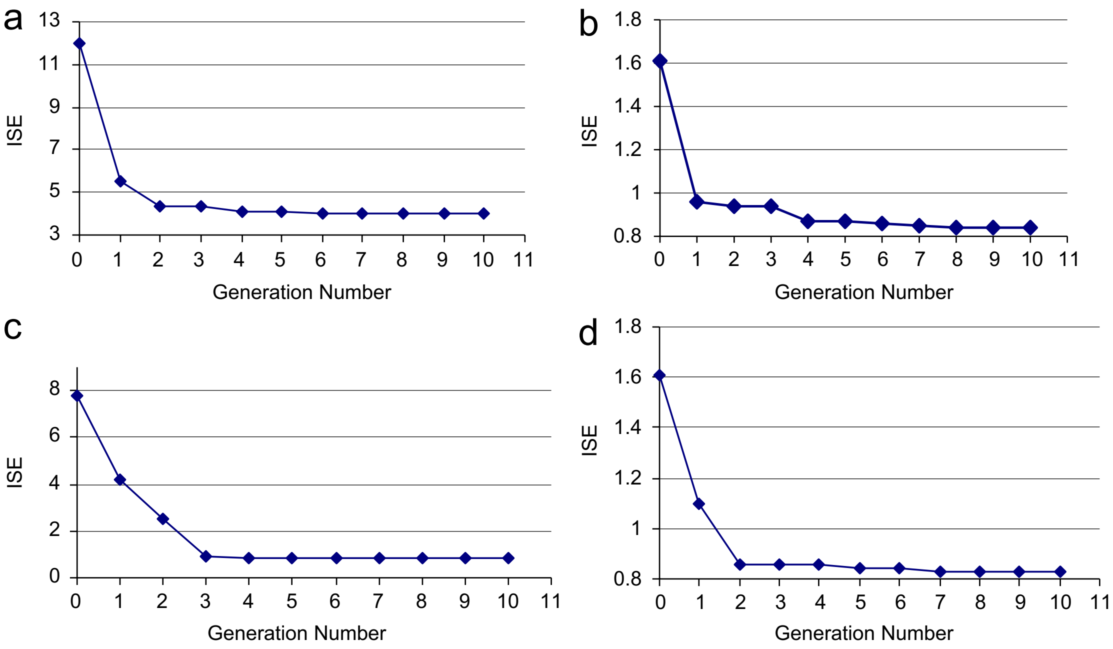

Fig. 9. ISE vs. generation number plot for the four digital semicircle segments of semicircle curve: (a) plot for semicircle1; (b) plot for semicircle2; (c) plot for semicircle3; and (d) plot for semicircle4.

Table 2
Results of the TDLSA and of six other methods of polygonal approximation

Curve	Method	n_d	n_{bc}	CR	E_2	E_∞	WE_2	WE_∞	FOM
Chromosome ($n=60$)	Teh and Chin [2]	15	—	4.00	7.20	0.71	1.88	0.18	0.555
	Ray and Ray [3]	18	—	3.33	5.57	0.71	1.67	0.21	0.599
	Cornic [4]	17	—	3.53	5.54	0.86	1.57	0.24	0.637
	Sarkar (1-point) [5]	16	—	3.75	5.100	0.707	1.36	0.188	0.735
	Sarkar (2-point) [5]	19	—	3.16	3.857	0.555	1.22	0.175	0.82
	Wen-Yen Wu [28]	17	—	3.53	5.01	0.64	1.42	0.18	0.798
	Proposed TDLSA	5	10	4.00	3.68	0.602	0.92	0.15	1.08
Infinity ($n=45$)	Teh and Chin	13	—	3.46	5.93	1.0	1.71	0.29	0.59
	Ray and Ray	12	—	3.75	5.989	0.894	1.60	0.24	0.626
	Cornic	10	—	4.55	4.30	0.78	0.96	0.17	1.046
	Sarkar (1-point)	12	—	3.75	5.91	0.707	1.576	0.188	0.634
	Sarkar (2-point)	14	—	3.214	3.67	0.632	1.142	0.197	0.876
	Wen-Yen Wu	13	—	3.46	5.17	1.11	1.49	0.32	0.669
	Proposed TDLSA	2	6	5.625	3.69	0.694	0.656	0.123	1.524
Leaf ($n=120$)	Teh and Chin	29	—	4.14	14.96	0.99	3.62	0.24	0.276
	Ray and Ray	32	—	3.76	14.72	1.00	3.93	0.27	0.255
	Cornic	a	—	A	a	a	a	a	a
	Sarkar (1-point)	23	—	5.22	13.17	0.784	2.523	0.15	0.396
	Sarkar (2-point)	23	—	5.22	13.17	0.784	2.523	0.15	0.396
	Wen-Yen Wu	23	—	5.22	20.34	1.00	3.90	0.19	0.256
	Proposed TDLSA	16	32	2.50	4.04	0.67	1.62	0.27	0.619
Semicircles ($n=102$)	Teh and Chin	22	—	4.64	20.61	1.00	4.45	0.22	0.225
	Ray and Ray	29	—	3.52	11.82	0.83	3.36	0.24	0.297
	Cornic	30	—	3.40	9.19	0.88	2.70	0.26	0.369
	Sarkar (1-point)	19	—	5.368	17.377	1.474	3.237	0.274	0.309
	Sarkar (2-point)	20	—	5.1	13.65	1.052	2.676	0.206	0.373
	Wen-Yen Wu	27	—	3.78	9.01	0.83	2.38	0.22	0.419
	Proposed TDLSA	4	8	8.5	6.935	0.688	0.815	0.08	1.225

Table 3
Result of the TDLSA and ESA with four other methods of curve approximation

Method	Chromosome ($n=60$)			Infinity ($n=45$)			Leaf ($n=120$)			Semicircle ($n=102$)		
	n_b	ISE	FOM_c	n_b	ISE	FOM_c	n_b	ISE	FOM_c	n_b	ISE	FOM_c
Pei and Horng [8]	15	6.18	0.65	9	4.24	1.18	31	19.66	0.20	12	10.90	0.92
Pei and Horng [9]	10	2.99	2.01	8	2.51	2.24	18	8.25	0.81	4	6.94	4.32
Horng and Li [11]	10	2.67	2.25	6	3.06	2.45	16	11.31	0.66	4	6.94	4.32
Sarkar et al. [12]	10	2.67	2.25	6	3.26	2.30	16	10.96	0.68	4	6.94	4.32
	11	2.60	2.10	8	2.36	2.38	18	7.40	0.90	6	6.46	3.10
	15	2.18	1.83	9	2.03	2.46	31	1.64	2.36	12	4.31	2.32
Proposed TDLSA	5	3.68	3.26	2	3.69	6.10	16	4.04	1.86	4	6.93	4.33
Proposed ESA	5	2.22	5.41	2	3.01	7.48	16	2.34	3.21	4	6.44	4.66

methods, seven performance evaluation criteria were used in the experiment. They are the number of dominant points or break points, the compression ratio, the sum of square error, the maximum error, the weighted sum of square error, the weighted maximum error and the figure of merit.

Number of dominant points (n_d): It is important to have the consistent number of dominant points for the curves representing the same object with different scales and orientation. A robust method of shape representation should be independent of scale and orientation.

Number of Bézier control points (n_{bc}): It represents the total number of Bézier control points required to represent the object by Bézier curve segments.

Compression ratio (CR): One of the objectives of dominant point detection is data compression. The larger the compression ratio, more effective the data compression is. For the same curve a smaller number of dominant points results in larger compression ratio. The compression ratio can be defined as

$$\mathrm{CR} = \frac{n}{(n_d + n_{bc})},$$

$$E_\infty = \sum_{i=1}^{n} \max\{e_i\}.$$

Weighted sum of square error (WE_2): The weighted sum of square error is to combine the compression ratio and the sum

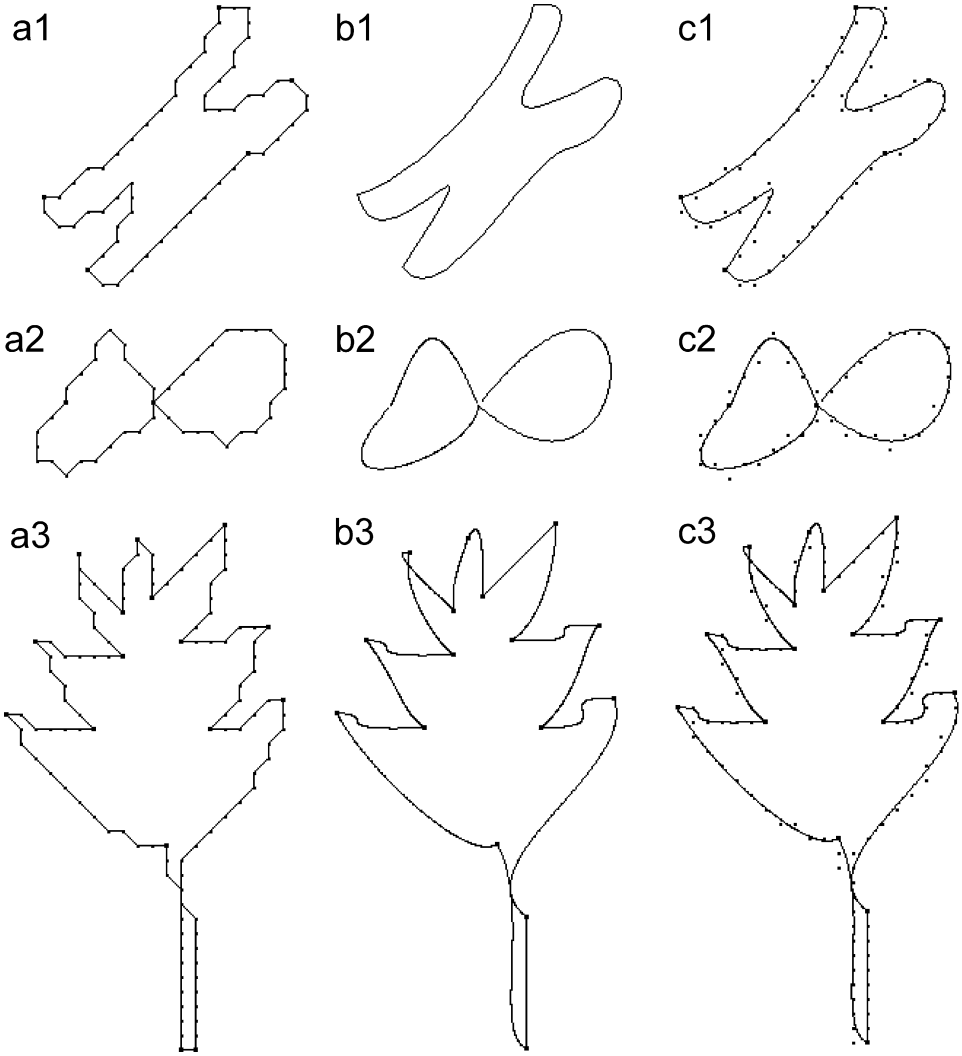

Fig. 10. (a1) Chromosome, (a2) infinity, (a3) leaf, (b1, b2, b3) Bézier curve approximation of a1, a2, a3, respectively, (c1, c2, c3) overlap of b1, b2, b3 with a1, a2, a3, respectively.

of square error. It is defined as

$$\mathrm{WE}_2 = \frac{1}{\mathrm{CR}} E_2.$$

Weighted maximum error (WE_∞): The weighted maximum error is to combine the compression ratio and the maximum error. It is defined as

$$\mathrm{WE}_\infty = \frac{1}{\mathrm{CR}} E_\infty.$$

Figure of merit (FOM): It is a measure proposed by Sarkar [5] and combines the number of dominant points with the ISE. It is defined as

$$\mathrm{FOM} = \frac{n}{(n_\mathrm{d} \times E_2)}.$$

Break point detection method where TDLSA are used to find error of fitting is applied to four chain-coded curves shown in Fig. 10(a1)–(a3). The chain codes of all these curves are found in Teh and Chin [2]. Table 2 is prepared to compare the performance of TDLS algorithm with the few of other polygonal approximation algorithms. The Bézier approximations of the curves are plotted in Fig. 10(b1)–(b3). Fig. 10(c1)–(c3) show the overlapping of the Bézier curves with the original chain-coded curves.

For the comparative study we have used Sarkar's FOM using the four well-known shapes of Chromosome, Infinity, Leaf and Semicircle. It is quite natural and expected to get a better result as we have compared polygonal approximation algorithm with cubic curve approximation algorithm in Table 2. So in Table 3, we have compared the performance of TDLSA and ESA with other curve approximation algorithms. Two algorithms by Pei and Horng [8,9] use only circular arcs for approximation. The optimal algorithms developed by Horng and Li [11] and Sarkar et al. [12] use line segment and circular arc. The figure of merit ($\mathrm{FOM_c}$) is defined as $n/n_\mathrm{b} \times \mathrm{ISE}$ for curve approximation of digital curve where n is total number of point and n_b is defined as total number of break point which is one less than the total number of curve segments to be approximated. A significant improvement of $\mathrm{FOM_c}$ is observed for chromosome and infinity.

6. Conclusions

In this paper, we propose a method for automatic detection of break points, then approximating the digitized curve by fitting with Bézier curve. The advantage of Bézier is that it has relatively less number of control points. This results in compression of edge boundary as well as smoothing of non-smooth edge contours. The Bézier curve is chosen because it directly goes through the specified initial point and terminating points, where as B-spline goes near to the starting point and the terminating point. Smooth edges have major advantages for natural representation. This increases the quality of the semi-synthetic images and human perception of shapes which deemed to be based on curvatures of parts of contour [13].

Approximation of the digital curves by cubic Bézier segment makes fitting more perfect than other reported curve approximation methods. The computation expense to fit the Bézier curve is reduced by interpolating two intermediate control points by simple algebraic interpolation technique. The refinement at the location of the two control points, are done by TDLSA and ESA. The use of a search algorithm enhances the fitting process, by reducing the ISF. The proposed algorithms have been tested on several well-known chain-coded curves and appreciable results have been shown in Tables 2 and 3. It is evident from Tables 2 and 3, the proposed algorithm yielded much lesser number of break points to represent most of the shapes than any other methods given. Also, the ISE is the least for all curves. The figures of merit for all the shapes are quite significantly better than any other algorithm. A much better result is obtained so far as all error measures are concerned.

The experimental results show that the proposed method of representation by Bézier curve can approximate the curves with minimum approximation errors. The numbers of break points are also compared to other methods of approximation. Pal et al. [29] has used Bézier curve based facial identification system where non-interceptive Bézier curves are used to outline the facial features (i.e. hair boundary, eye, eye-brow, nose, lips and face boundary). This has reduced the computational cost, as only four points are sufficient to represent most of the individual segment of facial features with various complex expressions. Our next approach is to develop a complex lip template, to model lip contours accurately for symmetric and non-symmetric expression analysis.

Acknowledgements

We gratefully acknowledge the valuable suggestion received from Dr. Debranjan Sarkar for improving the quality of the paper. The constant encouragement received from Mr. Subrata Dasgupta is thankfully acknowledged.

References

[1] E. Attneave, Some informational aspects of visual perception, Psychol. Rev. 61 (1954) 183–193.

[2] C.H. Teh, R.T. Chin, On the detection of dominant points on digital curves, IEEE Pattern Anal. Mach. Intell. 11 (1989) 859–872.

[3] B.K. Ray, K.S. Ray, Detection of significant points and polygonal approximation of digitized curves, Pattern recognition Lett. 13 (1992) 443–452.

[4] P. Cornic, Another look at the dominant point detection on digital curves, Pattern Recognition Lett. 173 (1997) 13–25.

[5] D. Sarkar, A simple algorithm for detection of significant vertices for polygonal approximation of chain coded curves, Pattern Recognition Lett. 14 (1993) 959–964.

[6] M. Marji, P. Siy, A new algorithm for dominant points detection and polygonalization of digital curves, Pattern Recognition 36 (2003) 2239–2251.

[7] P.Y. Yin, A tabu search approach to polygonal approximation of digital curves, Int. J. Pattern Recognition Artif. Intell. 14 (2000) 243–255.

[8] S.-C. Pei, J.-H. Horng, Fitting digital curves using circular arcs, Pattern Recognition 28 (1995) 107–116.

[9] S.-C. Pei, J.-H. Horng, Optimum approximation of digital planar curves using circular arcs, Pattern Recognition 29 (1996) 383–388.

[10] P.L. Rosin, G.A.W. West, Segmentation of edges into lines and arcs, Image Vision Comput. 7 (1989) 109–114.

[11] J.H. Horng, J.T. Li, A dynamic programming approach for fitting digital planer curves with line segments and circular arcs, Pattern Recognition Lett. 22 (2001) 183–197.

[12] B. Sarkar, L.K. Singh, D. Sarkar, Approximation of digital curves with line segments and circular arcs using genetic algorithms, Pattern Recognition Lett. 24 (2003) 2585–2595.

[13] D. Hoffman, The interpretation of visual illusions, Sci. Am. 249 (6) (1983) 154–162.

[14] L. Cinque, S. Levialdi, A. Malizia, Shape description using cubic polynomial Bézier curves, Pattern Recognition letters. 19 (1998) 821–828.

[15] A. Rosenfeld, C. Kak, Digital picture processing, Comput. Sci. Appl. Math. 2 (1982) 220–223.

[16] M.A. Bhuiyan, H. Hama, Identification of actors drawn in Ukiyoe pictures, Pattern Recognition 35 (2002) 93–102.

[17] H. Freeman, On the encoding of arbitrary geometric configurations, IRE Trans. Electron. Comput. 10 (1961) 260–268.

[18] H. Freeman, A scheme for the efficient encoding of graphical data for communication and information processing, in: Advances in Electronics, Proceedings of the 16th Electronics Congress, Rome, 1969, pp. 340–348.

[19] H. Freeman, Computer processing of line-drawing images, Computer Surv. 6 (1974) 57–97.

[20] J.E. Bézier, Numerical Control: Mathematics and Applications, Wiley, Chichester, UK, 1972.

[21] A. Watt, M. Watt, Advanced Animation and Rendering Techniques Theory and Practice, Addison-Wesley, Reading, MA, 1992, pp. 65–108.

[22] P.L. Rosin, Techniques for assessing polygonal approximations of curves, IEEE Pattern Anal. Mach. Intell. 19 (1997) 659–666.

[23] J.R. Jain, A.K. Jain, Displacement measurement and its application in interframe image coding, IEEE Trans. Commun. Dec (1981) 1799–1808.

[24] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[25] C.A. Coello Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, Dordrecht, 2002.

[26] M. Singh, A. Chatterjee, S. Chaudhury, Matching structural shape descriptions using genetic algorithms, Pattern Recognition 30 (1997) 1451–1462.

[27] J.E. Baker, Reducing bias and inefficiency in the selection algorithm, in: Proceedings of the Second International Conference on Genetic Algorithms (ICGA), 1987, pp. 14–21.

[28] W.-Y. Wu, An adaptive method for detecting dominant points, Pattern Recognition 36 (2003) 2231–2237.

[29] S. Pal, P.K. Biswas, A. Abraham, in: Face Recognition Using Interpolated Bézier Curve Based Representation, International Conference on Information Technology: Coding and Computing, vol. 1, April 5–7, 2004.

About the Author—SARBAJIT PAL received a B.E. degree in Electronics and Communication Engineering from R.E.C., Trichy in 1986 and M.Tech. in Computer Engineering from I.I.T. Kharagpur in 1998. He worked as a Scientific Officer in Bhaba Atomic Research Centre, Mumbai, India from 1987 to 1992. He is presently working as a Scientific Officer in Beam Diagnostics and Computer Control Division in Variable Energy Cyclotron Centre, Kolkata, India. His research interests are Image Processing and Pattern Recognition.

About the Author—PANKAJ GANGULY received the B.Tech. degree in Computer Science and Technology in 2003 from Kalyani University and the M.Tech. degree in Information Technology in 2005 from ABV-Indian Institute of Information Technology and Management, Gwalior, India. Currently, he is working with the Persistent System Pvt. Limited, Nagpur, India. His main research interests are in the field of Soft Computing, Autonomic Computing, Curve Analysis and Shape Representation.

About the Author—Dr. P.K. BISWAS completed his B.Tech. (Hons), M.Tech. and Ph.D. from the Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, India in the years 1985, 1989 and 1991, respectively. From 1985 to 1987 he was with Bharat Electronics Ltd., Ghaziabad as a Deputy Engineer. Since 1991 he has been working as a Faculty member in the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, where he is presently holding the position of Professor. Dr. Biswas visited University of Kaiserslautern, Germany under Alexander von Humboldt Research Fellowship during March 2002 to February 2003. Dr. Biswas has more than 60 research publications in International and National Journals and Conferences and has filed seven international patents. His areas of interests are Image Processing, Pattern Recognition, Computer Vision, Video Compression, Parallel and Distributed Processing and Computer Networks.