

# A DYNAMIC BEZIER CURVE MODEL

Ferdous Ahmed Sohel, Laurence S. Dooley, Gour C. Karmakar

Gippsland School of Computing and Information Technology  
Monash University, Churchill, Victoria – 3842, Australia.

E-mail: {Ferdous.Sohel, Laurence.Dooley, Gour.Karmakar}@infotech.monash.edu.au

## ABSTRACT

Bezier curves (BC) are fundamental to a wide range of applications from computer-aided design through to object shape descriptions and surface mapping. Since BC only consider global information with respect to their control points, this can lead to erroneous shape representations, though integrating local control point information minimises this error. This paper presents a new *Dynamic Bezier Curve (DBC)* model which combines both localised and global shape information by making a parametric shift of the BC points in the gap between the curve and its control polygon. The value of the shifting parameter is dynamically determined for a prescribed maximum distortion. DBC retains the kernel properties of the BC without increasing computational complexity order. The model's performance has been empirically evaluated on a number of arbitrary-shaped objects from geometric modelling to shape coding. Both qualitative and quantitative results confirm the improvement achieved compared with the classical BC representation.

**Keywords:** Bezier curve, globalised information, localised information, parametric shifting, smoothness, distortion.

## 1. INTRODUCTION

Bezier curves (BC) were independently developed by Casteljau and Bézier in the last century, and while their origin can be traced to the design of car body shapes in the automotive industry, their use is no longer confined to this field. Indeed, their robustness in curve and surface representation means they pervade many areas of multimedia technology including shape description of characters [1] and objects [2], active shape lip modelling [3], shape coding and error concealment for MPEG-4 coded objects [4] and surface mapping [5].

The classical BC is defined by a set of *control points* with the number and orientation of these points governing the shape of the curve. Bezier theory however, only considers global information [6] about the control points, since a particular BC point is produced by blending all the control points. As a result, there is often a large gap between the curve and its control polygon (CP) leading to errors in the shape approximation. Higher degree curves reduce this gap and provide more flexibility in shape representation. *Degree elevation* [7] has been exploited to form a curve with an increased number of control points though all of these, except the two end-points, must be

recalculated, so the computational overhead incurred in using higher-order BC is commensurately increased. To address this shortcoming, one alternative is to construct a *composite Bezier curve* [8] (CBC) using multiple BC segments. The calligraphic character outlining algorithms [1] and active lip-shape models [3] both exploit CBC, by detecting the *corner points* of a shape and then using consecutive pairs of points to form a shape segment. While suitable for regular shapes, these algorithms require more segments to accurately describe shapes which have more localised geometric features such as sharp edges and corners. Moreover, in such cases the control points are not normally located on the target shape, which can increase the image size and the descriptor length. This all means that while CBC serves the purpose of shape rendering, it is ineffectual for design purposes; hence the concept of *refinement* [8] has evolved, which comes in the form of *subdivision* and *refinement* techniques. When the control points of a BC are known, two sets of new control points, which are guaranteed closer to the curve, are calculated using methods such as *midpoint* [9] or *arbitrary subdivision* [10]. These algorithms increase the number of curve segments however and thereby the number of control points. Moreover, to ensure curve segments are smoothly conjoint, the number of subdivisions has always to be constrained.

While all the aforementioned algorithms minimise the gap between the BC and its CP up to a certain distortion level by increasing the number of control points, they also result in a higher coding and transmission cost to represent a particular shape. *Enhanced BC* models [11] address these issues, though they only decrease the gap to a limited extent and also do not provide a facility to dynamically select the size of the gap.

This paper in contrast, presents a novel *Dynamic Bezier Curve (DBC) model* that incorporates local information within the classical BC framework by shifting the curve points to new parametrically determined points situated between the BC point and the CP. The shifting parameter (SP) is dynamically determined to maintain a prescribed maximum distortion. It will be proven the DBC model retains the fundamental properties of the classical BC, while its performance as a generic shape descriptor is extensively analysed for a large number of arbitrary shapes using both quantitative and qualitative metrics, with results confirming its superiority over classical BC models.

The remainder of the paper is organised as follows: Section 2 provides a short overview of classical BC concepts highlighting in particular, its limitation in considering only global information. Section 3 introduces the theory underpinning the new DBC model, including the strategy to determine the shifting parameter, as well as proofs showing that core Bezier

properties are retained and the computational complexity is exactly the same as the classical BC. Section 4 presents experimental results illustrating the consistently improved performance of the DBC model relative to the classical BC, with some conclusions drawn in Section 5.

## 2. OVERVIEW OF THE CLASSICAL BEZIER CURVE

The classical BC is a recursive linear weighted subdivision of the edges of the generated polygon starting with a set of points to form the initial (control) polygon and ending when the final point is generated for a particular weight  $t$ . The set of  $N+1$  starting points is referred to as the *control points* which govern the shape of the BC of *degree*  $N$ . The polygon connecting the control points is called the *control polygon (CP)*. The matrix form [6] of the classical BC for an ordered set of control points  $V = \{v_0, v_1, \dots, v_N\}$  is defined as:-

$$v(t) = Pow^N(t) * Bez^N * V^T \quad (1)$$

where  $v(t)$  is the Bezier curve point for a particular  $t$ ,

$Pow^N(t)$  represents the power basis  $(1, t, t^2, \dots, t^N)$  and the

$ij^{th}$  term of matrix  $Bez^N$  is found from  $m_{ij} = (-1)^{j-i} \binom{N}{i} \binom{N}{j}$ .

$t$  is a parametric operator which defines the location of the curve point, with the number of curve points depending on the number of  $t$  values. From (1) it is evident that a particular BC point is generated by blending all control points, which implies the BC only considers the global information of a shape and yields a gap between the curve and its CP. Figure 1 shows an example of a quadratic BC using control points  $p_0, p_1, p_2$  to illustrate the large gap that occurs between the curve and its CP. For  $t = 0.5$ , the inner part of  $\Delta p_1 B$  is never reached and curve point  $C$  is generated on line  $AB$ . This highlights the significant shape distortion that can occur due to the classical BC only considering global information. In order to reduce this error, a novel strategy incorporating local information about the CP is now presented.

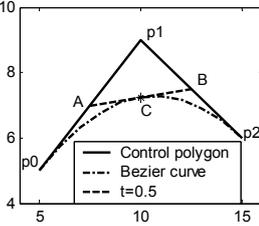


Figure 1: A quadratic Bezier curve illustrating the gap.

## 3. DYNAMIC BEZIER CURVE MODEL

To minimise the gap between the BC and its CP, the area bounded between them has to be accessed by shifting a generated BC point. Whenever a BC point is generated, one CP edge will be the minimum distance from it, and it is this edge that analytically exerts the maximum influence on that particular curve point. The DBC point is obtained by making a parametric shift of the BC point towards this particular CP edge. The *shifting parameter (SP)* plays a vital role as when SP is large, the curve approaches the CP (since local information is dominant over global information for the control points) resulting in a short distance from the CP, though in this case the curve loses smoothness. Conversely, when SP is small it

possesses maximum smoothness; however the distance between the BC and CP is now close to the maximum since global information dominates. Thus, the choice of SP provides a trade-off between the distance minimisation and level of smoothness. A strategy to optimise SP for an admissible distortion is discussed in Section 3.1, before which a summary of the mathematical formulation of the DBC is provided:-

For a particular  $t$ , if the edge having the shortest distance from BC point  $(P, Q)$  has end points  $(x_1, y_1)$  and  $(x_2, y_2)$ , which are also control points, then the intersection point  $(I, J)$  between this edge and the perpendicular line on this edge through the BC point is given by:-

$$I = \frac{A(A \times P + B \times Q) + B(B \times X - A \times Y)}{A^2 + B^2} \text{ and } J = \frac{B(A \times P + B \times Q) - A(B \times X - A \times Y)}{A^2 + B^2} \quad (2)$$

where  $A = x_1 - x_2$  and  $B = y_1 - y_2$  and  $(X, Y)$  is either  $(x_1, y_1)$  or  $(x_2, y_2)$ . If  $m$  is the shifting parameter, i.e.,  $m : (1 - m)$  is the shifting ratio at the curve point between  $(P, Q)$  and  $(I, J)$ , the DBC point  $(R, S)$  is calculated by:-

$$R = m * I + (1 - m) * P \text{ and } S = m * J + (1 - m) * Q. \quad (3)$$

which can be formalised in matrix form as:-

$$\begin{bmatrix} I \\ J \end{bmatrix}_t = \begin{bmatrix} \frac{A}{A^2 + B^2} & \frac{B}{A^2 + B^2} \\ \frac{B}{A^2 + B^2} & \frac{-A}{A^2 + B^2} \end{bmatrix}_t * \begin{bmatrix} P & Q \\ -Y & X \end{bmatrix}_t * \begin{bmatrix} A \\ B \end{bmatrix}_t \quad (4)$$

$$DBC(t) = [R \ S]_t = [m \ (1 - m)] * \begin{bmatrix} I & J \\ P & Q \end{bmatrix}_t, \quad 0 \leq m \leq 1 \quad (5)$$

### 3.1. Optimising the value of shifting parameter $m$

To maintain the maximum admissible distance between the curve and the CP, as well as preserving smoothness, the value of  $m$  must be as small as possible. The reason for this is that increasing  $m$  decreases the smoothness of the curve. The *Lagrangian optimisation* method [12] is used to find the optimal value of  $m$  for a prescribed maximum distance. If  $I(m)$  and  $D(m)$ , respectively represent an identity function and distance between the curve and the CP at a particular  $m$ , for any  $\lambda \geq 0$ , an unconstrained problem for the optimal solution  $m^*(\lambda)$  using the *generalised Lagrangian multiplier* [12] can be formulated as:-

$$\min_{m \in [0, 1.0]} (I(m) + \lambda \times D(m)) \quad (6)$$

The optimal solution to this unconstrained problem is also the optimal solution to the constrained problem:-

$$\min_{m \in [0, 1.0]} I(m) \text{ subject to: } D(m) \leq D_{adm} \quad (7)$$

where  $D_{adm}$  is the maximum admissible distance between the curve and the CP. Since  $D(m^*(\lambda))$  is a non-increasing function of  $\lambda$  [13], the *bisection method* [13] is used to find the optimal value of  $\lambda$ . It is noteworthy that the admissible distance  $D_{adm}$

is bounded by  $0 \leq D_{adm} \leq L$ , where  $L = \frac{\lfloor N/2 \rfloor \times \lceil N/2 \rceil}{2 * N} \|\Delta v_2\|_\infty$

with  $\|\Delta v_2\|_\infty$  being the maximum of the  $i^{th}$ -centred second difference of the coefficient sequence  $v_i, i = 0, \dots, N$ ; as [14] proves the maximum distance between the BC and its CP is  $L$ .

The complete DBC algorithm is presented in Algorithm 1:-

**Algorithm 1:** The Dynamic Bezier Curve (DBC) algorithm.

1. Find the optimal value of  $m$  for an admissible  $D_{adm}$ .
2. For each values of  $t$
3. Find the Bezier point  $(P, Q)$  using (1).
4. Determine the minimum distance edge  $(x_1, y_1, x_2, y_2)$  from  $(P, Q)$ .
5. Calculate DBC point  $(R, S)$  using (4) and (5).

As the foundations of the DBC model are underpinned by classical BC theory, the core properties [6] of the BC are preserved in DBC. Lemma 1, 2 and 3 examine some of these properties while Lemma 4 analyses the computational complexity of DBC.

**Lemma 1:** *End point interpolation:* The DBC will always pass through its first and last control points.

**Proof:** Bezier curve interpolates its end points [6] for the start ( $t=0$ ) and end ( $t=1$ ) control points, i.e.,  $(P, Q)$  is  $(x_1, y_1)$  or  $(x_2, y_2)$  at  $t=0$  or  $t=1$  respectively. For DBC at both  $t=0$  and  $t=1$ , from (4) and (5)  $R=I=P$  and  $S=J=Q$  so DBC also interpolates the end points.

**Lemma 2:** *Convex Hull Property:* The DBC always lies within the convex hull of its control points.

**Proof:** The BC curve always lie within the convex hull of its CP [6] while the DBC points lie in the area between the CP and the BC inclusive, so the DBC lies within the convex hull of its CP. When  $m \rightarrow 0$  DBC approaches the BC, while for  $m=0$  it coincides with the BC, while when  $m \rightarrow 1$  DBC approaches the CP. For  $m=1$ , it coincides with the CP when the number of  $t$  values tends to  $\infty$ .

**Lemma 3:** *Linear Precision:* When all control points are located on a straight line, the DBC will also be a straight line.

**Proof:** BC maintains the property of linear precision [6]. In such cases,  $I=P$  and  $J=Q$  and hence  $R=I=P$  and  $S=J=Q$  so DBC maintains the linear precision property.

**Lemma 4:** *Computational analysis:* The order of computational complexity for DBC is exactly the same as the classical BC.

**Proof:** From the definition of the classical BC in (1), a  $N$ -degree curve has a computational complexity  $O(N^2)$ . The DBC model has the same order of complexity, because Step 3 of Algorithm 1 takes  $O(N^2)$  since it is simply BC point generation; Step 4 takes  $O(N)$  computational time as it is a linear search of distances; and Step 5 has a constant time requirement because the matrices in (4) and (5) have constant dimensions irrespective of the degree of the curve. Also, since the dimensions of these matrices are small compared with the usual value of  $N$ , the multiplications in (4) and (5) incur negligible time with respect to (1), so the complexity of Algorithm 1 for Steps 3 to 5 is  $O(N^2)$ . To determine the optimal value of  $m$  in Step 1 requires  $O(N^2)$  time as Steps 3 to 5 are also used in this calculation, hence, the overall complexity is  $O(N^2)$ .

#### 4. EXPERIMENTAL RESULTS AND ANALYSIS

For presentational clarity, throughout this section, the black solid line represents the original shape, while the dashed blue and dot/dash-dot red lines denote the shape approximated by the classical BC and DBC respectively. For quantitative analysis, the maximum  $D_{max}$  (*class one – peak*) and overall  $D_{overall}$  (*class two – mean squared*) distortions were applied [13].

The performance of DBC model is firstly compared with the classical BC from the perspective of curve representation by maintaining different admissible distances,  $D_{adm}$ , between the curve and the CP, using a set of hypothetical control points. The results for some real-world test shapes are then analysed. From the plots in Figure 2, it is clear the lower the admissible distance  $D_{adm}$  (e.g.  $D_{adm}=0.3$ ), the closer DBC is to CP, while the higher the value of  $D_{adm}$  (e.g.  $D_{adm}=0.7$ ), the further away DBC is from CP.

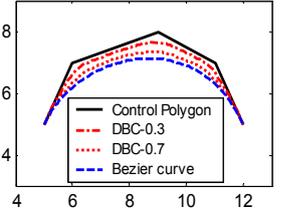


Figure 2: BC and DBC for different admissible distances.

For comparative purposes, both BC and DBC used the same segments in the respective referenced applications. The control points for DBC were determined by using the Beus corner detection method [15], since it is the most efficient curvature measurement and shape analysis method, as discussed in [16], and also it guarantees the corner (i.e., control) points always lie on the actual shape contour. For BC the control points were determined in accordance with the corresponding algorithms.

A series of experiments was conducted to model the *kettle* shape shown in Figure 3. The results for BC and DBC with a lowest achievable distortion of  $1pel$  for a prescribed number of segments are presented in Figures 3(a) and (b) respectively. It is visually evident that DBC provides a better shape approximation than BC, which is confirmed by the quantitative results in Table 1. The maximum and overall distortions of BC were  $3.5pel$  and  $3.1pel^2$  respectively, while the corresponding values for DBC were  $1pel$  and  $0.8pel^2$  respectively. Note it is feasible to model the shape using DBC at a lower distortion than  $1pel$  by increasing the number of segments.

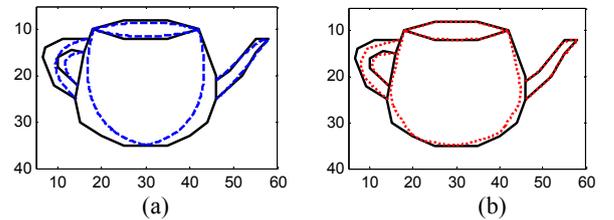


Figure 3: Shape modelling for *kettle* using a) BC and b) DBC.

The next set of experiments was performed upon the *lip* shape in Figure 4(a), which was also used in [3]. This shape was defined using five BC segments (s1-s5) where the control points  $c_i$  were defined outside the shape. The results produced by [3] using BC and the DBC for  $D_{adm}=0.5pel$  are presented in Figures 4(b) and (c) respectively. If the approximated shape in Figure 4(c) is compared with Figure 4(b), it manifests that DBC has

outperformed BC, with the numerical results in Table 1 reinforcing this finding. For BC, since the control points are outside of the shape, the image size has increased (Figure 4(b)), which also means a larger shape descriptor as it depends on the distance between two consecutive control points. For example, in case of the *lip*, using DBC the maximum distance between two consecutive control points is 15 pel, while using BC this distance is 28.3 pel. Moreover, for the BC, it is not possible for the user to stipulate a prescribed accuracy for approximation, whereas DBC provides the freedom to select the accuracy without changing the image size, by maintaining all the control points on the shape.

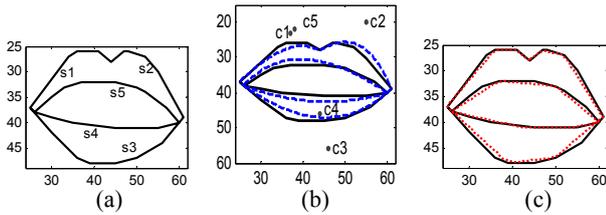


Figure 4: Shape modelling for *lip* using b) BC and c) DBC.

**Table 1: Summarised results for shape descriptions.**

Test Shape ( $D_{adm}$ )	Class one distortion (pel)		Class two distortion (pel <sup>2</sup> )		Whether the control pts lie on the shape		Max distance between consec control pts (pel)	
	BC	DBC	BC	DBC	BC	DBC	BC	DBC
Kettle (1)	3.5	1	3.1	0.8	Yes	Yes	18	18
Lip (0.5)	1.0	0.5	0.7	0.12	No	Yes	28.3	15
Arabic (1)	2	1	0.1	0.04	No	Yes	30	27

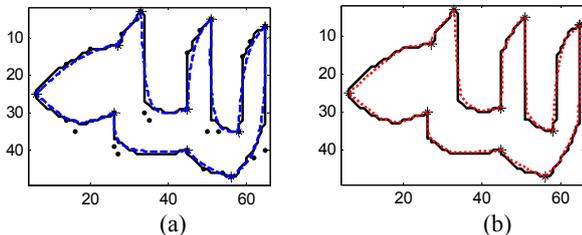


Figure 5: Shape modelling for Arabic-character by a) BC and b) DBC with  $D_{adm} = 1pel$ .

The final experiments were conducted upon one of the Arabic characters in [1] and the corresponding approximation results are shown in Figures 5(a) and (b). While [1] produced an optimal set of control points, DBC still provided an improved shape approximation. In [1], most control points were outside the shape and it was also not adaptive to user requirements. This limitation together with the quantitative performance results in Table 1, once again confirm the superiority of DBC.

## 5. CONCLUSIONS

While Bezier curves are well established tools for a range of applications, their main drawback is that they do not incorporate local information. This paper has focused upon integrating localised information into the classical BC framework. A new *Dynamic Bezier Curve* (DBC) model has been presented which includes a dynamic mechanism for handling the local and global information of the control points by determining the shifting parameter for a prescribed peak distortion through optimisation

process. The DBC retains the core properties of the classical BC, and the qualitative and quantitative results using a variety of shapes, used extensively in the literature, have clearly demonstrated that DBC model exhibits considerable improvement over BC in terms of shape distortion performance, while keeping the same order of computational complexity.

## 6. REFERENCES

- [1] M. Sarfraz, and M.A. Khan, "Automatic outline capture of Arabic fonts," Information Sciences, Elsevier Science Inc., pp. 269-281, 2002.
- [2] L. Cinque, S. Levialdi, and A. Malizia, "Shape description using cubic polynomial Bezier curves," Pattern Recognition Letters, Elsevier Science Inc., pp. 821-828, 1998.
- [3] I. Shdaifat, R. Grigat, and D. Langmann, "Active shape lip modeling," Proceedings of International Conference on Image Processing, pp. 875-878, Sept., 2003.
- [4] L.D. Soares, and F. Pereira, "Spatial shape error concealment for object-based image and video coding," IEEE Trans. on Image Processing, vol. 13, no. 4, pp. 586-599, 2004.
- [5] R. Zhang, and G. Wang, "Some estimates of the height of rational Bernstein-Bezier triangular surfaces," Proceedings of the Geometric Modeling and Processing, pp.79-84, 2004.
- [6] F.S. Hill Jr., *Computer Graphics*, Prentice Hall, Englewood Cliffs, 1990.
- [7] A.R. Forrest, "Interactive Interpolation and Approximation by Bezier Polynomials," Computer Journal 15(1), 1972.
- [8] R.H. Bartels, J.C. Beatty, and B.A. Barsky, *An Introduction to Splines for use in Computer Graphics & Geometric Modeling*, Morgan Kaufmann Publishers, 1987.
- [9] J.M. Lane, and R.F. Riesenfeld, "A theoretical development for the computer generation of piecewise polynomial surfaces," IEEE Trans. on PAMI, vol. 2, no.1, pp. 35-46, 1980.
- [10] M. Hosaka, and F. Kimura, "A theory and method for free form shape construction," Journal of Info. Proc., vol.3, no.3, pp. 140-151, 1980.
- [11] F.A. Soheli, G.C. Karmakar, L.S. Dooley, and J. Arkinstall, "Enhanced Bezier curve models incorporating local information," IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP 05), vol. IV, pp.253-256, ISBN: 0-7803-8875-5, Mar. 2005.
- [12] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," Operational Research, 11: 399-417, 1963.
- [13] G.M. Schuster, and A.K. Katsaggelos, *Rate-Distortion Based Video Compression-Optimal Video Frame Compression and Object Boundary Encoding*, Kluwer Academic Publishers, 1997.
- [14] D. Narin, J. Peters, and D. Lutterkort, "Sharp, quantitative bounds on the distance between a polynomial piece and its Bezier control polygon," Computer Aided Geometric Design, Elsevier Science Inc., pp. 613-631, 1999.
- [15] H. L. Beus, "An improved corner detection algorithm based on chain coded plane curves," Pattern Recognition, vol. 20, no. 3, pp. 291-296, 1987.
- [16] H.C. Liu, and M.D. Srinath, "Corner point detection from chain-code," Pattern Recognition, pp. 51-68, 1990.