A Bezier Curve-Based Generic Shape Encoder

Ferdous A. Sohel[1*], Gour C. Karmakar[2], Laurence S. Dooley[3], and Mohammed Bennamoun[1]

[1]School of Computer Science and Software Engineering, The University of Western Australia, WA 6009, Australia

[2]Gippsland School of Information Technology, Monash University, Churchill, Victoria 3842, Australia

[3] Department of Communication and Systems, The Open University, Milton Keynes, MK7 6AA, United Kingdom

Abstract

Existing Bezier curve based shape description techniques primarily focus upon determining a set of pertinent *Control Points* (CP) to represent a particular shape contour. While many different approaches have been proposed, none adequately consider domain specific information about the shape contour like its gradualness and sharpness, in the CP generation process which can potentially result in large distortions in the object's shape representation. This paper introduces a novel *Bezier Curve-based Generic Shape Encoder* (BCGSE) that partitions an object contour into contiguous segments based upon its cornerity, before generating the CP for each segment using relevant shape curvature information. In addition, while CP encoding has generally been ignored, BCGSE embeds an efficient vertex-based encoding strategy exploiting the latent equidistance between consecutive CP. A nonlinear optimisation technique is also presented to enable the encoder is automatically adapt to bit-rate constraints. The performance of the BCGSE framework has been rigorously tested on a variety of diverse arbitrary shapes from both a distortion and requisite bit-rate perspective, with qualitative and quantitative results corroborating its superiority over existing shape descriptors.

Keywords: Bezier curve, shape coding, bit-rate, nonlinear optimisation, distortion.

1. Introduction

* Corresponding author – E-mail: Ferdous.Sohel@csse.uwa.edu.au, Ferdous.Sohel@ieee.org, Tel.: +61 8 6488 2796, Fax: +61 8 6488 1089. Postal address: CSSE, UWA, 35 Stirling HWY, Crawley, WA 6009, Australia.
Other authors' e-mails: Gour.karmakar@infotech.monash.edu.au, l.s.dooley@open.ac.uk, and m.bennamoun@csse.uwa.edu.au

Bezier Curves (BC) were independently developed by P. de Casteljau and P. E. Bézier, and while their origin can be traced to the design of car body shapes in the automotive industry, their contemporary applications encompass many diverse disciplines. In particular, their robustness in curve and surface representation means they pervade many fields of multimedia technology including shape description of characters [1], [2] and objects [3], active shape lip modelling [4], shape error concealment for MPEG-4 objects [5] and surface mapping [6]. The classical BC is defined by a set of *control points* (CP) with the number and orientation of these points governing the overall size and shape of the curve. In shape encoding applications, the distance between a shape contour and the approximating curve (distortion) crucially depends upon the generated CP, so efficacious CP computation is vital for any BC-based shape coder. Furthermore, the use of a single BC to represent a complex shape is computationally very expensive as a high order BC will be mandated, with the corresponding CP calculations also incurring a substantial computational cost. To reduce this overhead, composite BC [7] have been used to represent more complex shapes, whereby the entire shape is sub-divided into segments, with each individually represented by a BC.

Cinque et al. [3] introduced a *Shape Description using Cubic polynomial Bezier curves* (SDCB) technique which divides the shape-boundary into an *a priori* number of segments, each comprising the same number of boundary points, with the CP evenly distributed over the entire boundary irrespective of its complexity. As both the segment division and CP generation processes consider only the number of shape points, this approach is independent of boundary complexity which can lead to large distortions, even when a high number of relatively short segments are used. Moreover, as the segments are of equal length in terms of boundary points, the strategy does not consider aspects of a shape such as its *cornerity* and *branches*. Composite BC (CBC) has successfully been applied in the *Automatic outline Capture of Arabic Fonts* (ACAF) [1] and *Shape Description for Chinese calligraphy Characters* (SDCC) [2] algorithms to respectively describe Arabic and Chinese character outlines, as well as in *Active Shape Lip Modelling* (ASLM) [4] applications to represent different lip formations. In all these algorithms however, while the shape is divided into segments by considering the cornerity at the shape points, more localised shape information within each segment is not subsequently considered during CP generation. The two end-points of each segment are respectively chosen as the start and end CP, with intermediate points calculated in a variety of ways. For instance in both SDCC and ASLM, the intermediate CP locations are derived from the two tangents at the segment endpoints using either the intersection of the tangents [2] or a computationally intensive trial-and-error approach [4]. The CP on the tangents at the segment ends however, does not guarantee efficient shape approximation

as they are not necessarily representative of the entire segment. In contrast, ACAF [1] adopts a distortion minimisation approach, though this is computationally expensive and usually dictates a further subdivision of a segment in order to obtain low distortion.

Within the MPEG-4 standardisation process, the vertex-based *Operational-Rate-Distortion* (ORD) optimal shape coding framework has been developed using parametric B-splines [8] and [9], which has been further extended in [10]. While these techniques provide *rate-distortion* (RD) optimisation flexibility, they as evidenced in Section 4, are also computationally very expensive due to using an exhaustive shortest-path search method in a directed-acyclic-graph containing a large number of vertices.

One recurring feature in all the aforementioned algorithms is that the CP may not of necessity reside on the shape contour, a consequence of which is an increase in the descriptor length. As the CP describe a shape, their efficient encoding will considerably aid in reducing the descriptor length and thereby the resulting communications cost. While SDCB [3] adopts a parametric approach to CP encoding, the techniques delineated in ACAF, SDCC and ASLM, conspicuously do not comment upon their respective CP encoding strategy. The parametric descriptor used in SDCB generates four CP for each segment and specifically comprises: *i*) the absolute coordinates of the first and fourth CP, *ii*) the directional angle and magnitude distance of the second CP from the first CP, and *iii*) the directional angle and magnitude distance of the third CP from the fourth CP. Both the angle and distance parameters are encoded using floating point notation which renders this approach unsuitable for low-bit rate video applications such as, video streaming over the Internet and mobile video transmission for hand-held devices, where innate bandwidth limitations demand highly efficient bit minimisation techniques.

This paper presents a novel *Bezier Curve-based Generic Shape Encoder*[†] (BCGSE) that reduces both the distortion and shape descriptor length by applying an efficient CP generation strategy incorporating shape information, concomitant with an improved vertex coding scheme. Unlike [1], [2] and [4], CP generation takes cognisance of the cornerity of a contour in segmenting a shape boundary, with localised information about the curvature of the segment being employed to obtain the CP. In the ensuing coding phase, an innovative extension to the *Object-Adaptive Vertex Encoding* (OAVE) technique [13] is proposed. The original OAVE encodes a set of vertices by adapting the representation to the dynamic range of the relative locations of the vertices before utilising

[†] Preliminary ideas involved in this research were introduced in [11] and [12]

an octant-based representation for each individual vertex. The new enhancement E-OAVE, entails exploiting the inherent cyclic relationship between consecutive CP to further improve the coding efficiency of OAVE, with the corollary of combining the new CP generation scheme with better bit-rate coding being superior RD performance. In addition, to adaptively accommodate bit-rate constraints for BCGSE, a nonlinear optimisation has been utilised to sustain admissible bit-rates. The performance of the BCGSE framework has been extensively tested and analysed on a variety of different object shapes, with both quantitative and qualitative results consistently confirming its superiority compared with existing shape descriptor methods.

The remainder of this paper is organised as follows: Section 2 provides a short overview of Bezier curve theory, while Section 3 presents the new BCGSE shape descriptor framework including the new CP generation and efficient vertex coding strategies. Experimental results are analysed in Section 4 to endorse the improved RD performance of the BCGSE model, while some general conclusions are provided in Section 5.

2. The Classical Bezier Curves

The Bernstein form of an n^{th} order BC with a CP set $V_j = \{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}$ is defined by:

$$v_j(t) = \sum_{k=0}^{n} v_{j,k} B_k^n(t) \qquad ,0 \le t \le 1 \tag{1}$$

where $B_k^n(t) = \binom{n}{k}(1-t)^{n-k} t^k$ are the Bernstein polynomials with $\binom{n}{k}$ being the *combination* function, t is the weight of subdivision which determines the number of points on the BC, and subscript j represents the j^{th} curve segment within the series of CBC that defines the shape.

The attraction of BC being a member of a family of parametric curves, is that even with a few CP at the encoder, an arbitrary number of curve points can be generated at the decoder by controlling the steps in t, with the greater the number of CP on a curve leading to a smoother reconstruction. This unique parametric curve set includes both Hermite curves and splines, with BC chosen as the shape descriptor because it is computationally efficient and straightforward to calculate and unlike B-splines, no coordination is required at the segment end-points. Another cogent reason for choosing BC curves is they are affine invariant [14], so can be effectively used in searching any affine transformed shape such as for example, in multimedia retrieval applications. In this paper, cubic BC are used

to define each shape segment, as lower order curves such as quadratic BC are less smooth. Whilst higher degree curves are preferable for shape approximation, they require more CP to be calculated and as a consequence the curve generation overhead is computationally greater. Moreover, encoding large numbers of CP inevitably mandates a higher bit-rate. The functional form of the cubic BC is represented as:

$$v_j(t) = (1-t)^3 v_{j,0} + 3t(1-t)^2 v_{j,1} + 3t^2(1-t)v_{j,2} + t^3 v_{j,3} \quad (2)$$

3. Shape Encoding Strategy

The BCGSE framework comprises two constituent components, namely the calculation of the CP for each segment, and secondly an efficient CP encoding strategy based on a new extension to the OAVE algorithm [13]. A set of strategies have also been proposed to accommodate the stringent bit-rate limits. These will now be respectively considered.

Determining the Control Points

In the first step of CP determination, the starting point of the contour needs to be determined, with in this paper, the highest curvature point being selected as this point. There are two main reasons for this choice: *i)* it is highly probable the highest curvature point will eventually be a segment end [15], and *ii)* such a starting point affords the desirable characteristic of being affine invariant, which is not always the case in other popular approaches, such as the first point on a raster-scan basis (from left to right, top to bottom).

The next step in the CP generation involves sub-dividing a shape into segments at its corner points, where the number of segments may be either defined *a priori* or dynamically. The *cornerity* of a boundary point is characterised by the maximum arc-chord deflection at that point, so any point possessing a local cornerity maximum is in fact a corner [16]. Different approaches for detecting the corner points of a shape has been analysed and compared in [17], with the generic conclusion that the *Beus-Tiu algorithm* [18] provides the best perceptual results, and as a consequence this technique is employed in the new BCGSE shape descriptor framework. CP are selected from a set of points obtained from the boundary points and curvature-related shape information concerning the boundary, with the concept of *significant* and *supplementary* points being introduced so contour portions with more rapidly changing shape features like sharp edges and corners, are given greater priority over flatter portions. Significant points are actually the least number of shape points that can generate the original shape without any

distortion, i.e., the boundary points are significant points where there is a change in shape direction. This crucially means that shape information is integrated into the CP generation process and that consecutive significant points will not necessarily be separated by 1 *pel* as is the case with shape points, so the larger the distance between consecutive significant points, the greater their influence upon the shape approximation. In these circumstances, a shape descriptor based solely on significant points can produce a higher distortion because influential significant points may be excluded from being CP. To reduce the likelihood of losing such influential significant points as CP, *supplementary points* are inserted equidistantly between the significant points.

If the combination of significant and supplementary points is collectively referred to as *candidate boundary points* (CBP) in CP calculations, then a higher number of supplementary points infers the CBP tend towards the original shape points, while if there are insufficient supplementary points some significant points may not be adequately represented. To balance these extrema, the average distance between consecutive significant points over the entire shape is used to govern the judicious insertion of supplementary points. This procedure is mathematically formalised as follows:

Let the shape segments be represented as $\underset{1\le i\le N}{S_i} = \left\{s_{i,0}, s_{i,1}, \cdots, s_{i,|S_i|-1}\right\}$, where N is the number of segments, $|S_i|$ the number of shape points in the i^{th} segment and for a closed shape $s_{i,|S_i|-1} = s_{(i+1)\%N,0}$, where % is the modulus operator, so the set S_i forms an enclosed contour. If the significant points of the i^{th} segment are denoted by Sig_i then $\underset{1\le i\le N}{Sig_i} = \left\{sig_{i,0}, sig_{i,1}, \cdots, sig_{i,|Sig_i|-1}\right\} \subseteq S_i$ where $|Sig_i|$ is the cardinality of the set Sig_i. If $d\left(sig_{i,k-1}, sig_{i,k}\right)$ denotes the Euclidean distance between two consecutive significant points $sig_{i,k-1}$ and $sig_{i,k}$ in the i^{th} segment, the average distance between consecutive significant points over the entire shape is $d_{avg} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Sig_i-1|}\sum_{k=1}^{|Sig_i-1|} d\left(sig_{i,k-1}, sig_{i,k}\right)$. If $d\left(sig_{i,k-1}, sig_{i,k}\right) > d_{avg}$, then supplementary points are now inserted between $sig_{i,k-1}$ and $sig_{i,k}$, with the first (sp_1) being placed a distance d_{avg} from $sig_{i,k-1}$, and provided $d\left(sp_1, sig_{i,k}\right) > d_{avg}$, a further supplementary point is placed a distance d_{avg} from sp_1. This process is repeated until $d\left(sp_l, sig_{i,k}\right) \le d_{avg}$, where sp_l is the last supplementary point. Now let the CBP to be used to calculate the CP for

each segment be defined as:- $\underset{1\le i\le N}{B_i} = \{b_{i,0}, b_{i,1} \cdots, b_{i,|B_i|-1}\}$. The CP of a cubic BC for the i^{th} segment can then be determined by equal sampling, which as will be shown in the next section, reduces the overall bit-rate requirement:

$$v_{i,0} = b_{i,0}; v_{i,1} = b_{i,\left\lfloor \frac{|B_i|}{3} \right\rfloor}; v_{i,2} = b_{i,\left\lfloor \frac{|2*B_i|}{3} \right\rfloor}; v_{i,3} = b_{i,|B_i|-1} \quad \textbf{(3)}$$

It is therefore only necessary to encode those particular CP from which the approximated shape can be decoded, and in order to achieve better efficiency, these CP are normally differentially encoded.

Control Point Encoding

For encoding purposes, an *Enhanced Object-Adaptive Vertex Encoding* (E-OAVE) method is proposed that both differentially encodes the CP and utilises the inherent equidistance between the consecutive CP obtained from (3). The main reason for choosing OAVE is that apart from efficiently exploiting the coordinate information, it crucially enables the inherent regularity in the CP distances to be seamlessly embedded, thereby facilitating further bit-rate savings. OAVE comprises two major component blocks: *i)* object-level relative location dynamic range adaptation and *ii)* vertex-based encoding. Observing the equidistance between the consecutive CP, OAVE is enhanced in the first block by taking advantage of the fact that since the CP are approximately equidistance, the dynamic range will be lower. The overall bit-rate is then reduced by the second component block. The complete E-OAVE algorithm will now be delineated. If $C = \{c_0, c_1, \cdots, c_{L-1}\}$ is the ordered set of vertices to be encoded, then the various steps involved in the new enhanced object-level relative location dynamic range adaptation process are given in Algorithm 1.

Table 1: Relative-location dynamic ranges and their indication symbols

Dynamic Range Indicator	Relative-location dynamic range
0	$-1 \le x,y \le 1$
1	$-3 \le x,y \le 3$
2	$-7 \le x,y \le 7$
3	$-15 \le x,y \le 15$
4	$-31 \le x,y \le 31$
5	$-63 \le x,y \le 63$
6	$-127 \le x,y \le 127$
7	$-255 \le x,y \le 255$

Algorithm 1: Object-level relative location dynamic range adaptation.

1. Calculate the relative address R of vertices $R_i = (R_{i,x}, R_{i,y}) = (c_{i,x} - c_{i-1,x}, c_{i,y} - c_{i-1,y}),\ \ 1 \le i \le L-1$.
2. Determine $X_{\min} = \min_{1 \le i \le L-1} \{|R_{i,x}|\}$ and $Y_{\min} = \min_{1 \le i \le L-1} \{|R_{i,y}|\}$, where min is the minimum and $|\cdot|$ is the absolute value.
3. Obtain $\Delta x_{\max} = \max\{\Delta x_i\} = \max\{|\,|R_{i,x}| - X_{\min}\,|\}$ and $\Delta y_{\max} = \max\{\Delta y_i\} = \max\{|\,|R_{i,y}| - Y_{\min}\,|\}$
4. Select two indicators ind_x and ind_y from Table 1, which correspond to the smallest dynamic range that includes $\Delta x_{\max}$ and $\Delta y_{\max}$. Also two indicators $ind1_x$ and $ind1_y$ corresponding to $X_{\min}$ and $Y_{\min}$.
5. Encode indicators ind_x, ind_y, $ind1_x$ and $ind1_y$ using 3-bit *fixed length code* (FLC).

Since the CP are sampled at equal distances in (3), the deviation of the $|R_{i,x}|$ components with respect to $X_{\min}$ (and also $|R_{i,y}|$ with respect to $Y_{\min}$) are not very high, so both $\Delta x_{\max}$ and $\Delta y_{\max}$ in Step 3 will be small, as will the ensuing bit requirement. If conversely the deviations between consecutive CP are high, the coding will incur at most 6 extra bits (for an additional pair of dynamic range indicators) compared with the original OAVE algorithm. It needs to be highlighted however, that given the CP calculation scheme in (3), pragmatically the likelihood of this occurrence is extremely low (almost zero).

The next stage in the BCGSE model is vertex coding, where with the exception of the first CP (c_0), whose absolute address is directly encoded, all other CP are differentially encoded in accordance with Algorithm 2.

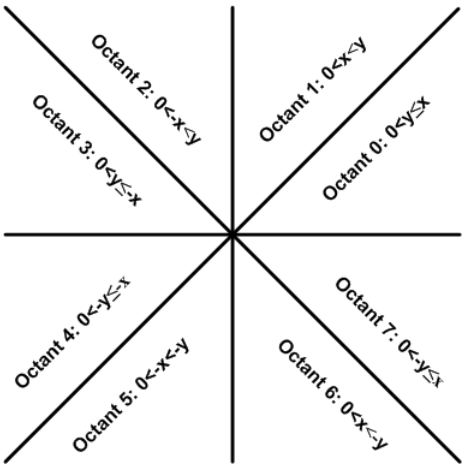


Figure 1: Octants in the Cartesian coordinates

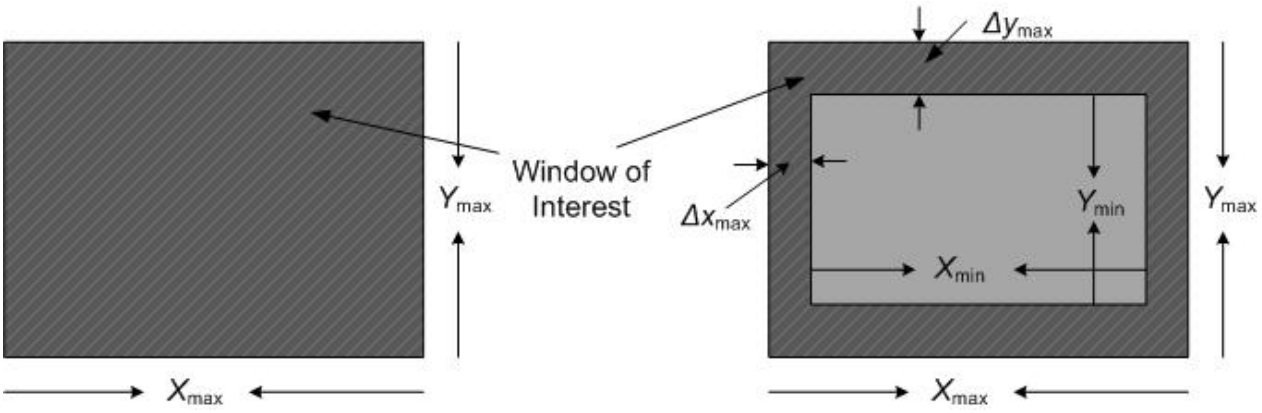
Algorithm 2: Encoding the relative address R_i for each vertex.

1. Determine the *octant number* of R_i according to Figure 1.
2. Encode each octant number using a 3-bit FLC.
3. Determine the major component from the octant number and encode its corresponding Δx_i or Δy_i values respectively using $ind_x + 1$ or $ind_y + 1$ bits depending upon the coordinate.
4. For the minor component, encode Δx_i or Δy_i, respectively using $\min\left\{ind_x + 1, \left\lceil \log_2\left(\left|R_{i,y}\right| - Y_{\min} + 1\right)\right\rceil\right\}$ and $\min\left\{ind_y + 1, \left\lceil \log_2\left(\left|R_{i,x}\right| - X_{\min} + 1\right)\right\rceil\right\}$ bits.
5. Add up all required bits, R_{total}.

To appreciate the overall coding efficiency achieved by the new E-OAVE algorithm, consider the following example. Let $C = \{(60,51),(80,60),(62,52),(45,45),(62,37)\}$, so $R = \{(20,9),(-18,-8),(-17,-7),(-17,-8)\}$. This means $X_{\min} = 17$ and $Y_{\min} = 7$, so $ind1_x = 4$ and $ind1_y = 2$. Also $\Delta x = \{3,1,0,0\}$ and $\Delta y = \{2,1,0,1\}$ from Step 3 of Algorithm 1. Hence $\Delta x_{\max} = 3$ and $\Delta y_{\max} = 2$, with $ind_x = 1$ and $ind_y = 1$. As all initial parameters must be coded,

to encode R_1, E-OAVE requires 3 *bits* for the octant number, and 2 *bits* each for Δx_1 and Δy_1 giving a total of 7 *bits*. In contrast, for the original OAVE technique [13]: $X_{\max} = 20$ and $Y_{\max} = 9$, so $ind_x = 4$ and $ind_y = 3$, thus to encode R_1 incurs 3 *bits* for the octant, 5 *bits* for the major component and a further 4 *bits* for the minor component, i.e., a total of 14 *bits*. This represents a 50% saving for every CP, though it needs to be emphasised that setting the initial parameters in OAVE incurs only 6 *bits* (for ind_x, ind_y), while it takes 12 *bits* for E-OAVE (for ind_x, ind_y, $ind1_x$ and $ind1_y$) and in the worst case, a further 14 *bits* to encode $X_{\min}$ and $Y_{\min}$, so the overhead for parameter initialisation can be up to a maximum of 26 *bits.* A pragmatic interpretation of this impost is that provided the number of CP is greater than four, E-OAVE will guarantee a lower bit-rate requirement, since four CP merely translates to a single cubic BC segment.

The main conclusion from the above analysis is that whenever E-OAVE is applied in preference to OAVE, increasing the number of curve segments will always lead to more efficient coding. The key difference between the OAVE and E-OAVE approaches is best visualised in the symbolic example in Figure 2. While in the original OAVE, the window-of-interest to encode the next CP is bounded by $X_{\max}$ and $Y_{\max}$, in E-OAVE, the area bounded by $X_{\min}$ and $Y_{\min}$ is ignored because the next CP cannot lie with this region, so the overall window size is reduced and coding efficiency improved.



(a) Original OAVE (b) Enhanced OAVE

Figure 2: Illustration of the difference between the original OAVE and E-OAVE techniques

As will be evinced in Section 4, this new encoding strategy provides notably superior results compared with the *dynamic fixed length coding* (DFLC) strategy in [11] and [12] because DFLC only utilises the periodicity in the

CP intervals. In contrast, E-OAVE exploits both the coordinate level information from the octant basis and object level information from the dynamic range indicators, conjointly with the CP distance regularity. In certain CP coding scenarios, the resulting bit-rate may exceed the admissible bit-rate whereupon it is important to take remedial action to maintain the permissible bit-rate. The next section presents a nonlinear optimisation solution for determining the number of segments able to be accommodated for a prescribed bit-rate.

Nonlinear optimisation technique to accommodate admissible bit-rates

Without loss of generality, the overall bit-rate requirement R_{total} is a non-decreasing function of N, since as N increases, so does the number of CPs, with the corollary that in such situations, R_{total} will increase. Since the octant number together with the major and minor components of each CP has to be encoded, for an admissible bit-rate constraint $R_{\max}$, pragmatically N cannot be made arbitrarily large. Conversely, a smaller N implies a less efficient utilisation of the available bit-rate and the propensity of large distortion, so an optimum value of N must be determined. The relationship between N and the bit-rate requirement is not of necessity linear, because for larger N, the distance between consecutive CP is smaller. Consequently, while the CP number linearly increases with N, the corresponding bit-rate requirement will not be linear, which precludes N being determined by a linear search method. A nonlinear optimization strategy is therefore proposed, with the problem formulated as follows:

$$\begin{aligned} &\text{maximise } N', \text{ where } N' \text{ is the candidate value of } N \\ &\text{subject to : } R_{total}(N') \le R_{\max} \\ &\qquad\qquad N' \in Z^{+}, \text{ where } Z^{+} \text{ is the set of positive integers.} \end{aligned} \tag{4}$$

The solution to (4) is the optimal value of N for BCGSE, which can be obtained by various search techniques including, the bisection method [19], convex hull search [20] and Bezier search [20]. For simplicity, the *bisection* method is adopted in this framework.

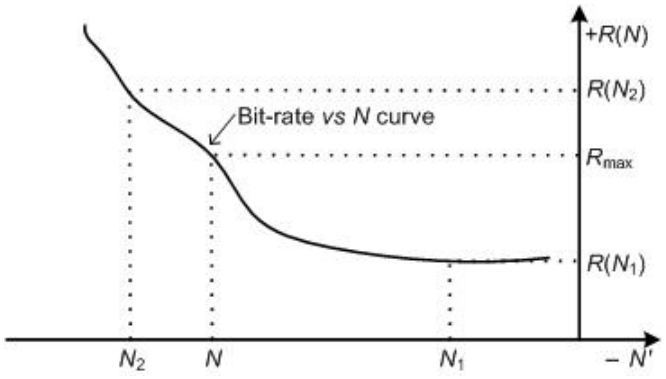


Figure 3: Example showing the *bisection* method for determining optimal *N* while maintaining $R_{\max}$.

Figure 3 illustrates the principle behind the *bisection* method to obtain the optimal N, i.e., maximum N', for an admissible bit-rate $R_{\max}$. Two initial values of N' are chosen such that N_1 results in R_{total} being lower than $R_{\max}$ whilst N_2 gives an R_{total} greater than $R_{\max}$. By exploiting the non-decreasing property of R_{total}, the optimal value of *N* must then lie between N_1 and N_2. The bisection interval thus becomes $N_m = \frac{N_1+N_2}{2}$, with R_{total} being recalculated so if $R_{total}(N_m) \geq R_{\max}$ then $N_2 = N_m$, otherwise $N_1 = N_m$. Repeating this procedure generates ever tighter bounds on the optimal *N*, so if at some point $N_1 \geq N_2$, then N_2 is the maximum value of N', i.e., the optimal *N*.

The choice of the two initial values of N' controls both the convergence and computational cost of determining the optimal *N*, with the important constraint $R_{total}(N_1) \leq R_{total}(N) \leq R_{\max} \leq R_{total}(N_2)$ always being upheld. Ideally the initial values of N_1 and N_2 will be 1 and ∞ respectively, though $R_{\max}$ imposes an inherent upper and lower bound upon N_2, with at least *5 bits* being required to encode a CP: *3 bits* for the octant number, and *1 bit* each for the major and minor components. Furthermore, as each additional cubic curve segment requires 3 CP, if the initial parameters are coded by r' bits, the maximum value of N_2 will be $\left\lfloor \frac{R_{\max}-r'}{3 \cdot 5} \right\rfloor = \left\lfloor \frac{R_{\max}-r'}{15} \right\rfloor$, where $\lfloor\ \rfloor$ is the *flooring* operator. With regard to N_1, a maximum of *19 bits* is mandated to encode any CP, comprising *3 bits* for the octant number, and *8 bits* each for the major and minor components. Therefore, if the $R_{\max}$ is meant to be fully utilised, the minimum N_1 value will be $\left\lfloor \frac{R_{\max}-r'}{3 \cdot 19} \right\rfloor = \left\lfloor \frac{R_{\max}-r'}{57} \right\rfloor$. In summary, since $\frac{N_2}{N_1} \approx 4$, the optimal value of N

can be determined in two iterations of the *bisection* method, so highlighting the computational efficiency of the BCGSE non-linear optimisation technique in automatically adapting to bit-rate constraints.

Decoding shape information

From a decoder perspective, since the differential and parametric representation of the encoded shape information and its periodicity are dynamically determined, and by implication also the delimiter of each parameter, the decoder will be able to correctly parse these parameters from the encoded bit stream to achieve shape reconstruction.

Computational complexity analysis

The three constituent components for the total computational time for these algorithms are: the division of the boundary contour into segments, CP calculation for each segment and CP encoding. As the new BCGSE model applies a similar type of cornerity detection to that used in [1], [2] and [4], the first phase always takes the same computational time. For CP calculation, BCGSE, [1] and [4] take $O\left(\left|S_{\max}\right|\right)$, where $\left|S_{\max}\right|$ is the maximum number of boundary points for a segment, while the SDCC algorithm [2] requires $O\left(r\cdot\left|S_{\max}\right|\right)$, where r is the number of iterations necessary to ensure convergence and also involves the computationally intensive process of calculating both a chord-length and the Bernstein function for each value of t. SDCB [3] conversely, is computationally efficient because it simply divides the shape into segments and generates the CP based on the number of boundary points though since the segment division is made arbitrarily without any perceptual consideration, large distortions can ensue as will be witnessed in the next section. The computational expenditure of the nonlinear optimisation solution for determining segment numbers in the BCGSE framework increases commensurately with the number of bisection iterations, with the overhead incurred to calculate the optimal N taking $\lceil \log_2(N_2 - N_1) \rceil$ iterations, where $\lceil\ \rceil$ is the *ceiling* operator. Given the proposed initial values of N_1 and N_2, only two iterations are required to obtain the optimal value of N.

Apart from these BC-based shape coding techniques, for completeness the experimental results of BCGSE will be compared with the vertex-based ORD optimal shape coding framework [8], [9] in the next section, so it is worth

highlighting the overall computational complexity of vertex-based algorithms is in fact $O\left(N_B^3\right)$, where N_B is the total number of contour points on a shape.

4. Results and Analysis

The widely-adopted [20] shape distortion measurement metrics D_{max} and D_{ms} are used for the peak and *Mean-Square* (MS) distortions respectively. These are formally expressed as:

$$D_{\max} = \max_{1 \le i \le N} \max_{0 \le j \le |S_i|-1} d'\left(s_{i,j}, v_i\right) \tag{5}$$

$$D_{ms} = \frac{1}{M} \sum_{1 \le i \le N} \sum_{0 \le j \le |S_i|-1} d'^2\left(s_{i,j}, v_i\right) \tag{6}$$

where $d'\left(s_{i,j}, v_i\right)$ is the minimum Euclidean distance of $s_{i,j}$ for the j^{th} shape point of the i^{th} segment, from the corresponding BC approximation v_i, and M is the number of contour points, with $D_{\max}$ and D_{ms} both being measured by the *accurate distortion measurement* technique in [21]. The performance of BCGSE has been rigorously tested upon a number of popular object shapes used extensively in the literature [1], [3] and [4]. In addition, the MPEG-4 D_n metric was also used to validate the new BCGSE model provided superior performance to existing techniques for any accepted metric. D_n represents shape distortion by the ratio of the number of erroneous pixels in the approximating shape to the total number of pixels in the original shape [8]. It is formally defined as:

$$D_n = \frac{\text{number of pixels mismatched in the approximated shape}}{\text{number of pixels in the original shape}} \tag{7}$$

and is normally defined in percentile terms.

A series of experiments were performed upon the various test shapes. The subjective results are presented in Figure 4, while the corresponding numerical distortion and bit-rate results are summarised in Tables 2 and 3 respectively. Figure 4(a) shows a comparison of the decoded shape using SDCB and the BCGSE algorithm applied upon the *Lip* object [4], which is characterised by repetitious vertices and loops with 5 segments. It is visually apparent the BCGSE shape approximation is very close to the original, while SDCB generates a structurally very different shape. This is numerically confirmed by the corresponding distortion results in Table 2 which reveal SDCB

produced a D_{max}, D_{ms} and D_n of 10.2 *pel*, 5.4 pel^2 and 2.55% respectively, in contrast to only 1.45 *pel*, 0.89 pel^2 and 0.61% for BCGSE, so endorsing the underlying strategy of considering both cornerity and loops of a shape. Table 2 also corroborates that BCGSE consistently produced the lowest distortion of all the methods analysed (including those which considered curvature information when sub-dividing the shape into segments), so vindicating the integration of shape information within the CP generation process.

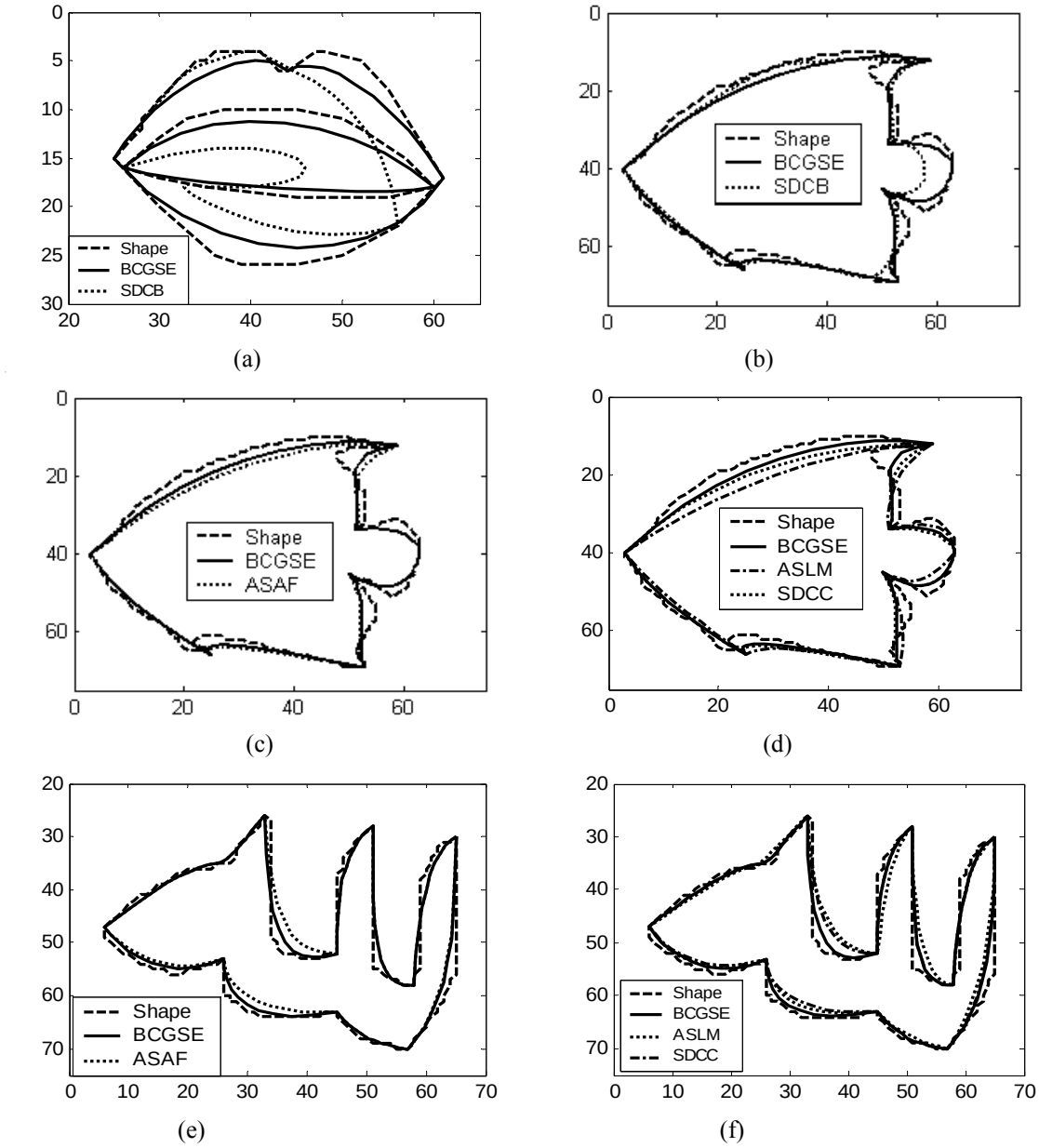


Figure 4: Experimental results for various selected tests shapes: a) *Lip*; b)-d) *Fish*; and e)-f) *Arabic* characters

Table 2: Distortion values for various shape representations

Shape → Algorithm↓	*Fish* [3]			*Arabic character* [1]			*Lip* [4]		
	D_{max} (*pel*)	D_{ms} (*pel*^{2})	D_n (%)	D_{max} (*pel*)	D_{ms} (*pel*^{2})	D_n (%)	D_{max} (*pel*)	D_{ms} (*pel*^{2})	D_n (%)
BCGSE	3.0	2.7	0.85	1.12	0.80	0.61	1.45	0.89	0.61
SDCB	6.0	4.3	1.81	2.1	1.35	0.95	10.2	5.4	2.55
ACAF	4.0	3.2	1.46	1.3	0.95	0.65	1.65	1.05	0.65
SDCC	3.9	3.46	1.45	1.35	1.2	0.95	1.7	1.2	0.71
ASLM	6.0	6.55	1.84	1.4	1.4	0.66	1.8	1.5	0.73

The next series of experiments were performed upon the *Fish* shape [3] which is characterised by having some sharp as well as more gradual contour portions. The comparative results for SDCB, ACAF, ASLM and SDCC for 7 segments are presented in Figures 4(b)-(d) and Table 2. BCGSE once again produced the lowest peak, MS and D_n distortion values of 3.0 *pel*, 2.7 *pel*^{2} and 0.85% respectively, outperforming all other BC-based shape descriptors, with a similar observation being made for the *Arabic character* [1] shape in Figures 4(e)-(f).

Table 3a: Bit-rate requirements in shape representation where N is the number of segments. The values in the parentheses indicate the generated peak distortion (D_{max}) in *pel*.

Shape	**Algorithm**	$N = 5$	$N = 6$	$N = 7$	$N = 8$
Fish [3]	BCGSE	126 (3.85)	166 (3.0)	194 (3.0)	234 (2.5)
	SDCB [3]	244 (6.0)	292 (6.0)	340 (5.0)	388 (4.0)
	DFLC [11], [12]	193 (3.85)	230 (3.0)	267 (3.0)	304 (2.5)
	OAVE*	195 (3.85)	228 (3.0)	267 (3.0)	280 (2.5)
Lip [4]	BCGSE	123 (1.45)	159 (1.42)	191 (1.0)	215 (1.0)
	SDCB [3]	244 (10.2)	292 (9.5)	340 (8.0)	388 (7.0)
	DFLC [11], [12]	193 (1.5)	230 (1.42)	267 (1.0)	304 (1.0)
	OAVE*	170 (1.5)	190 (1.42)	217 (1.0)	235 (1.0)

* CP encoded using the original OAVE technique [13].

Table 3b: Average bit-rate requirements in shape representation where N is the number of segments. The values in the parentheses indicate the generated peak distortion (D_{max}) in *pel*.

Shape	**Algorithm**	N **= 5**	N **=10**	N **=15**	N **=20**
MissAmerica (qcif)	BCGSE	205 (8.0)	355 (5.0)	525 (2.0)	515 (1.0)
	SDCB [3]	480 (10)	940 (6.0)	1140 (2.4)	1920 (1.0)
	DFLC [11], [12]	302 (8.0)	401 (5.0)	555 (2.0)	550 (1.0)
	OAVE*	242 (8.0)	375 (5.0)	530 (2.0)	520 (1.0)
Akiyo(qcif)	BCGSE	200 (8.0)	352 (5.0)	520 (2.0)	512 (1.0)
	SDCB [3]	480 (10)	940 (6.0)	1140 (2.4)	1920 (1.0)
	DFLC [11], [12]	300 (8.0)	400 (5.0)	550 (2.0)	547 (1.0)
	OAVE*	241 (8.0)	370 (5.0)	527 (2.0)	518 (1.0)
Stefan (sif)	BCGSE	265 (10.0)	435 (8.0)	540 (4.0)	585 (2.0)
	SDCB [3]	480 (12)	960 (10.0)	1140 (6.0)	1920 (3.0)
	DFLC [11], [12]	355 (10.0)	510 (8.0)	610 (4.0)	650 (2.0)
	OAVE*	280 (10.0)	458 (8.0)	585 (4.0)	605 (2.0)

* CP encoded using the original OAVE technique [13]. *qcif* (176×144 *pels*) and *sif* (352×240 *pels*) are the spatial resolutions of a frame.

To evaluate the impact of the E-OAVE strategy for CP encoding on the shape descriptor length, an investigation was undertaken analysing the bit-rate requirement for different techniques for various segment numbers, N. For an equitable comparison, it was assumed that the absolute coordinate values used in each algorithm required one byte of memory, though as mentioned in Section 1, both the direction and magnitude components in the SDCB algorithm are usually specified as floating point numbers, and so in practice these would mandate a parameter storage overhead of at least 4 bytes within the IEEE Standard 754 floating point numbering standard. Table 3a evinces that for the *Fish* object with 7 segments, the BCGSE descriptor length was 194 *bits* compared with 267 *bits* for the original OAVE CP encoding technique, a 28% bit rate improvement. This is directly attributable to the coding performance of E-OAVE in exploiting the causal regularity in CP distances, with analogous performance improvements observed for differing N values, as well as for the *Lip* object. Interestingly for the *Fish* shape, the DFLC techniques [11] and [12] also required 267 *bits* compared with the 340 *bits* for parametric coding [3], to corroborate the rationale behind E-OAVE in achieving superior bit rates in comparison to existing CP encoding methods. It is especially noteworthy in Table 3a, that while BCGSE, DFLC [11], [12] and the original OAVE [13] algorithm all encoded exactly the same CP set, thereby generating the same distortion and N values for each shape, BCGSE always incurred the lowest bit-rate. In contrast, SDCB [3] produced a different CP set and by implication, different distortion and bit-rate values. Table 3b also presents results for the popular standard test shape sequences

Miss America, Akiyo and *Stefan*. These results also exhibit the same trend that BCGSE always provided superior results compared to the other techniques.

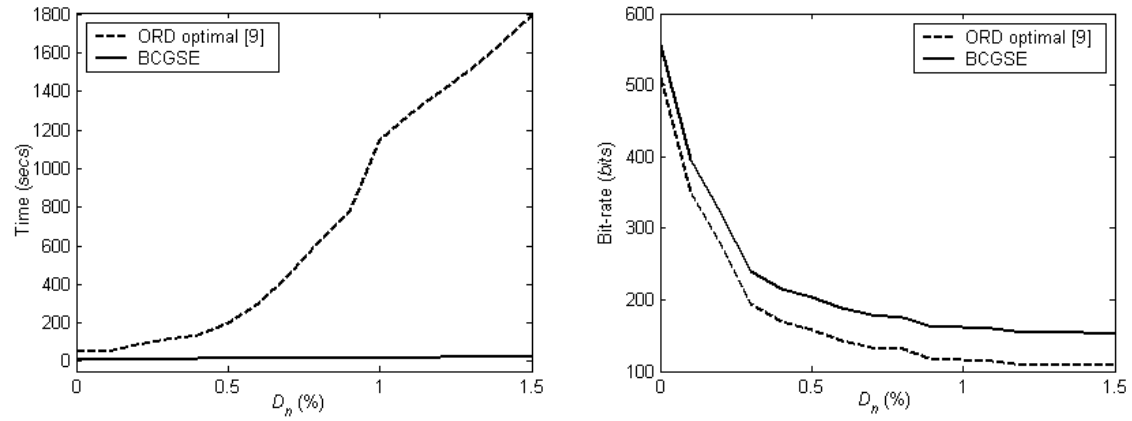
To provide a comparative analysis of the computational complexities, the overall CPU times required for CP generation of the various algorithms were determined. Each algorithm was implemented in Matlab 6.1 (The Mathworks Inc.) and run on a 2.8GHz Pentium-4 processor, with 512MB RAM under Windows XP. As all existing algorithms did not have an explicit CP coding strategy, only the CPU times incurred in the CP generation phase were compared. The results in Table 4 reveal that BCGSE required less time than both the ASAF and SDCC techniques, so that for example, the overall CPU time requirements for the *Arabic* character were 7.9, 9.01 and 9.03 seconds respectively. The higher ASAF time is due to every boundary point in a curve segment involving a computationally expensive chord-length parameterisation and CP calculation, while for SDCC the trial-and-error method employed for CP generation involves iteratively computing the distortion for all vertices in the segment together with their tangents. In contrast, BCGSE firstly reduces the number of vertices by determining the significant points and then using this smaller vertex set instead of the larger set of boundary points. Interestingly Table 4 shows that both SDCB and ASLM were computationally faster than BCGSE, though it is important to stress that ASLM employs only a quadratic BC which necessitates the calculation of just one intermediate CP compared with BCGSE, requires two points, since it uses a cubic order BC [7]. The principal drawback in using lower order curves is that they produce higher distortions as evidenced by the corresponding results in Table 2. SDCB in fact, only processes the index number of the vertices on the boundary and so does not consider shape information at all which inevitably leads to higher distortion values, particularly when the shape contour exhibits sharp variations such as in the *Lip* object and *Arabic* character.

Table 4: Required CPU time (in *seconds*) for various algorithms to generate the CP for different shapes.

Shape → Algorithm↓	*Fish* [3]	*Arabic character* [1]	*Lip* [4]
BCGSE	3.01	7.01	2.32
SDCB	1.72	4.01	1.62
ACAF	4.02	9.01	3.05
SDCC	3.99	9.03	3.12
ASLM*	2.51	6.05	2.25

*ASLM uses quadratic BC while all other algorithms use cubic BC

For completeness a further set of experiments were conducted comparing the proposed BCGSE model with the classical vertex-based ORD optimal shape coding framework [9], [10] using B-splines. Figures 5(a) and (b) respectively plot the computational-time (for coding only) vs. distortion and bit-rate vs. distortion (RD) results upon the *Fish* shape.



(a) Distortion vs coding time (b) Distortion vs bit-rate

Figure 5: Comparative performance of the new BCGSE model and the vertex-based ORD optimal shape coding framework for the *Fish* shape

Figure 5(a) reveals that while the coding time for BCGSE remains approximately constant for increasing admissible distortion, confirming the complexity analysis in Section 3, the vertex-based optimal ORD framework incurs significantly higher computational and power overheads at larger distortions due to the exhaustive search method used to locate the minimum bit-rate path within the admissible distortion bound from a large-weighted directed-acyclic-graph. This huge discrepancy in coding performance more than offsets the bit-rate savings evident in Figure 5(b), of the slightly better overall RD performance of the ORD framework.

Table 5: Adaptive admissible bit-rate results: The optimal number of segments *N* obtained at different prescribed bit rates for various shape sequences. The values in parentheses represent the respective utilised bit-rates**.**

Shape	**Admissible bit-rate** $R_{\max}$ (*bits*)	
	230	440
MissAmerica(qcif)	*N=5* (205)	*N=13* (438)
Akiyo(qcif)	*N=6* (228)	*N=13* (435)
Stefan (sif)	*N=4* (229)	*N=10* (435)

Finally, a series of experiments were conducted to appraise the performance of the nonlinear optimisation technique for adaptive bit-rate constraint, with the corresponding results summarised in Table 5 for the test shape sequences used earlier in Table 3b. These reveal that with an admissible bit-rate $R_{max} = 230$ *bits* for *MissAmerica*, a maximum of 5 segments are required with 205 *bits* being utilised, while for 440 *bits,* 13 segments is the optimal number which correspondingly used 438 *bits*. These results are notably congruent with the findings in summarised Table 3b. Similar findings are observed for the other sequences to corroborate the capability of the BCGSE paradigm to sustain an admissible bit rate by automatically adapting to any bit-rate constraints.

5. Conclusion

While Bezier curves have been applied in many different domains and applications including object shape description, a critical aspect in their performance is the appropriate selection of the segments and *control points* (CP) This paper has presented a *generic shape encoder using Bezier curves* (BCGSE) algorithm which provides an innovative strategy for CP calculation along with sub-dividing a shape into segments by considering domain specific shape information and a new efficient CP coding strategy called *Enhanced Object-Adaptive Vertex Encoding*. It has also developed a nonlinear optimisation strategy to enable the encoder to adapt to admissible bit-rate constraints. Both perceptual and numerical results have conclusively proven the improved performance in terms of both minimum distortion and bit-rate requirements of the new BCGSE model in comparison with other existing shape descriptor techniques.

6. Acknowledgements

The authors acknowledge this work has been supported by a Monash University Post Publications Award, ARC discovery grants (DP0664228 and DP0771294), a University of Western Australia postdoctoral fellowship and a research development award. The authors especially wish to thank the reviewers for their perceptive comments, criticisms and overall guidance in significantly improving this paper.

7. References

[1] Sarfraz, M., and Khan, M.A.: 'Automatic outline capture of Arabic fonts', Information Sciences, 2002, 140, (3-4), pp. 269-281

[2] Yang, H.-M., Lu, J.-J., and Lee, H.-J.: 'A Bezier curve-based approach to shape description for Chinese calligraphy characters', Proceedings of Sixth International Conference on Document Analysis and Recognition, 2001, pp. 276-280

[3] Cinque, L., Levialdi, S., and Malizia, A.: 'Shape description using cubic polynomial Bezier curves', Pattern Recognition Letters, 1998,19, (9), pp. 821-828

[4] Shdaifat, I., Grigat, R., and Langmann, D.: 'Active shape lip modelling', Proceedings of IEEE International Conference on Image Processing (ICIP), 2003, pp. 875-878

[5] Soares, L.D., and Pereira, F.,: 'Spatial shape error concealment for object-based image and video coding', IEEE Transactions on Image Processing, 2004, 13, (4), pp.586-599

[6] Zhang, R., and Wang, G., : 'Some estimates of the height of rational Bernstein-Bezier triangular surfaces', Proc. Geometric Modeling and Processing, 2004, pp.79-84

[7] Bartels, R.H., Beatty, J.C., and Barsky, B.A.,: 'An introduction to splines for use in computer graphics & geometric modeling', Morgan Kaufmann Publishers, 1987

[8] Katsaggelos, A.K., Kondi, L.P., Meier, F.W., Ostermann, J., and Schuster, G.M.,: 'MPEG-4 and rate-distortion-based shape-coding techniques', Proceedings of IEEE, 1998, 86, (6), pp.1126-1154

[9] Kondi, L.P., Melnikov, G., and Katsaggelos, A.K.,: 'Joint optimal object shape estimation and encoding', IEEE Transactions on Circuits and Systems for Video Technology, 2004, 14, (4), pp.528-533

[10] Sohel, F.A., Dooley, L.S., and Karmakar, G.C.,: 'New- dynamic enhancements to the vertex-based rate-distortion optimal shape coding framework', IEEE Transactions on Circuits and Systems for Video Technology, 2007, 17, (10), 1408-1413.

[11] Sohel, F.A., Karmakar, G.C., and Dooley, L.S.,: 'A generic shape descriptor using Bezier curves', Proc. Int. Conf. Information Technology: New Trends in Image Processing (ITCC), 2005, II, pp. 95-100

[12] Sohel, F.A., Karmakar, G.C., and Dooley, L.S., 'An improved shape descriptor using Bezier curves', Proc. First Int. Conf. Pattern Recognition and Machine Intelligence (PReMI), *Lecture Notes on Computer Science*, Springer, 2005, 3776, pp.401-406

[13] O'Connell, K.J.,: 'Object-adaptive vertex-based shape coding method', IEEE Transactions on Circuits and Systems for Video Technology, 1997, 7, (1), pp. 251-255

[14] Hill Jr., F. S.,: 'Computer Graphics', Prentice Hall, Englewood Cliffs, 1990

[15] Chetverikov, D. and Szabo, Z.,: 'A simple and efficient algorithm for detection of high curvature points in planar curves', Proc. 10th International Conference, CAIP 2003, Groningen, The Netherlands, *Lecture Notes on Computer Science*, Springer, 2003, 2756, pp. 746-753

[16] Phillips, T.-Y., and Rosenfeld, A.,: 'A method of curve partitioning using arc-chord distance', Pattern Recognition Letters, 1987, 5, (4), pp. 285-288

[17] Liu, H.C., and Srinath, M.D.,: 'Corner detection from chain-code', Pattern Recognition, 1990, 20, (3), pp. 51-68

[18] Beus, H.L., and Tiu, S.S.H.,: 'An improved corner detection algorithm based on chain coded plane curves', Pattern Recognition, 1987, 20, (3), pp. 291-296

[19] Scarborough, J. B.,: 'Numerical mathematical analysis', Baltimore, Johns Hopkins, 1966

[20] Schuster, G.M., and Katsaggelos, A.K.,: 'Rate-distortion based video compression-optimal video frame compression and object boundary encoding, Kluwer Academic Publishers, 1997

[21] Sohel, F.A., Dooley, L.S., and Karmakar, G.C.,: 'Accurate distortion measurement for generic shape coding', Pattern Recognition Letters, 2006, 27, (2), pp.133-142