Far East Journal of Mathematical Sciences (FJMS)
Volume 56, Number 2, 2011, Pages 201-212
Published Online: September 15, 2011
This paper is available online at http://pphmj.com/journals/fjms.htm

OFFSET APPROXIMATION BASED ON PRECISE REPRESENTATION OF SWEEPING CIRCLE

ZHAO HONGYAN

College of Fundamental Studies
Shanghai University of Engineering Science
Shanghai 201620, P. R. China
e-mail: zhaohy81@yahoo.com.cn

Abstract

In this paper, we present a new algorithm for planar offset approximation based on precise representation of the sweeping circle. The basic idea of such algorithms is to regard the unit normal of the original curve as a unit circular arc, to express it as a rational curve precisely, and to perform a reparameterization work to approximate the offset curve. Compared with previous algorithms, our new algorithm overcomes the shortcomings of most of them which could not offset circle segments precisely, or those which could but highly dependent on sample technique, or those whose error estimation was roughly given. Furthermore, the method yields C^1 continuous offset curve approximation with smaller numbers of segmented curves and control points for regular Bézier or rational Bézier curves, and so it is of much significance in terms of saving computing time, reducing the data storage and smoothing curves entirely.

1. Introduction

Offset curves/surfaces, also called *parallel curves*/*surfaces*, are defined as locus of the points which are at constant distance r along the normal from the original

2010 Mathematics Subject Classification: 65D17.

Keywords and phrases: CAD/CAM, offset, circle, convolution, Bézier/B-spline curves, rational curve.

Received May 23, 2011

curves/surfaces. Offsets are widely used in various CAD/CAM (Computer Aided Design and Computer Aided Manufacturing) areas such as tool path generation, 3D NC machining, solid modeling, and so on [1-5].

Given a planar parametric curve $\boldsymbol{C}(t) = (x(t), y(t))$, the offset curve with an offset radius r is defined by $\boldsymbol{C}_r(t) = \boldsymbol{C}(t) + r\boldsymbol{N}(t)$, where $\boldsymbol{N}(t) = (y'(t), -x'(t)) / \sqrt{x'^2(t) + y'^2(t)}$ is the unit normal of $\boldsymbol{C}(t)$. In general, the offset curve is not rational because of the square root function in the denominator of $\boldsymbol{N}(t)$, and so is hard to be applied in CAD system.

Although Farouki and Sakkalis [6] introduced *Pythagorean hodograph* (PH) curves, whose offsets are rational curves, they were not widely used due to less flexibility. So approximation techniques seem to be a more feasible solution to the planar curve offsetting.

Several research results on offset approximation have been reported such as translating methods [7-9], interpolation and fitting algorithms [10-12]. Cobb [7] translated each control point, whereas Tiller and Hanson [8] translated each edge of the control polygon to obtain a new control net for approximated offset curve. Unfortunately, Cobb [7] always underestimated the offset [13]. Tiller and Hanson [8] had a similar performance to that of Cobb [7] for offsetting high degree curves.

As an interpolation and fitting algorithm, Hoschek [10] suggested a least squares solution to determine the hodographs at the endpoints. Piegl and Tiller [11] approximated the offsets of NURBS curves/surfaces by interpolating sample points. Li and Hsu [12] used Legendre series. However, those interpolation and fitting algorithms all estimated the approximation error only at finite sample points, thus there is no guarantee that the exact error is bounded by the given tolerance.

Taking a slightly different approach, in 1996, Lee et al. [14] proposed a novel algorithm CAO, and developed another two methods MO and CAMO in [15, 16]. They considered the offset problem as a sweeping problem in which the centre of a circle with the radius r is moving along the given planar curve. The boundary of the sweep is obtained as an envelope curve of the swept circle, which is identical to the exact offset curve.

The method CAO [14] approximated the offset circle with piecewise quadratic polynomial Bézier curves. Ahn et al. [17] developed the idea by using conic approximation technique [18], and yielded a Bézier curve with the same degree as the original curve. However, both the methods could not offset arcs precisely, which are widely applied in engineering. The method MO and its variants LRC, SRC [15] used the exact rational representation of the swept circle, and then approximated the reparameterization function for offset approximation. The methods could offset arcs precisely. However, MO and SRC methods are highly dependent on sample techniques. So the approximation effect is quite unstable. Besides the final approximated offset could not be obtained until a program is run, which made the methods inconvenient for application. LRC used a simple linear transformation, which is usually improper for most approximations. In addition, the approximated offsets of CAO, MO and their variants are at most G^1 continuous.

To modify the shortcomings of previous offset approximation algorithms, in this paper, we introduce a new method based on circle representation. The basic idea is to view the locus of the unit normal of the original curve as a circular arc, represent it with a rational polynomial, and then to reparameterize its convolution with the original curve in a new way to approximate the offset. Our method could offset circles precisely and generate high quality approximation without using samples. The global error can also be controlled by an effective error bound. Experimental results show that our approach outperforms the previous methods in the error bound and the number of curve subdivision.

2. Circle Representation

In this section, we review the quadratic rational Bézier representation of circular arc with the central angle 2θ, where $0 < 2\theta < \pi/2$. The result will be used in the construction of the approximated offset curve in Section 3.

The quadratic rational Bézier curve with three control points $\boldsymbol{P}_0$, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ and three weights ω_0, ω_1 and ω_2 is defined by

$$\boldsymbol{Q}(s) = (x_Q(s),\ y_Q(s))$$

$$= \frac{(1-s)^2\omega_0\boldsymbol{P}_0 + 2(1-s)s\omega_1\boldsymbol{P}_1 + s^2\omega_2\boldsymbol{P}_2}{(1-s)^2\omega_0 + 2(1-s)s\omega_1 + s^2\omega_2}, \quad 0 \le s \le 1.$$

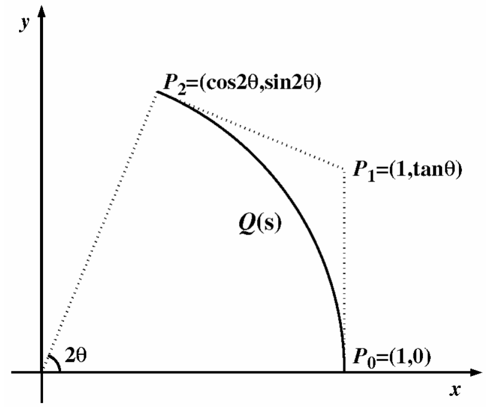


Figure 1. Circular arc with quadratic rational representation.

For the arc representation (see Figure 1), we assume that $\boldsymbol{P}_0 = (1, 0)$, $\boldsymbol{P}_1 = (1, \tan\theta)$, $\boldsymbol{P}_2 = (\cos 2\theta, \sin 2\theta)$, $\omega_0 = \omega_2 = 1$, $\omega_1 = \cos\theta$. The two components of $\boldsymbol{Q}(s)$ are

$$x_Q(s) = \frac{(1-s)^2 + 2(1-s)s\cos\theta + s^2\cos 2\theta}{(1-s)^2 + 2(1-s)s\cos\theta + s^2},$$

$$y_Q(s) = \frac{2(1-s)s\sin\theta + s^2\sin 2\theta}{(1-s)^2 + 2(1-s)s\cos\theta + s^2}.$$

3. Offset Curve Approximation

This section considers a regular parametric polynomial/rational curve $\boldsymbol{C}(t)$ of degree d. In the following, its offset approximation will be represented with piecewise rational curve segments. The approximation is based on the original curve $\boldsymbol{C}(t)$, the quadratic rational representation $\boldsymbol{Q}(s)$ of the circle, and the parameter transformation $s = s(t)$.

Consider a simple kind of original curve $\boldsymbol{C}(t) = (x(t), y(t))$ $(0 \le t \le 1)$ which satisfies the following conditions. $\boldsymbol{C}(t)$ is convex for $0 \le t \le 1$, and the direction angle range of the normal vectors of the curve is less than $\pi/2$. In addition, for the beginning point $\boldsymbol{C}(0)$, there should be $y(0) = x'(0) = 0$, $y'(0) > 0$. For those curves not satisfying the above conditions, a rotation can be done to bring them into accord.

Here we define a mapping $\alpha : t \in [0, 1] \mapsto \alpha(t) = \arctan\left(-\frac{x'(t)}{y'(t)}\right) \in [0, 2\pi]$, where $\alpha(t)$ is the directed angle from the positive half of X-axis to the normal $\boldsymbol{N}(t)$ at the original curve point $\boldsymbol{C}(t)$. Consequently, $\alpha(0) = 0$. Assume that $\alpha(1) = 2\theta$, or $y'(1)\sin 2\theta = -x'(1)\cos 2\theta$. Thus the locus of $\boldsymbol{N}(t)$ is a unit circular arc with the central angle 2θ. It can be represented by a quadratic rational Bézier curve $\boldsymbol{Q}(s)$ $(0 \le s \le 1)$ defined in Section 2. Then $\boldsymbol{C}_r(t)$ can be interpreted as a convolution of $\boldsymbol{C}(t)$ and $\boldsymbol{Q}(s)$ with a non-rational reparameterization $s = s(t)$.

We may approximate the reparameterization by a rational polynomial, and then derive the approximated offset curve $\boldsymbol{C}_r^a(t) = \boldsymbol{C}(t) + r\boldsymbol{Q}(s(t))$. The reparameterization may be designed to make $\boldsymbol{C}_r^a(t)\, C^1$ continuous, interpolate the endpoints of the exact offset, and offset arcs precisely.

In this case, the approximation error only comes from the inconsistency of the directions of $\boldsymbol{Q}(s(t))$ and $\boldsymbol{N}(t)$. Error analysis in Section 4 shows that the error is much smaller than that of Lee et al. [14]. So our algorithm not only improves the continuity but also reduces the approximation error.

Inspired by Lee et al. [14], the approximated reparameterization may be defined by

$$s(t) = \frac{(a(1-t) + bt)x'(t) + (c(1-t) + dt)y'(t)}{(e(1-t) + ft)x'(t) + (g(1-t) + ht)y'(t)},$$

where a, b, c, d, e, f, g, h are undetermined coefficients.

Note that $\boldsymbol{C}_r^a(t)$ should be a C^1 endpoints interpolation of the exact offset, which requires that

$$s(i) = i, \quad \left.\frac{d\boldsymbol{Q}(s(t))}{dt}\right|_{t=i} = \left.\frac{d\boldsymbol{N}(t)}{dt}\right|_{t=i}, \quad i = 0, 1. \tag{1}$$

In addition, arcs should be offset precisely. Assume that the arc is represented by

$$\boldsymbol{C}^*(t) = (x^*(t), y^*(t))$$

$$= \left(\frac{(1-t)^2 + 2(1-t)t\cos\theta + t^2\cos 2\theta}{(1-t)^2 + 2(1-t)t\cos\theta + t^2}, \frac{2(1-t)t\sin\theta + t^2\sin 2\theta}{(1-t)^2 + 2(1-t)t\cos\theta + t^2}\right).$$

In this case, $\boldsymbol{C}^*(t) + r\boldsymbol{Q}(s(t))$ should be equal to the exact offset. That is,

$$\boldsymbol{C}^*(t) + r\boldsymbol{Q}(s(t)) = \boldsymbol{C}^*(t) + r\boldsymbol{N}^*(t),$$

where $\boldsymbol{N}^*(t)$ is the unit normal of $\boldsymbol{C}^*(t)$.

It is easy to transform the above equation to

$$t = s(t). \tag{2}$$

Solving the linear system of equations (1) and (2), we obtain

$$a = \cos\theta, \quad b = 1, \quad c = d = 0, \quad e = \cos\theta - \cos 2\theta,$$

$$f = 1 - \cos\theta\cos 2\theta, \quad g = -\sin 2\theta, \quad h = -\sin 2\theta\cos\theta.$$

Then the reparameterization $s(t)$ is represented by

$$s(t) = \frac{[\cos\theta(1-t) + t]x'(t)}{[(\cos\theta - \cos 2\theta)(1-t) + (1 - \cos\theta\cos 2\theta)t]x'(t) - \sin 2\theta[(1-t) + t\cos\theta]y'(t)}. \tag{3}$$

Here we give some explanations for $s(1) = 1$. Applying $y'(1)\sin 2\theta = -x'(1)\cos 2\theta$ to $s(1)$, we have

$$s(1) = \frac{-\sin 2\theta}{(1 - \cos\theta\cos 2\theta)(-\sin 2\theta) - \sin 2\theta\cos\theta\cos 2\theta} = 1.$$

For those regular, but not simple original curves, it should be subdivided so that each segment is convex or concave, and the range of direction angles of normal vectors is less than $\pi/2$. Then the parameter domain should be transformed to $[0, 1]$ by a simple translation and scaling. Finally, a rotation is required to locate the beginning point on the X-axis, with its tangent parallel to the positive Y-axis.

For a polynomial curve $\boldsymbol{C}(t)$ of degree d, the degree of our approximated offset curve is $3d$, and for a rational curve, the degree is $5d - 2$, a little higher than $3d - 2$ and $5d - 4$ of the method suggested by Lee et al. [14]. However, Section 5 shows that the experimental results of our method are much better.

4. Error Analysis

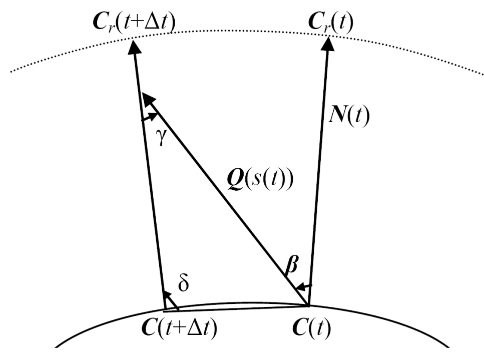


Figure 2. The curve segment is offset in the convex direction.

It has been mentioned that, for any $t \in [0, 1]$, the distance between the original curve point $\boldsymbol{C}(t)$ and the approximated offset point $\boldsymbol{C}_r^a(t)$ is guaranteed to be equal to the offset radius r, while the direction of the difference vector $\boldsymbol{C}_r^a(t) - \boldsymbol{C}(t)$ is inconsistent with that of the normal vector $\boldsymbol{N}(t)$. Assume that the deflection angle between $\boldsymbol{C}_r^a(t) - \boldsymbol{C}(t)$ and $\boldsymbol{N}(t)$ is β, and $\cos\beta = \boldsymbol{Q}(s(t)) \cdot \boldsymbol{N}(t)$. In the neighborhood of the point $\boldsymbol{C}(t)$, we can find a unique point $\boldsymbol{C}(t + \Delta t)$ on the original curve, at which the normal line just passes through $\boldsymbol{C}_r^a(t)$. Here Δt is a slight increment and $t + \Delta t \in [0, 1]$. That is, for any $t \in [0, 1]$, a constant λ can be found so that $\boldsymbol{C}(t + \Delta t) + \lambda r\boldsymbol{N}(t + \Delta t) = \boldsymbol{C}(t) + r\boldsymbol{Q}(s(t))$, and therefore $r(\lambda - 1)$ is considered as the actual approximation error [19]. Denote the unit normal vector at $\boldsymbol{C}(t + \Delta t)$ by $\boldsymbol{N}(t + \Delta t)$. Assume that the direction angle between $\boldsymbol{N}(t + \Delta t)$ and $\boldsymbol{Q}(s(t))$ is γ, and that between $\boldsymbol{N}(t + \Delta t)$ and $\boldsymbol{C}(t) - \boldsymbol{C}(t + \Delta t)$ is δ, then we have

$$\lambda = \boldsymbol{Q}(s(t)) \cdot \boldsymbol{N}(t + \Delta t) + \frac{\boldsymbol{C}(t) - \boldsymbol{C}(t + \Delta t)}{r} \cdot \boldsymbol{N}(t + \Delta t)$$

$$= \cos\gamma + \frac{\| \boldsymbol{C}(t) - \boldsymbol{C}(t + \Delta t) \|}{r} \cdot \cos\delta.$$

To estimate $r(\lambda - 1)$, the error bounds proposed in [19] are used. The original curve segments are classified into two kinds. For those offset in their convex side, the error bound is

$$-r\sin^2\beta \le r(\lambda - 1) \le 0, \tag{4}$$

and for those offset in their concave side, the error bound is

$$-\frac{r\sin^2\beta}{(\cos\beta - \eta)^2 + \sin^2\beta} \le r(\lambda - 1) \le 2r\left(\sin^2\frac{\beta}{2}\right)\frac{\cos\beta(1 + 2\eta) - \eta^2}{(\cos\beta - \eta)^2}, \tag{5}$$

where $\eta = r/\rho(t)$, and $\rho(t)$ is the curvature radius function of the original curve.

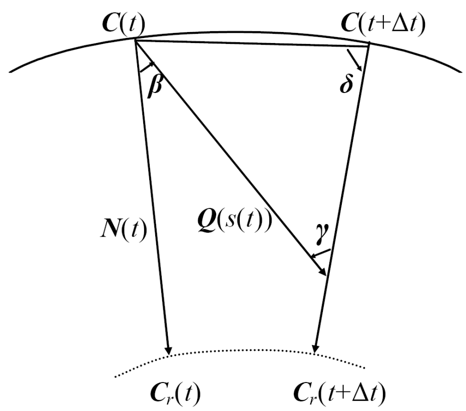


Figure 3. The curve segment is offset in the concave direction.

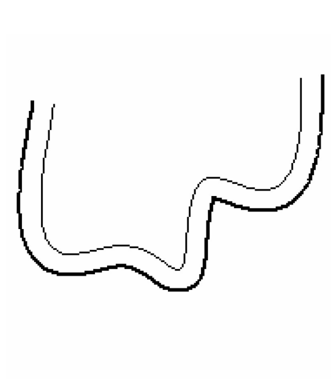
For those points where ρ is close to the offset radius r, the bound in equation (5) may be too large to be estimated. However, since self-intersections would happen in the neighborhood of those points, they are usually intentionally eliminated in advance, in practice. Thus we can give no consideration of them. Hence a positive constant $\eta^* > \eta = r/\rho$ can be found for any point $\boldsymbol{C}(t)$, and then in equation (5), replacing η with η^* we obtain an error bound only related to β.

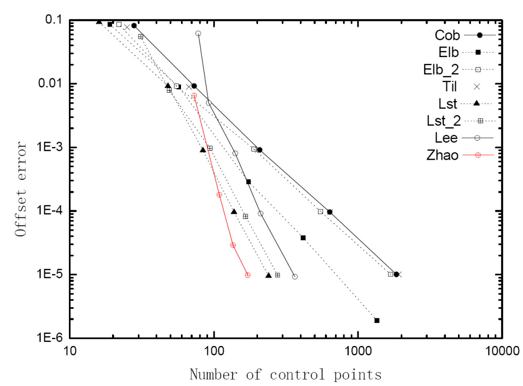
Note that $\cos\beta = \boldsymbol{Q}(s(t)) \cdot \boldsymbol{N}(t)$. Hence the range of $\cos\beta$ or β can be derived by symbolic computation. Applying it to equations (4) and (5), we get the error bounds for the two cases, respectively. Clearly, both the bounds are dependent on θ. Numerical analysis and symbol computation indicate that our error bound is at least $O((\theta)^6)$, better than $O((\theta)^4)$ in Lee et al. [14].

5. Experimental Results

In this section, we compare experimental results of our method with previous methods. The examples are the same as those in Lee et al. [14]. The total number of the control points of the approximated offset curve is compared between different algorithms.

The first example is the circle represented by piecewise quadratic rational Bézier curve segments. It is clear that our algorithm can precisely offset it. The resultant offset has the same degree and control points with the exact offset curve.





$\lvert\delta\rvert$	Cob	Elb	Elb_2	Til	Lst	Lst_2	Lee	Zhao
10^{-1}	28	19	22	25	16	31	78	73
10^{-2}	73	57	55	67	48	49	92	73
10^{-3}	208	174	190	202	84	94	141	109
10^{-4}	637	417	550	640	138	166	211	136
10^{-5}	1846	1357	1690	1918	240	277	365	172

Figure 4. Uniform cubic B-spline curve and its offset.

For the other example, the comparisons are shown in the table in Figure 4. The input curve in Figure 4 is a uniform cubic B-spline curve with 7 control points: (−3.01619, 2.34143), (−3.97193, −2.20842), (−1.07045, 0.0722807), (0.319568,

−2.77522), (−0.152767, 2.299), (2.92416, −0.939865), (2.8027, 3.02775). An offset radius 0.5 is used for the example. The input curve in our algorithm generates less control points under the same tolerance than the other methods, especially for high precision offset approximation. In addition, the approximated offset is C^1 continuous. Many other algorithms achieve only at most G^1 continuity.

Although it is difficult to represent an accurate error bound for the global control due to the complicated expression of the normal $\boldsymbol{N}(t)$, a practical one can be found by a statistical technique, based on strict numerical analysis of many examples. For the length limitation of the paper, below we only give some explanations to the above two examples.

The uniform cubic B-spline curve (see Figure 4) can be divided into four curve segments. Compute 2θ and $\sin^2\beta$ for each curve segment. By fitting technique, we have $\sin^2\beta \approx 0.019(\sin\theta)^{6.105}$, $\sin^2\beta \approx 0.03(\sin\theta)^{6.181}$, $\sin^2\beta \approx 0.024(\sin\theta)^{6.182}$ and $\sin^2\beta \approx 0.0214(\sin\theta)^{6.175}$. So for the cubic B-spline curve, $\sin^2\beta$ is at least $O((\theta)^6)$.

It is well known that data explosion is a serious problem in offset computation. Therefore, it is necessary to store only a minimum amount of computation results. Under such consideration, our algorithm is of much significance because of its excellent approximation effect.

6. Conclusion

We presented an offset curve approximation method based on exact presentation for the swept circle. Theoretical analysis and experimental results show that our algorithm is superior to most of the other methods in terms of the approximation effect, especially for high precision approximation. Our algorithm can offset circles without using sample techniques, and applies a more effective error bound. Furthermore, C^1 offset approximation is yielded.

Like Lee et al.'s method [14], the main disadvantage of our method is that the approximated offset curves are rational curves of relatively high degree. The degrees are $3d$ and $5d - 2$, respectively, a little higher than $3d - 2$ and $5d - 4$ in

Lee et al. [14], where d is the degree of the original Bézier or rational Bézier curve. However, it is deserved at the expense of little increase of the degree to achieve much better approximation with a smaller number of segmented curves and control points, especially for high precision approximation.

Acknowledgements

This work is supported by the National Grand Fundamental Research 973 Program of China (No. 2004CB719400), the National Natural Science Foundation of China (Nos. 60373033, 60333010) and the National Natural Science Foundation for Innovative Research Groups (No. 60021201).

References

[1] A. Hansen and F. Arbab, An algorithm for generating NC tool paths for arbitrarily shaped pockets with islands, ACM Trans. Graphics 11(2) (1992), 152-182.

[2] M. Held, On the Computational Geometry of Pocket Machining, Springer-Verlag, Berlin, Germany, 1991.

[3] Y. J. Chen and B. Ravani, Offset surface generation and contouring in computer-aided design, J. Mechanisms, Transmissions and Automation in Design: ASME Transactions 109(3) (1987), 133-142.

[4] T. Kuragano, N. Sasaki and A. Kikuchi, The FRESDAM system for designing and manufacturing free form objects, USA-Japan Cross Bridge, Flexible Automation, R. Martin, ed., 2 (1988), 931-938.

[5] N. M. Patrikalakis and P. V. Prakash, Free-form plate modeling using offset surfaces, J. Offshore Mechanics and Arctic Engineering 110(3) (1988), 287-294.

[6] R. T. Farouki and T. Sakkalis, Pythagorean hodographs, IBM J. Research and Development 34(5) (1990), 736-752.

[7] E. S. Cobb, Design of sculptured surfaces using the B-spline representation, Ph.D. Dissertation, University of Utah, USA, 1984.

[8] W. Tiller and E. G. Hanson, Offsets of two-dimensional profiles, IEEE Computer Graphics and Application 4(9) (1984), 36-46.

[9] G. Elber and E. Cohen, Offset approximation improvement by control points perturbation, Mathematical Methods in Computer Aided Geometric Design II, T. Lyche and L. L. Schumaker, eds., New York, 1992, pp. 229-237.

[10] J. Hoscheck, Spline approximation of offset curves, Comput. Aided Geom. Design 20(1) (1988), 33-40.

[11] L. A. Piegl and W. Tiller, Computing offsets of NURBS curves and surfaces, Computer-Aided Design 31(2) (1999), 147-156.

[12] Y. M. Li and V. Y. Hsu, Curve offsetting based on Legendre series, Comput. Aided Geom. Design 15(7) (1998), 711-720.

[13] G. Elber, I. K. Lee and M. S. Kim, Comparing offset curve approximation methods, IEEE Computer Graphics and Applications 17(1) (1997), 62-71.

[14] I. K. Lee, M. S. Kim and G. Elber, Planar curve offset based on circle approximation, Computer Aided Design 28(8) (1996), 617-630.

[15] I. K. Lee, M. S. Kim and G. Elber, New approximation methods of planar offset and convolution curves, Geometric Modeling: Theory and Practice, W. Strasser, R. Klein and R. Rau, eds., Springer-Verlag, Heidelberg, 1997, pp. 83-101.

[16] I. K. Lee, M. S. Kim and G. Elber, Polynomial/rational approximation of Minkowski sum boundary curves, Graphical Models and Image Processing 60(2) (1998), 136-165.

[17] Y. J. Ahn, Y. S. Kim and Y. Shin, Approximation of circular arcs and offset curves by Bézier curves of high degree, J. Comput. Appl. Math. 167(2) (2004), 405-416.

[18] M. Floater, An $O(h^{2n})$ Hermite approximation for conic sections, Computer Aided Geometric Design 14(2) (1997), 135-151.

[19] H. Y. Zhao and G. J. Wang, Error analysis for curve offsetting methods based on precise circle representation, Computer-Aided Design 39(2) (2007), 142-148.