ENHANCED BEZIER CURVE MODELS INCORPORATING LOCAL INFORMATION

Ferdous Ahmed Sohel, Gour C Karmakar, Laurence S Dooley and ***John Arkinstall****
{Ferdous.Sohel, Gour.Karmakar, Laurence.Dooley}@infotech.monash.edu.au
Gippsland School of Computing and Information Technology
*John.Arkinstall@sci. monash.edu.au, Gippsland School of Applied Sciences & Engineering
Monash University, Churchill, Victoria 3842, Australia.

ABSTRACT

The Bezier curve is fundamental to many challenging and practical applications, ranging from computer aided geometric design and postscript font representations through to generic object shape descriptors and surface representation. A drawback of the Bezier curve however, is that it only considers global information about the control points, so there is often a large gap between the curve and its control polygon, leading to considerable error in curve representations. To address this issue, this paper presents *enhanced Bezier curve* (EBC) models which seamlessly incorporate local information. The performance of the models is empirically evaluated upon a number of natural and synthetic objects having arbitrary shape and both qualitative and quantitative results confirm the superiority of both EBC models in comparison with the classical Bezier curve representation, with no increase in the order of computational complexity.

Index Terms – Bezier curve, Global information, Control Polygon, Local information, Centre of Gravity.

1. INTRODUCTION

Bezier curves were developed by Casteljau [1] and Bézier [2], and have been applied to many computer-aided design (CAD) applications. While their origin can be traced back to the design of car body shapes, their usage is no longer confined to this field. Indeed, their robustness in curve and surface representation means many variations have evolved, including recently into areas such as shape description of characters [3-4] and objects [5], shape error concealment for MPEG-4 objects [6] and surface mapping [7-8].

A Bezier curve is defined in terms of a set of control points, though it only considers global information [9] and calculates the curve points in a linear recursive approach starting with the edges of the control polygon. Frequently, there is a large gap between the Bezier curve and its control polygon, which restricts the maximum length of a curve segment. To represent complex curves, more curve segments and hence more control points are required. Moreover, to approximate a given shape most control points have to be defined outside the original shape which will not necessarily be inside the coordinate system, thereby increasing the computational overhead of the Bezier curve for many applications. A higher-degree Bezier curve obviously provides a better shape representation. *Degree elevation* [10] has been applied to form a curve with an increased number of control points, though all these, bar the end points, have to be relocated incurring significant computational cost. *Subdivision* and *refinement* techniques have been introduced to minimize the gap between the Bezier curve and its control polygon. When the control points are known, a set of new control points that are closer to the curve is defined using subdivision algorithms such as *midpoint* [11] or *generalized arbitrary Bezier* [12]. These algorithms however, increase the number of curve segments along with the number of points and to ensure the requisite conjoint curve segments, the number of subdivisions has also to be constrained.

All the aforementioned algorithms minimize the distance between the Bezier curve and its control polygon by increasing the number of control points. For multimedia communications this means a higher coding and transmission overhead to represent a particular shape. This paper introduces two *enhanced Bezier curve* (EBC) models which incorporate local information within the classical Bezier curve framework, and incur no increase in computational complexity. It is noteworthy that both models can be seamlessly integrated into all refinement algorithms including *degree elevation* and s*ubdivision*. It has also been shown that the EBC models retain all the central properties of the classical Bezier curve. The performance of EBC as a generic shape descriptor for a number of different natural shapes as well as different polygons was analyzed using both objective and subjective metrics, with results clearly confirming its superiority compared with the original Bezier curve [1-2].

The remainder of the paper is organized as follows: Section 2 provides a short overview of the classical Bezier curve, while Section 3 discusses the theoretical basis of the new EBC models along with proofs that all the key properties of the Bezier curve are retained. Section 4 presents experimental results confirming the superior performance of EBC models relative to the original Bezier curve, with some conclusions given in Section 5.

2. OVERVIEW OF THE CLASSICAL BEZIER CURVE

The Bezier curve is a recursive linear weighted subdivision of the edges of the generated polygon starting with a set of points to form the initial polygon and ends iteration when the final point is generated. The set of $N+1$ starting points is referred to as the *control points* which govern the characteristics of the Bezier curve of degree N . The polygon connecting the control points is called the *control polygon*. The Casteljau form of the Bezier curve for an ordered set of control points $P=\{p_0, p_1, \ldots, p_N\}$ is defined as:-

$$p_i^r(u)=\begin{cases} i^{th} \text{ member of } P, p_i; & \text{if } r=0; \\ (1-u)p_i^{r-1}(u)+up_{i+1}^{r-1}(u); & r=1,\cdots,N; \quad i=0,\cdots,N-r; \; 0\le u\le 1 \end{cases} \tag{1}$$

where u is the weight of subdivision and determines the number of points on the Bezier curve. The *final* generation $p_0^N(u)$ is called the Bezier curve of P .

Figure 1 shows the quadratic Bezier curve produced using the control points p_0, p_1, p_2. There is a large gap between the Bezier curve approximation and its control polygon which represents a significant shape distortion due to the Bezier curve considering only global information.

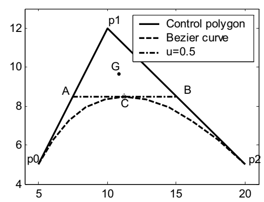
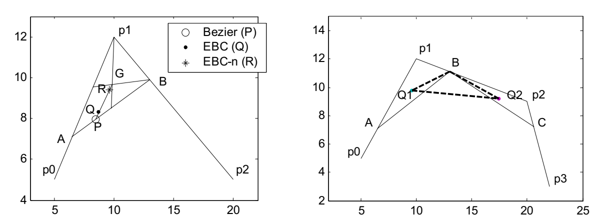


Figure 1: A quadratic Bezier curve illustrating the gap.

For $u = 0.5$, the inner part of ΔAp_1B is never reached and point C is generated on line AB. To reduce this gap, a number of refinement techniques have been proposed [10-12], which require an increased number of control points. To minimize the gap between the curve and its control polygon without increasing the number of control points, it is required to move the Bezier point inside ΔAp_1B. The next section introduces a novel strategy to achieve this objective.

3. NEW ENHANCED BEZIER CURVE (EBC) MODELS INCORPORATING LOCAL INFORMATION

In this section, the quadratic Bezier curve will firstly be presented before being generalized for any arbitrary degree. To minimize the gap between the Bezier curve and its control polygon, the centre of gravity (CoG) G of ΔAp_1B in Figure 1 is exploited in shifting a specific point generated by the original Bezier curve. If this point for a particular u is moved directly to the CoG, two major problems arise:- 1) the generated curve will not be smooth as all the generated points are confined to a small region of the curve since control point p_1 is common and so has a significant influence in all triangles; 2) End-point interpolation, which is one of the important properties of the Bezier curve is not satisfied, since for terminal values $u = 0$ *or* $u = 1$, the CoGs can never be end control points.



(a) (b)

Figure 2: Enhanced Bezier Curve for a) Quadratic; b) Cubic.

To obtain a smooth curve, all generated points must be well distributed over the entire curve. This is achieved by generating a point using a suitably weighted linear interpolation between the Bezier curve point and its CoG. If $u : 1-u$ is used as the interpolation weighting factor, the end-point interpolation property for the last control point will not be upheld for the reason discussed above, since the generated point is shifted to the CoG of the corresponding triangle. However, as will be proven in Lemma 1, the ratio $u(1-u) : 1-u(1-u)$ for a Bezier curve point and its CoG concomitantly guarantees the end-point interpolation criteria and ensures a smooth curve. This technique of shifting a Bezier curve point by using the above ratio is called the *enhanced Bezier curve* (EBC) and is pictorially depicted in Figure 2(a), where P is the Bezier curve point for $u = 0.3$ and G is the CoG of ΔAp_1B. In the new EBC model, P moves to any point along the line PG. The actual shifted amount Q is such that it segments line PG so $PQ : QG = u(1-u) : 1-u(1-u) = 0.21 : 0.79$, and the quadratic EBC is formulated as:-

$$p(u) = \tfrac{1}{3}(1-u)^2(3u^2 - 2u + 3)p_0 + \tfrac{2}{3}u(1-u)(3u^2 - 3u + 4)p_1 + \tfrac{1}{3}u^2(3u^2 - 4u + 4)p_2 \quad ; \quad 0 \le u \le 1 \qquad (2)$$

where $\{p_0, p_1, p_2\}$ is the set of control points.

The cubic EBC is explained by Figure 2(b). Points $Q1$ and $Q2$ are generated using the quadratic Bezier described above for control point sets $\{p_0, p_1, p_2\}$ and $\{p_1, p_2, p_3\}$ respectively. A new quadratic control polygon is then formed with $\{Q_1, B, Q_2\}$, where B is the weighted (u) interpolation of successive initial control points p_1 and p_2, and the final curve point is generated by quadratic EBC with control points $\{Q_1, B, Q_2\}$.

The quadratic EBC can be extended for an arbitrary degree N by using successively generated quadratic EBC points along with polygon point to form another quadratic EBC until it converges to a single point. It can be formulated recursively as:-

$$p_i^r(u) = \begin{cases} \tfrac{1}{3}(1-u)^2(3u^2-2u+3)p_i + \tfrac{2}{3}u(1-u)(3u^2-3u+4)p_{i+1} \\ \quad + \tfrac{1}{3}u^2(3u^2-4u+4)p_{i+2}\,; \qquad if \;\; r=0 \\ \tfrac{1}{3}(1-u)^2(3u^2-2u+3)p_i^{r-1}(u) + \tfrac{2}{3}u(1-u)(3u^2-3u+4)s_i^r(u) + \\ \tfrac{1}{3}u^2(3u^2-4u+4)p_{i+1}^{r-1}(u); \;\; 1 \le r \le N-2; 0 \le i \le N-r-2; 0 \le u \le 1 \end{cases} \qquad (3)$$

$$s_i^r(u) = \begin{cases} u \cdot p_{i+2} + (1-u)\cdot p_{i+1}; & if \;\; r=1 \\ p_{i+2}; & if \;\; r=2 \\ s_{i+1}^{r-2}(u); & else \end{cases} \qquad (4)$$

The first and last control points of the required three control points to form a quadratic curve are chosen from the generated EBC points in the previous iteration and the middle control point $s_i^r(u)$ is selected from either the initial control points or the interpolation point by (4), so $p_0^{N-2}(u)$ is the resulting EBC.

Due to the low value of $u(1-u)$ in EBC, the displacement of a Bezier curve point towards the CoG is small. To ensure a larger displacement, i.e. reduce the gap between the generated curve and control polygon, $u(1-u)$ is normalized as follows:-

$$\max_{0 \le u_j \le 1, \;\; u_{j+1}-u_j=\Delta u} \{u_j(1-u_j)\} = 0.25 \;\; for \; u = 0.5 \qquad (5)$$

So the normalized ratio becomes $u(1-u)/0.25 : 1-u(1-u)/0.25$. For a particular value $u = 0.5$, P is shifted to the CoG of the triangle, so it ensures a smooth curve as the generated points are well distributed over the entire curve and also reduce the gap between the curve and the control polygon. Using the normalized parameter, the EBC model is referred to as the EBC-*n*. For $u = 0.3$, R is the EBC-*n* point shown in Figure 2(a), where the ratio is $PR : RG = 0.84 : 0.16$. Applying the same rationale as in the EBC, the generic EBC-*n* form can be expressed as:-

$$p_i^r(u) = \begin{cases} \tfrac{1}{0.75}(1-u)^2(3u^2-2u+0.75)p_i + \tfrac{2}{.75}u(1-u)(3u^2-3u+1.75)p_{i+1} \\ \quad + \tfrac{1}{0.75}u^2(3u^2-4u+1.75)p_{i+2}; \qquad for \;\; r=0; \\ \tfrac{1}{0.75}(1-u)^2(3u^2-2u+0.75)p_i^{r-1}(u) + \tfrac{2}{0.75}u(1-u)(3u^2-3u+1.75)s_i^r(u) + \\ \tfrac{1}{0.75}u^2(3u^2-4u+1.75)p_{i+1}^{r-1}(u); \;\; 1 \le r \le N-2; 0 \le i \le N-r-2; 0 \le u \le 1 \end{cases} \qquad (6)$$

All other conditions in (6) are the same as in (3).

As the foundations of both EBC models are underpinned by classical Bezier curve theory, all properties [9] are preserved. The following examines some of these properties, where without loss of generality, all proofs are provided for the EBC model.

Lemma 1: End point interpolation: The EBC always passes through its first and last control points.

Proof: Any Bezier curve interpolates its end points [9] for the starting ($u=0$) and end ($u=1$) control points. EBC makes a parametric shift of the original Bezier curve point towards the CoG by the ratio $u(1-u):1-u(1-u)$. For both $u=0$ and $u=1$, $u(1-u):1-u(1-u)=0:1$, which means the endpoints will not be shifted in EBC. This is evident in (2) and (3) i.e. $p(0)=p_0$ and $p(1)=p_N$.

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

Proof: Suppose a curve is defined as $p(u)=\sum_{0\le k\le N}\alpha_k(u)p_k;\ \alpha_k\ge 0$ where p_k is the k-th control point. If $\sum_{0\le k\le N}\alpha_k(u)=1;\ \forall u,$ the curve $p(u)$ lies within its convex hull [9].

EBC curves in (3) can be written as $p(u)=\sum_{0\le k\le N}\alpha_k(u)p_k;\ \forall u$ and using (2) it can be shown that $\sum_{0\le k\le 2}\alpha_k(u)=1;\ \forall u;\forall r\,.\ s_i^r(u)$ always lies on the control polygon. So any EBC point will lie within the corresponding triangle and thus EBC lies within the convex hull of the control points.

Lemma 3: Affine Invariance: EBC curve is invariant under affine transformations.

Proof: A curve is affine invariant if the curve drawn with affine transformed control points is the same as the entire affine transformed curve with the same parameters i.e.

$\sum_{k=0}^{N}(p_kR+t)\alpha_k(u)=\sum_{k=0}^{N}p_k\alpha_k(u)R+t$ where R is a transformation matrix and t is an offset vector [9].

EBC with affine transformed control points can be written as,

$$\sum_{k=0}^{N}(p_kR+t)\alpha_k(u)=\sum_{k=0}^{N}p_kR\cdot\alpha_k(u)+\sum_{k=0}^{N}t\alpha_k(u)=\sum_{k=0}^{N}p_k\alpha_k(u)R+t$$

by Lemma 2. Therefore EBC is affine invariant.

Lemma 4: Linear Precision: When all the control points are on a straight line, EBC will be a straight line.

Proof: By Lemma 2, EBC lies within the convex hull of its control points. Therefore, when all the control points are on a straight line, EBC will be a straight line.

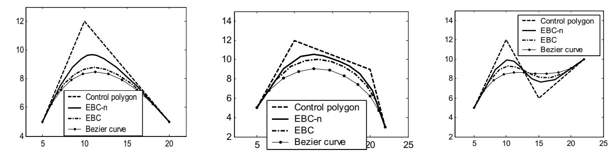
All of the proofs for EBC-*n* can be done in the same way as EBC.

Computational complexity analysis: The EBC models have the same order of complexity as the original Bezier curve, since in (3), for an N degree curve, EBC takes $N-2$ iterations to find the final curve point for each value of u, while the original Bezier (1) requires N iterations, so the computational order in both cases is $O(N)$ iteration.

4. EXPERIMENTAL RESULTS

In this section, the performance of the EBC models is firstly compared with the original Bezier curve from the perspective of curve representation, by using some hypothetical control point sets of different degree and orientation, before analyzing the results on real-world natural shapes.

Figure 3 shows a comparative study among the original Bezier curve, EBC and EBC-*n* for various degrees and orientations. EBC-*n* is always the closest to the control polygon, followed by EBC and then the Bezier curve. It reflects the fact that both EBC and EBC-*n* integrate local information concerning each control point in addition to the inherent global information of the Bezier curve.



(a) (b) (c)

Figure 3: Curves of different degree and orientation; a) Quadratic; b) Cubic; c) Cubic curves in a different orientation.

Another experiment was conducted to illustrate the potential of EBC and EBC-*n* using the *midpoint subdivision* algorithm [13]. The results are shown in Figure 4. EBC and EBC-*n* curves were drawn using the resultant control points generated by [13]. It is evident that both EBC models generated better curves than the original Bezier curve using the same subdivided control points set.

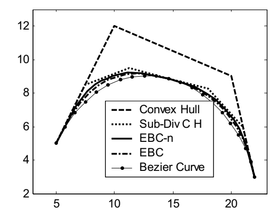
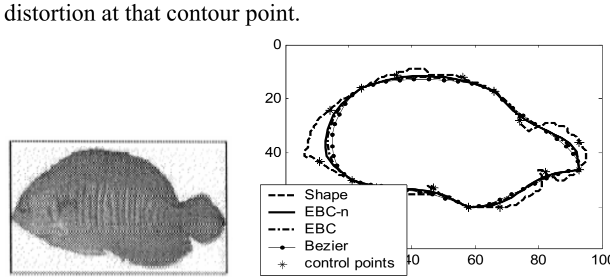


Figure 4: Curves using Bezier subdivisions.

Cubic Bezier curve had been used for shape description [5]. It used *a priori* number of curve segments (segment rate-SR) having an equal number of contour points each to describe a particular shape. The control points for a segment approximating the shape $V=\{v_0,v_1,\ldots,v_{M-1}\}$ between v_i to v_{i+m} (where $m=\frac{M}{SR}$) are:-

$$p_0=v_i;\ p_1=v_{i+\lceil\frac{m}{4}\rceil};\ p_2=v_{i+\lceil\frac{3\times m}{4}\rceil};\ p_3=v_{i+m} \quad (7)$$

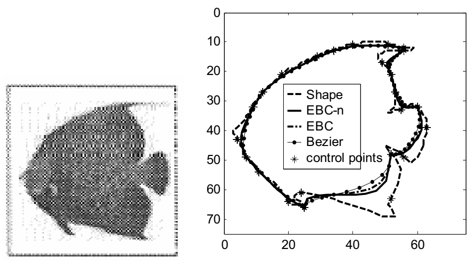
In the experiments same control points generated by (7) were used for the Bezier curve, EBC and EBC-*n* for two different natural images shown in [5]. The minimum gap between a point on the contour and the approximated shape represents the shape distortion at that contour point.



(a) (b)

Figure 5: a) Fish image 1; b) Shape described by 5 segments.

The popular and widely used *shape distortion measurement* algorithms [14] were used for numerical analysis. In Figure 5, shape descriptions of an object (fish 1) are shown for a fixed number of segments (SR=5). The Bezier curve had a maximum distortion of 9.25 pel for the head portion of the object, while EBC and EBC-*n* produced a maximum distortion of 7.8 and 7 pel respectively. Considering the entire object, EBC-*n* provided the best shape description, while Bezier curve was the worst, which is confirmed by the numerical results in Table 1, for the maximum and overall average (Avg) distortion [14] values, for various segment numbers. The results revealed that EBC-*n* provided better performance (lower distortion) even with a smaller number of curve segments. For example, the maximum and overall distortions for the Bezier curve with 6 segments was 7.8 and 6.7 units respectively, while for 5 segments, it was 7.8 and 6.6 units for EBC and 7 and 5.4 units for EBC-*n* respectively. This again highlights the fact that both EBC and EBC-*n* considered local information in addition to the Bezier global information.



(a) (b)

Figure 6: a) Fish image 2; b) Shape described by 5 segments.

Another series of experiments was performed on the image in Figure 6(a), with the corresponding shape approximations shown in Figure 6(b) for 5 segments. The lower half of Table 1 presents the results for a range of different segment rates and both the visual and empirical results produced by the EBC and EBC-*n* models confirm the improved performance in comparison with the original Bezier curve.

Table 1: Distortion (max distortion is in pels; Avg distortion is in pel^2) in shape representation.

Fish		SR = 5		SR = 6		SR = 7		SR =10	
		max	Avg	max	Avg	max	Avg	max	Avg
1	BC	9.25	9.6	7.8	6.7	6.3	4	3.5	1.3
	EBC	7.8	6.6	6.5	5.7	5.5	3	3.2	0.9
	EBC-*n*	7	5.4	6	3.7	5	2.3	2.9	0.7
2	BC	7.6	6.4	7.0	5	4.7	3	4.5	1.8
	EBC	7.4	5	6.6	4	4.4	2.3	3.9	1.3
	EBC-*n*	7.2	4.2	6.2	3.16	4.2	2.1	3.5	1.2

A final experiment was performed to test the performance of EBC models for higher degree curves. Different control point sets were used for different degree curves; however, for a particular degree the same set was used for both EBC models and the original Bezier. The curves were closed by joining the first and last curve points and the total area covered by the curves used as the comparison metric in Table 2. EBC-*n* covered the maximum area for all curve degrees while the Bezier curve covered the least area, with the control polygon being the upper bound, so confirming that the EBC models more closely follow the control polygon than the original Bezier curve.

Table 2: Area coverage in pel^2 for each curve of different degree.

Degree of the curve	Control polygon	EBC-*n*	EBC	Bezier Curve
2	42.5	34	29.8	28
3	91.5	77.7	72.3	60.7
4	297.5	209.4	205.3	170.7
9	593	382	335	255
19	240	210	193	180

5. CONCLUSIONS

While the Bezier curve is a well established tool for a wide range of applications, its main drawback is that it does not consider local information. This paper has primarily focused upon bridging this hiatus by integrating local information into the classical Bezier curve framework. Two enhanced Bezier curve (EBC and EBC-*n)* models have been presented and mathematically proven that they retain the core properties of the original Bezier curve. The qualitative and quantitative results using different polygons and shapes also showed that both EBC models exhibited significant improvement over the classical Bezier curve in terms of shape distortion performance, while having the same order of the computational complexity.

6. REFERENCES

[1] P. de Casteljau, *Outillage Méthodes Calcul*, Citroën, 1959.

[2] P.E. Bézier, *Employ des Machines á Commande Numérique*, Mason et Cie, Paris, 1970. Translated by Forrest, A. R., and A.F. Pankhurst as P. Bézier, *Numerical Control– Mathematics and Applications*, Wiley, London, 1972.

[3] M. Sarfraz, and M.A. Khan, “Automatic Outline Capture of Arabic Fonts,” *Information Sciences,* Elsevier Science Inc., pp. 269-281, 2002.

[4] H.-M. Yang, J.-J. Lu, and H.-J. Lee, “A Bezier Curve-Based Approach to Shape Description for Chinese Calligraphy Characters,” Proceedings of Sixth International Conference on Document Analysis and Recognition, pp. 276-280, 2001.

[5] L. Cinque, S. Levialdi, and A. Malizia, “Shape Description using Cubic Polynomial Bezier Curves,” *Pattern Recognition Letters,* Elsevier Science Inc., pp. 821-828, 1998.

[6] L.D. Soares, and F. Pereira, “Spatial Shape Error Concealment for Object-Based Image and Video Coding,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 586-599, 2004.

[7] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic press, New York, 1988.

[8] 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.

[9] F.S. Hill Jr., *Computer Graphics*, Prentice Hall, Englewood Cliffs, 1990.

[10] L.M. Kocić, “Modification of Bézier Curves and Surfaces by Degree-Elevation Technique,” *Computer-Aided Design*, vol. 23, no. 10, pp. 692-699, 1991.

[11] J.M. Lane, and R.F. Riesenfeld, “A Theoretical Development for the Computer Generation of Piecewise Polynomial Surfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 2, no.1, pp. 35-46, 1980.

[12] M. Hosaka, and F. Kimura, “A Theory and Method for Free form Shape Construction,” *Journal of Information Processing, vol.3, no.3*, pp. 140-151, 1980.

[13] R.H. Bartels, J.C. Beatty, and B.A. Barsky, *An Introduction to Splines for use in Computer Graphics & Geometric Modeling*, Morgan Kaufmann Publishers, pp. 219-223, 1987.

[14] G.M. Schuster, and A.K. Katsaggelos, *Rate-Distortion Based Video Compression-Optimal Video Frame Compression and Object Boundary Encoding*, Kluwer Academic Publishers, pp.222-223, 1997.