Beziers: A Curve-Based Approach to Shape Representation

F.A. Sohel, G.C. Karmakar, L.S. Dooley, and M. Bennamoun

1School of Computer Science and Software Engineering, The University of Western Australia, WA 6009, Australia
2Gippsland School of Information Technology, Monash University, Churchill, Victoria 3842, Australia
3Department of Communication and Systems, The Open University, Milton Keynes, MK7 6AA, UK

E-mail: Ferdous.Sohel@ieee.org

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 study 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, although CP encoding has generally been ignored, BCGSE embeds an efficient vertex-based encoding strategy exploiting the latent equidistance between consecutive CP. A non-linear 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.

1 Introduction

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 (ASLM) [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 (CBC) [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 (SCB) technique which divides the shape-boundary into an a priori number of segments, each comprising the same number of control 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. 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 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 end-points 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, 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 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 (Note: Preliminary ideas involved in this research were introduced in [11, 12]). Unlike [1, 2, 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 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 non-linear 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, whereas 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 Classical BCs

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

\[
v_j(t) = \sum_{k=0}^{n} \binom{n}{k} B_k^n(t), \quad 0 \leq t \leq 1
\]  

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. Although 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

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

(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.

3.1 Determining the CPs

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 defined either a priori or dynamically. The cornerity of a boundary point is characterised by the modulus operator, so the set \(S_i\), where \(S_i\) is the cardinality of the set \(S_i\), is the number of shape points in the \(i\)th segment and for a closed shape \(S_i = S_i(N,0)\), where \(N\) is the number of segments, \(S_i\) is the number of shape points in the \(i\)th segment and \(S_i\) is the cardinality of the set \(S_i\), which is the last supplementary point. If \(d(S_i, S_i)\) denotes the Euclidean distance between two consecutive significant points \(S_i, S_i\) in the \(i\)th segment, 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 \(S_{i,1 \leq i \leq N} = \{S_{i,1}, S_{i,2}, \ldots, S_{i,[N]}\}\), where \(N\) is the number of segments, \(S_{i,1}\) is the number of shape points in the \(i\)th segment and \(S_{i,1} = S_{i}(N,0)\). Then if \(S_{i,1} < \frac{1}{2}\), then supplementary points are now inserted between \(S_{i,1}\) and \(S_{i,1}\) with the first \(S_{i,1}\) being placed a distance \(d_{avg}\) from \(S_{i,1}\), and provided \(d(S_{i,1}, S_{i,1}) \geq d_{avg}\), a further supplementary point is placed a distance \(d_{avg}\) from \(S_{i,1}\). This process is repeated until \(d(S_{i,1}, S_{i,1}) \leq d_{avg}\), where \(S_{i,1}\) is the last supplementary point.

Now let the CBP to be used to calculate the CP for each segment be defined as \(B_{1 \leq i \leq N} = \{b_{i,1}, b_{i,2}, \ldots, b_{i,[N]}\}\). 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...
Algorithm 1: Object-level relative location dynamic range adaptation

1. Calculate the relative address \( R_i = (R_{ix}, R_{iy}) = (x_i, y_i), 1 \leq i \leq L - 1 \)

2. Determine \( X_{\min} = \min_{1 \leq i \leq L-1} \{ |R_{ix}| \} \) and \( Y_{\min} = \min_{1 \leq i \leq L-1} \{ |R_{iy}| \} \), where \( \min \) is the minimum and \( | \cdot | \) is the absolute value.

3. Obtain \( \Delta x_{\max} = \max \{ \Delta x_i \} = \max \{ |R_{ix}| - X_{\min} \} \) and \( \Delta y_{\max} = \max \{ \Delta y_i \} = \max \{ |R_{iy}| - Y_{\min} \} \).

4. Select two indicators \( \text{ind}_x \) and \( \text{ind}_y \) from Table 1, which correspond to the smallest dynamic range that includes \( \Delta x_{\max} \) and \( \Delta y_{\max} \). Also, two indicators \( \text{ind}_1^x \) and \( \text{ind}_1^y \) corresponding to \( X_{\min} \) and \( Y_{\min} \).

5. Encode indicators \( \text{ind}_1^x, \text{ind}_1^y, \text{ind}_x \), and \( \text{ind}_y \) using 3-bit fixed length code (FLC).

Since the CP are sampled at equal distances in (3), the deviation of the \( |R_{ix}| \) components with respect to \( X_{\min} \) (and also \( |R_{iy}| \) 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 \((0)\), whose absolute address is directly encoded, all other CP are differentially encoded in accordance with Algorithm 2.

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

1. Determine the octant number of \( R_i \) according to Fig. 1.

2. Encode each octant number using a 3-bit FLC.

3. Determine the major component from the octant number and encode its corresponding \( \Delta x \), or \( \Delta y \), values, respectively.
using $\text{ind}_x + 1$ or $\text{ind}_y + 1$ bits depending upon the coordinate.

4. For the minor component, encode $\Delta x_i$ or $\Delta y_i$, respectively, using $\min(\text{ind}_x + 1, \lceil \log_2(|R_x| - Y_{\text{min}} + 1) \rceil)$ and $\min(\text{ind}_y + 1, \lceil \log_2(|R_y| - X_{\text{min}} + 1) \rceil)$ bits.

5. Add up all required bits, $R_{\text{total}}$.

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 $R = (20, 9), (18, 9), (17, 7), (17, 8)$. This means that $X_{\text{min}} = 17$ and $Y_{\text{min}} = 7$, so $\text{ind}_x = 4$ and $\text{ind}_y = 2$. Also $\Delta x = 3, 1, 0, 0$ and $\Delta y = 2, 1, 0, 1$ from Step 3 of Algorithm 1. Hence $\Delta x_{\text{max}} = 3$ and $\Delta y_{\text{max}} = 2$, with $\text{ind}_x = 1$ and $\text{ind}_y = 1$. As all initial parameters must be coded, to encode $R_{\text{total}}$, E-OAVE requires 3 bits for the octant number, and 2 bits each for $\Delta x_i$ and $\Delta y_i$ giving a total of 7 bits. In contrast, for the original OAVE technique [13]: $X_{\text{max}} = 20$ and $Y_{\text{max}} = 9$, so $\text{ind}_x = 4$ and $\text{ind}_y = 3$, thus to encode $R_{\text{t}}$ incurs 3 bits for the octant, 5 bits for the major component and a further 4 bits for the minor component, that is, 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 $\text{ind}_x$, $\text{ind}_y$, $\text{ind}_1$, and $\text{ind}_2$) and in the worst case, a further 14 bits to encode $X_{\text{min}}$ and $Y_{\text{min}}$, 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 Fig. 2. While in the original OAVE, the window-of-interest to encode the next CP is bounded by $X_{\text{max}}$ and $Y_{\text{max}}$, in E-OAVE, the area bounded by $X_{\text{min}}$ and $Y_{\text{min}}$ is ignored because the next CP cannot lie with this region, so the overall window size is reduced and coding efficiency improved.

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, 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 non-linear optimisation solution for determining the number of segments able to be accommodated for a prescribed bit-rate.

3.3 Non-linear optimisation technique to accommodate admissible bit-rates

Without loss of generality, the overall bit-rate requirement $R_{\text{total}}$ is a non-decreasing function of $N$, since as $N$ increases, so does the number of CPs, with the corollary that in such situations, $R_{\text{total}}$ 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 $R_{\text{max}}$, 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 non-linear optimisation strategy is therefore proposed, with the problem formulated as follows

\[
R_{\text{total}}(N') \leq R_{\text{max}}
\]

\[
N' \in \mathbb{Z}^+, \quad \text{where } \mathbb{Z}^+ \text{ is the set of positive integers}
\]
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.

Fig. 3 illustrates the principle behind the bisection method to obtain the optimal $N$, that is, maximum $N'$, for an admissible bit-rate $R_{\text{max}}$. Two initial values of $N'$ are chosen such that $N_1$ results in $R_{\text{total}}$ being lower than $R_{\text{max}}$ whereas $N_2$ gives an $R_{\text{total}}$ greater than $R_{\text{max}}$. By exploiting the non-decreasing property of $R_{\text{total}}$, the optimal value of $N$ must then lie between $N_1$ and $N_2$. The bisection interval thus becomes $N'_m = N_1 + N_2/2$, with $R_{\text{total}}$ being recalculated so if $R_{\text{total}}(N_m) \geq R_{\text{max}}$ 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 $N'$, that is, 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 $R_{\text{total}}(N_{\text{lf}}) \leq R_{\text{total}}(N) \leq R_{\text{max}} \leq R_{\text{total}}(N_{\text{lf}})$ always being upheld. Ideally, the initial values of $N_1$ and $N_2$ will be 1 and $\infty$, respectively, though $R_{\text{total}}$ 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 $\lfloor (R_{\text{max}} - r'/3 \cdot 5) = \lfloor (R_{\text{max}} - r'/15) \rfloor$, 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 $R_{\text{max}}$ is meant to be fully utilised, the minimum $N_1$ value will be $\lfloor (R_{\text{max}} - r'/3 \cdot 19) = \lfloor (R_{\text{max}} - r'/57) \rfloor$. In summary, since $N_2/N_1 \approx 4$, 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.

### 3.4 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.

### 3.5 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, 4], the first phase always takes the same computational time. For CP calculation, BCGSE, [1, 4] take $O(4/5 |s_{\text{max}}|)$, where $|s_{\text{max}}|$ is the maximum number of boundary points for a segment, whereas the SDCC algorithm [2] requires $O(\cdot (|s_{\text{max}}|))$, 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 non-linear 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 $\log_2 (N_2 - N_1)$ 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 that the overall computational complexity of vertex-based algorithms is in fact $O(N_B^2)$, 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 $D_{\text{max}}$ and $D_{\text{ms}}$ are used for the peak and mean-square (MS) distortions, respectively. These are formally expressed as

\[
D_{\text{max}} = \max_{i=1}^{N} \max_{0 \leq |s_i| \leq |s_{\text{lf}}|-1} d'(s_{ij}, v_i) \tag{5}
\]

\[
D_{\text{ms}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{0 \leq |s_i| \leq |s_{\text{lf}}|-1} d^2(s_{ij}, v_i) \tag{6}
\]
where \( d'(s_{ij}, v_i) \) is the minimum Euclidean distance of \( s_{ij} \) 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 \( D_{\text{max}} \) and \( D_{\text{ms}} \) 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, 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].

\[
D_n = \frac{\text{number of pixels mismatched in the approximated shape}}{\text{number of pixels in the original shape}}
\]  

(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 Fig. 4, whereas the corresponding numerical distortion and bit-rate results are summarised in Tables 2 and 3a, b, respectively. Fig. 4a 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 five segments. It is visually apparent that the BCGSE shape approximation is very close to the original, whereas SDCB generates a structurally very different shape. This is numerically confirmed by the corresponding distortion results in Table 2, which reveal SDCB produced a $D_{\text{max}}$, $D_{\text{ms}}$ 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 subdividing the shape into segments), so vindicating the integration of shape information within the CP generation process.

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 seven segments are presented in Figs. 4b–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 Figs. 4e and f.

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 seven 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

### Table 2 Distortion values for various shape representations

<table>
<thead>
<tr>
<th>Shape</th>
<th>Algorithm</th>
<th>$D_{\text{max}}$, pel</th>
<th>$D_{\text{ms}}$, pel$^2$</th>
<th>$D_n$, %</th>
<th>$D_{\text{max}}$, pel</th>
<th>$D_{\text{ms}}$, pel$^2$</th>
<th>$D_n$, %</th>
<th>$D_{\text{max}}$, pel</th>
<th>$D_{\text{ms}}$, pel$^2$</th>
<th>$D_n$, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fish [3]</td>
<td>BCGSE</td>
<td>3.0</td>
<td>2.7</td>
<td>0.85</td>
<td>1.12</td>
<td>0.80</td>
<td>0.61</td>
<td>1.45</td>
<td>0.89</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>SDCB</td>
<td>6.0</td>
<td>4.3</td>
<td>1.81</td>
<td>2.1</td>
<td>1.35</td>
<td>0.95</td>
<td>10.2</td>
<td>5.4</td>
<td>2.55</td>
</tr>
<tr>
<td></td>
<td>ACAF</td>
<td>4.0</td>
<td>3.2</td>
<td>1.46</td>
<td>1.3</td>
<td>0.95</td>
<td>0.65</td>
<td>1.65</td>
<td>1.05</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td>SDCC</td>
<td>3.9</td>
<td>3.46</td>
<td>1.45</td>
<td>1.35</td>
<td>1.2</td>
<td>0.95</td>
<td>1.7</td>
<td>1.2</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>ASLM</td>
<td>6.0</td>
<td>6.55</td>
<td>1.84</td>
<td>1.4</td>
<td>1.4</td>
<td>0.66</td>
<td>1.8</td>
<td>1.5</td>
<td>0.73</td>
</tr>
</tbody>
</table>

SDCB and the BCGSE algorithm applied upon the Lip object [4], which is characterised by repetitious vertices and loops with five segments. It is visually apparent that the BCGSE shape approximation is very close to the original, whereas SDCB generates a structurally very different shape. This is numerically confirmed by the corresponding distortion results in Table 2, which reveal SDCB produced a $D_{\text{max}}$, $D_{\text{ms}}$ 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 subdividing the shape into segments), so vindicating the integration of shape information within the CP generation process.

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 seven segments are presented in Figs. 4b–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 Figs. 4e and f.

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 seven 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

### Table 3a Bit-rate requirements in shape representation where $N$ is the number of segments

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

The values in the parentheses indicate the generated peak distortion ($D_{\text{max}}$) in pel

$^a$CP encoded using the original OAVE technique [13]
the Lip object. Interestingly for the Fish shape, the DFLC techniques [11, 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 2.8 GHz Pentium-4 processor, with 512 MB 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 s, 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 uses 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

---

**Table 3b** Average bit-rate requirements in shape representation where N is the number of segments

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

The values in the parentheses indicate the generated peak distortion (D_max) in pel.

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

---

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BCGSE</td>
<td>3.01</td>
<td>7.01</td>
<td>2.32</td>
<td></td>
</tr>
<tr>
<td>SDCB</td>
<td>1.72</td>
<td>4.01</td>
<td>1.62</td>
<td></td>
</tr>
<tr>
<td>ACAF</td>
<td>4.02</td>
<td>9.01</td>
<td>3.05</td>
<td></td>
</tr>
<tr>
<td>SDCC</td>
<td>3.99</td>
<td>9.03</td>
<td>3.12</td>
<td></td>
</tr>
<tr>
<td>ASLM a</td>
<td>2.51</td>
<td>6.05</td>
<td>2.25</td>
<td></td>
</tr>
</tbody>
</table>

aASLM uses quadratic BC while all other algorithms use cubic BC.
distortion values, particularly when the shape contour exhibits sharp variations such as in the Lip object and Arabic character.

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. Figs. 5a and b, respectively, plot the computational-time (for coding only) against distortion and bit-rate against distortion (RD) results upon the Fish shape.

Fig. 5a 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 Fig. 5b, 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

<table>
<thead>
<tr>
<th>Shape</th>
<th>Admissible bit-rate R&lt;sub&gt;max&lt;/sub&gt; (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miss America (qcif)</td>
<td>N = 5 (205)  N = 13 (438)</td>
</tr>
<tr>
<td>Akiyo (qcif)</td>
<td>N = 6 (228)  N = 13 (435)</td>
</tr>
<tr>
<td>Stefan (sif)</td>
<td>N = 4 (229)  N = 10 (435)</td>
</tr>
</tbody>
</table>

The values in parentheses represent the respective utilised bit-rates

Finally, a series of experiments were conducted to appraise the performance of the non-linear 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<sub>max</sub> = 230 bits for Miss America, a maximum of five segments are required with 205 bits being utilised, whereas 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 BCs 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 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 E-OAVE. It has also developed a non-linear 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 Acknowledgment

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


