Qualitative and Quantitative Comparisons of Offset Curve Approximation Methods

Gershon Elber[+,*], In-Kwon Lee[*], and Myung-Soo Kim[*]
[+]Department of Computer Science, Technion, IIT, Haifa 32000, Israel
[*]Department of Computer Science, POSTECH, Pohang 790-784, Korea

April 18, 2006

Offset curves have diverse engineering applications, which have consequently motivated extensive research concerning various offset techniques. Offset research in the early 1980s focused on approximation techniques to solve immediate application problems in practice. This trend continued until 1988, when Hoschek [1, 2] applied non-linear optimization techniques to the offset approximation problem. Since then, it has become quite difficult to improve the state-of-the-art of offset approximation.

Offset research in the 1990s has been more theoretical. The foundational work of Farouki and Neff [3] clarified the fundamental difficulty of exact offset computation. Farouki and Sakkalis [4] suggested the Pythagorean Hodograph curves which allow simple rational representation of their exact offset curves. Although many useful plane curves such as conics do not belong to this class, the Pythagorean Hodograph curves may have much potential in practice, especially when they are used for offset approximation.

In a recent paper [5] on offset curve approximation, the authors suggested a new approach based on approximating the offset circle, instead of approximating the offset curve itself. To demonstrate the effectiveness of this approach, we have made extensive comparisons with previous methods. To our surprise, the simple method of Tiller and Hanson [6] outperforms all the other methods for offsetting (piecewise) quadratic curves, even though its performance is not as good for high degree curves.

The experimental results have revealed other interesting facts, too. If these details had been reported several years ago, we believe, offset approximation research might have developed somewhat differently. This paper is intended to fill in an important gap in the literature. Qualitative as well as quantitative comparisons are conducted employing a whole variety of contemporary offset approximation methods for freeform curves in the plane. The efficiency of the offset approximation is measured in terms of the number of control points generated while the approximations are made within a prescribed tolerance.

*This work was supported in part by the Fund for Promotion of Research at the Technion, Haifa, Israel.

Offset of Planar Curves

Given a regular parametric curve, $C(t) = (x(t), y(t))$, in the plane, its offset curve $C_d(t)$ by a constant radius d, is defined by:

$$C_d(t) = C(t) + d \cdot N(t), \tag{1}$$

where $N(t)$ is the unit normal vector of $C(t)$:

$$N(t) = \frac{(y'(t), -x'(t))}{\sqrt{x'(t)^2 + y'(t)^2}}. \tag{2}$$

The regularity condition of $C(t)$ guarantees that $(x'(t), y'(t)) \neq (0, 0)$ and $N(t)$ is well defined on the curve $C(t)$. Equation (2) has a square root term in the denominator. Therefore, even if the given curve $C(t)$ is a polynomial curve, its offset is not a polynomial or rational curve, in general. This fundamental deficiency has motivated the development of various polynomial and rational approximation techniques of $C_d(t)$. While the offset to a polynomial or rational parametric curve must be approximated, it is somewhat counter-intuitive that a close cousin of the offset, the evolute, is indeed always representable as a rational curve (see Sidebar on Evolute).

Evolute

The evolute of $C(t)$ is defined as:

$$E(t) = C(t) + \frac{N(t)}{\kappa(t)},$$

where $\kappa(t)$ is the curvature of $C(t) = (x(t), y(t))$:

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}.$$

That is, $E(t)$ is a variable radius offset with offset radius $d(t) = \frac{1}{\kappa(t)}$. Figure 1 shows two examples of evolute curves. Quite surprisingly, $E(t)$ is a rational curve, provided $C(t)$ is a rational or polynomial curve:

$$\begin{aligned} E(t) &= C(t) + \frac{N(t)}{\kappa(t)} \\ &= C(t) + \frac{\frac{(y'(t), -x'(t))}{(x'(t)^2 + y'(t)^2)^{1/2}}}{\frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}} \\ &= C(t) + \frac{x'(t)^2 + y'(t)^2}{x'(t)y''(t) - x''(t)y'(t)} \cdot (y'(t), -x'(t)). \end{aligned}$$

In contrast to the offset computation in Equation (1), there is no square root term in the representation of $E(t)$. In Figure 1, the curves and their evolutes are both represented as B-spline curves.

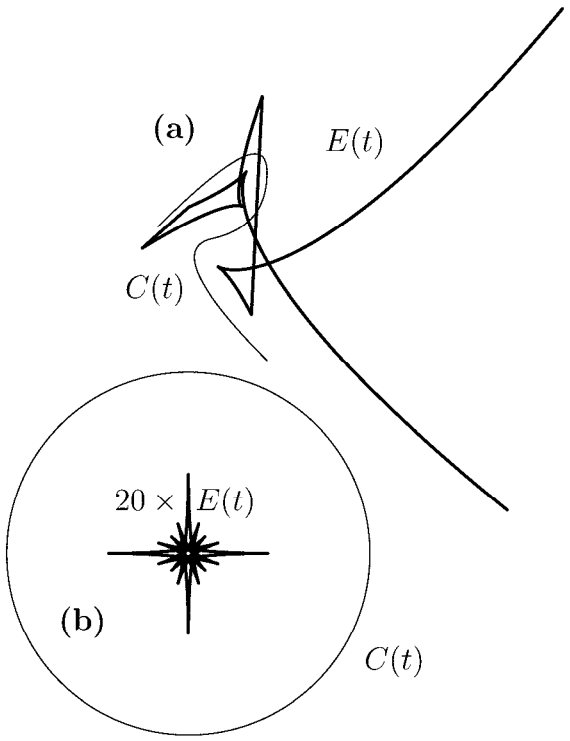


Figure 1: In (a), a B-spline curve, $C(t)$, in light curve, is shown along with its evolute, $E(t)$, in bold curve. In (b), the evolute $E(t)$ for a cubic polynomial approximation of a circle, $C(t)$, is shown. $E(t)$ is scaled up by a factor of 20.

Most offset approximation techniques are based on an iterative process of fitting an approximation curve, measuring the accuracy, and subdividing the problem into smaller problems if the approximation error is larger than the tolerance. This divide and conquer approach exploits the subdivision property of the base curve $C(t)$. Henceforth, we assume $C(t)$ is represented by a Bézier or NURBS curve.

Denote by $C_d^a(t)$ the *approximation* of $C_d(t)$. Traditionally, the offset approximation error has been measured only at finite sample points along $C(t)$, computing $\epsilon_i = \|C(t_i) - C_d^a(t_i)\| - d$. Elber and Cohen [7] proposed a symbolic method which computes the global error between squared distances:

$$\epsilon(t) = \|C(t) - C_d^a(t)\|^2 - d^2. \tag{3}$$

The error function $\epsilon(t)$ is obtained by symbolically computing the difference and inner product of Bézier or NURBS curves (see Sidebar on Symbolic Computation and Reference [8]). Therefore, it can be represented as a Bézier or NURBS scalar function. As a scalar field, the largest coefficient of $\epsilon(t)$ *globally* bounds the maximal possible error due to the convex hull property of Bézier or NURBS formulation. In this article, we exploit the error functional $\epsilon(t)$ of Equation (3) to measure all the offset approximation errors. This provides not only a global bound for each method, but also an equal basis for the comparison of different methods.

Qualitative Comparisons

Control Polygon Based Methods

Let $C(t)$ be a B-spline curve with k control points of order n and defined over a knot sequence $\tau = \{t_i\}$, $0 \le i < n + k$. The i-th *node* parameter value ξ_i of $C(t)$ is defined as:

$$\xi_i = \frac{\sum_{j=i+1}^{i+n-1} t_j}{n-1}, \tag{7}$$

for $0 \le i < k$. Hence, a node parameter value is an average of $n-1$ consecutive knots in τ. Each control point, P_i, of $C(t)$, is associated with one node, ξ_i.

Symbolic Computation

In this article, we have employed $\epsilon(t)$ (in Equation (3)) and $\epsilon_m(t)$ (in Equation (11)) to estimate the offset approximation error. Furthermore, we also need to compute the composition of $U(s(t))$ (in Equation (10)). The symbolic computation of these equations involves the difference, product, and sum of (piecewise) scalar polynomial or rational curves.

Let $C_1(t) = \sum_{i=0}^{m} P_i B_{i,\tau}(t)$ and $C_2(t) = \sum_{j=0}^{n} P_j B_{j,\eta}(t)$ be two (piecewise) polynomial regular parametric curves, in the Bézier or NURBS representations. The computation of $C_1(t) \pm C_2(t)$ can be accomplished by elevating both $C_1(t)$ and $C_2(t)$ to a common function space. The order of the common function space is equal to the maximal order of $C_1(t)$ and $C_2(t)$. If either $C_1(t)$ or $C_2(t)$ is a B-spline curve, the common function space is defined by considering both knot vectors τ and η and preserving the *lowest* degree of continuity at each knot. Once the common function space is determined, both $C_1(t)$ and $C_2(t)$ are elevated to this space via degree raising and refinement. (See [8, 9] for more details as well as the extension to rationals.)

The computation of $C_1(t)C_2(t)$ is somewhat more involved. Here, we consider only the case of Bézier polynomial curves. (See [8] for the more general cases of piecewise polynomials and rationals.) The i-th Bernstein Bézier basis function of degree k is defined by:

$$B_i^k(t) = \binom{k}{i} t^i (1-t)^{k-i}. \tag{4}$$

The product of two Bernstein Bézier basis functions is:

$$\begin{aligned} B_i^k(t) B_j^l(t) &= \binom{k}{i} t^i (1-t)^{k-i} \binom{l}{j} t^j (1-t)^{l-j} \\ &= \binom{k}{i}\binom{l}{j} t^{i+j} (1-t)^{k+l-i-j} \\ &= \frac{\binom{k}{i}\binom{l}{j}}{\binom{k+l}{i+j}} B_{i+j}^{k+l}(t). \end{aligned} \tag{5}$$

Therefore, we have:

$$\begin{aligned} C_1(t)C_2(t) &= \sum_{i=0}^{m} P_i B_i^m(t) \sum_{j=0}^{n} P_j B_j^n(t) \\ &= \sum_{i=0}^{m}\sum_{j=0}^{n} P_i P_j B_i^m(t) B_j^n(t) \\ &= \sum_{i=0}^{m}\sum_{j=0}^{n} P_i P_j \frac{\binom{m}{i}\binom{n}{j}}{\binom{m+n}{i+j}} B_{i+j}^{m+n}(t) \\ &= \sum_{k=0}^{m+n} Q_k B_k^{m+n}(t), \end{aligned} \tag{6}$$

where Q_k accumulates all the combinatorial terms $P_i P_j \frac{\binom{m}{i}\binom{n}{j}}{\binom{m+n}{i+j}}$, for $k = i+j$. Hence, $C_1(t)C_2(t)$ is represented as a Bézier polynomial curve of degree $m+n$.

$C(\xi_i)$ is typically close to P_i; however, it is not the closest point of $C(t)$ to P_i, in general.

Cobb [10] translated each control point, P_i, by $d \cdot N(\xi_i)$, whereas Tiller and Hanson [6] translated each edge of the control polygon into the edge normal direction by a distance d. Unfortunately, Cobb [10] always *under-estimates* the offset; i.e., $\epsilon(t) \leq 0$, for all t. (For the proof and related issues, see Sidebar on Under and Over-Estimation.)

Tiller and Hanson [6] do not under-estimate the offset curve. In addition to computing the exact linear and circular offset curves, their method outperforms all the other methods for the case of offsetting (piecewise) quadratic curves. However, for offsetting high degree curves, this simple method has a similar performance to that of Cobb [10].

Coquillart [11] solved the under-estimating problem. The distance between P_i and $C(\xi_i)$, and the curvature $\kappa(\xi_i)$ of $C(t)$ at ξ_i are taken into account. Numerical approximation is also taken to compute the closest point of $C(t)$ to the control point P_i, while using $C(\xi_i)$ as an initial solution. With all these enhancements, Coquillart [11] was able to offset the linear and circular segments exactly.

Elber and Cohen [12] took a different approach that exactly computes the offsets of linear and circular elements. Using the values of $\epsilon(t)$ (in Equation (3)) at $t = \xi_0, \ldots, \xi_n$, the error in the neighborhood of each control point, P_i, is estimated and used to adjust the translational distance applied to P_i. This perturbation based approach is an iterative method that converges to the exact circular offset segment. For general curves, when the result of Cobb [10] is used as an initial solution, the perturbation process typically reduces the offset approximation error of [10] by an order of magnitude. In principle, this method can be applied to any offset approximation method that produces piecewise polynomial curves.

Most traditional techniques subdivide $C(t)$ at the middle of the parametric domain; however, a better candidate is the parameter of the location with the maximum error. Since $\epsilon(t)$ represents the exact squared error function, one can find the parameter location of the maximal error and subdivide $C(t)$ there. Alternatively, instead of subdividing $C(t)$,

Under and Over-Estimation

The offset approximation of Cobb [10] is formally defined as follows:

$$\begin{aligned} C_d^a(t) &= \sum_{i=0}^{n} (P_i + dN(\xi_i))\, B_i(t) \\ &= \sum_{i=0}^{n} P_i B_i(t) + d\sum_{i=0}^{n} N(\xi_i) B_i(t) \\ &= C(t) + dV(t). \end{aligned}$$

where $\|N(\xi_i)\| = 1$, for $i = 0, \ldots, n$. The vector field curve $V(t) = \sum_{i=0}^{n} N(\xi_i) B_i(t)$ has all its control points $N(\xi_i)$ on the unit circle S^1. By the convex hull property, we have $\|V(t)\| \leq 1$ and

$$\|C(t) - C_d^a(t)\| = \|dV(t)\| = d\|V(t)\| \leq d.$$

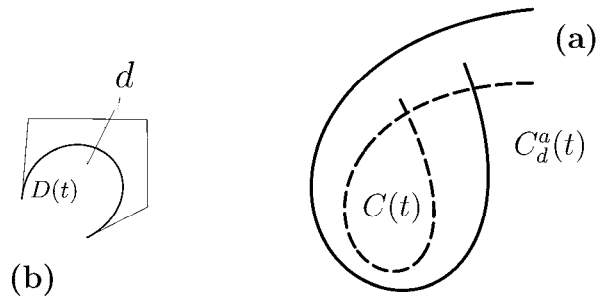


Figure 2: In (a), an offset approximation $C_d^a(t)$ computed by translating the control points of the original curve $C(t)$ (dashed lines) by an amount equal to the offset distance will always under-estimate the real offset. In (b), $D(t) = C(t) - C_d^a(t)$ is found to be fully contained in a circle of the offset radius size, d.

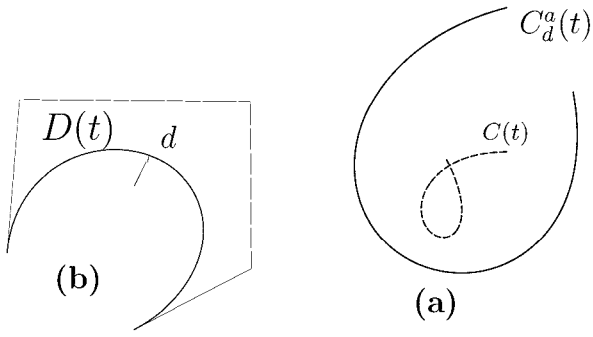


Figure 3: In (a), an offset approximation $C_d^a(t)$ of a quartic Bézier curve $C(t)$ (dashed lines) is computed by forcing $C_d^a(t)$ to over-estimate the error. $D(t) = C(t) - C_d^a(t)$ is shown in (b). Compare with Figure 2.

If $N(\xi_i) \neq N(\xi_j)$, for some $0 \leq i, j \leq n$, we have $\min \|V(t)\| < 1$, and this results in an error in the offset approximation. Hence, this method always *under-estimates* the exact offset. Figure 2(a) shows a quartic Bézier curve $C(t)$ and its offset approximation $C_d^a(t)$. In Figure 2(b), the difference vector field $D(t) = C(t) - C_d^a(t)$ is completely contained in a disk of radius d. All the control points of $D(t)$ are on the circumference of the disk.
Under-estimation of offsets may lead to undesirable results. For example, in NC machining, the under-estimation leads to gouging. Assume the under-estimation of the offset is bounded from below by:

$$d_{\min} = \min(\|dV(t)\|).$$

When we translate control point P_i in the direction of $N(\xi_i)$, by a distance $d\frac{d}{d_{\min}}$, the resulting curve completely *over-estimates* the exact offset (see Figure 3):

$$\|C(t) - C_d^a(t)\| = \left\| d\frac{d}{d_{min}} V(t) \right\| \geq \left\| d\frac{d}{\|dV(t)\|} V(t) \right\| = d.$$

One can reduce the relative error in the offset approximation by alternating the under and over-estimations. This can be done by adjusting the offset distance at each control point appropriately. Figure 4 shows an example of this approach. We use the same quartic Bézier curve as in Figures 2 and 3. The quartic Bézier offset approximation curve interpolates the exact offset at five discrete locations, corresponding to the node values, $\frac{i}{4}$, $0 \leq i \leq 4$.

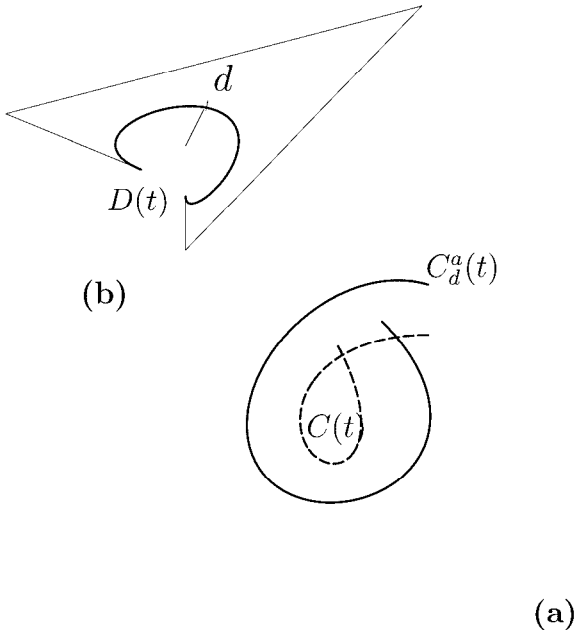


Figure 4: In (a), an offset approximation $C_d^a(t)$ of a quartic Bézier curve $C(t)$ (dashed lines) is computed by enforcing $C_d^a(t)$ to interpolate at five locations on the exact offset curve computed at the node values on $C_d^a(t)$. $D(t) = C(t) - C_d^a(t)$ is shown in (b). Compare with Figures 2 and 3.

one can insert new knots into $C(t)$ at the parameter locations with error larger than the allowed tolerance. Elber and Cohen [7] took this refinement approach.

Interpolation Methods

Klass [13] used a cubic Hermite curve to approximate the offset curve. The cubic Hermite curve is determined by interpolating the position and velocity of the exact offset curve at both endpoints. The numerical approximation procedure of Klass [13] is quite unstable when the offset curve becomes almost flat. Therefore, instead of using the original algorithm [13], we compute the first derivative of the offset curve based on the following simple closed form equation (see also [3]):

$$C_d'(t) = (1 + d \cdot \kappa(t))C'(t), \tag{8}$$

where $\kappa(t)$ is the curvature of $C(t)$.

Hoschek [1] suggested a least squares solution for the determination of $C_d'(t)$ at the curve endpoints. That is, at each endpoint of $C_d(t)$, the direction of $C_d'(t)$ is maintained to be parallel to $C'(t)$; however, instead of using Equation (8), their lengths are determined so that the cubic Hermite curve best fits $C_d(t)$ in the least squares sense. For computational efficiency, only finite samples of $C_d(t)$ are used in the optimization.

Hoschek and Wissel [2] used a general non-linear optimization technique to approximate a high degree spline curve with low degree spline curves. They applied the same technique to approximate an exact offset curve with low degree spline curves.

The least squares based methods [1, 2] are expected to perform better than other methods. However, there still remains a question about whether the least squares solution is optimal when searching for the smallest number of (say cubic) curve segments to approximate an exact offset curve. The answer is negative, in general. In the special case of offsetting quadratic curves, the simple method of Tiller and Hanson [6] performs much better than the least squares methods [1, 2].

It is important to question how this unexpected result could be obtained. The answer might be quite useful in improving the accuracy of offset approximation. The least squares solution optimizes the integrated summation of the least squares errors in the approximation. Therefore, even if a small portion of the approximation curve has a large error, as long as the rest of the curve tightly approximates the exact curve, the overall least squares error can be very small. That is, the least squares solution provides an optimal solution with respect to an L_2 norm. When this L_2 optimal solution is further evaluated with respect to the L_∞ norm (of Equation (3)), the optimality is no longer guaranteed. This is an important observation which suggests possible improvements over the nearly optimal solutions [1, 2].

Pham [14] suggested a simple B-spline interpolation method to approximate the offset curve. Finite sample points are generated on the exact offset curve, and they are interpolated by a piecewise cubic B-spline curve. It is also interesting to note that this simple method also performs pretty well. In many examples, its performance is only slightly worse than and sometimes even better than the local least squares methods [1, 2].

Circle Approximation Methods

Assume the base curve $C(t)$, $t_0 \le t \le t_1$, is a polynomial curve with no inflection point, and a unit circular arc $U(s)$, $s_0 \le s \le s_1$, is parameterized so that:

$$C'(t_0) \parallel U'(s_0) \quad \text{and} \quad C'(t_1) \parallel U'(s_1).$$

If one can compute a reparameterization $s(t)$ so that:

$$C'(t) \parallel U'(s(t)),$$

the offset curve is then computable as:

$$C_d(t) = C(t) + d \cdot U(s(t)). \tag{9}$$

The offset curve is not a polynomial or rational curve; therefore, we have to approximate $U(s)$ and/or $s(t)$ by a polynomial or rational.

Lee et al. [5] approximated the unit circle $U(s)$ with piecewise quadratic polynomial curve segments $Q_j(s)$, $j = 0, \ldots, n$. The Hodograph curve $Q_j'(s)$ is piecewise linear; therefore, the parallel constraint:

$$C'(t) \parallel Q'(s(t))$$

provides the reparameterization of $s(t)$ as a rational polynomial of degree $d-1$, where d is the degree of $C(t)$. For a polynomial curve $C(t)$ of degree d, the resulting offset approximation (computed as in Equation (9)) is a rational curve of degree $3d-2$. (For a rational curve $C(t)$ of degree d, the offset approximation curve is of degree $5d-4$.)

For a quadratic polynomial curve $C(t)$, this technique also provides a simple method to represent the *exact* offset curve $C_d(t)$ as a rational curve of degree six. Assume that the exact circle, $Q(s)$, $0 \leq s \leq 1$, is represented by a rational quadratic curve. Then, the parallel constraint:

$$C'(t(s)) \parallel Q'(s)$$

provides the reparameterization of $t(s)$ as a rational polynomial of degree two. Therefore, the exact offset curve $C_d(t)$ is a rational curve of degree six. Even with the high degree of six, the exact offset capability suggests this method as the method of choice for offsetting (piecewise) quadratic polynomial curves, especially for high precision offset approximation. However, this exact offset capability does not extend to rational quadratic curves. (There are some rational quadratic curves which have no exact rational parametrization of their offset curves.)

One can attempt to globally approximate $s(t)$ by maximizing the constraint energy:

$$\max_{s(t)} \int_{t_0}^{t_1} \frac{\langle C'(t), U'(s(t))\rangle}{\|C'(t)\| \, \|U'(s(t))\|} dt. \qquad (10)$$

This approach was taken in Lee et al. [15], in which the composition of $U(s(t)) = (U \circ s)(t)$ is carried out symbolically [8] (see also Sidebar on Symbolic Computation).

The offset approximation in [5] depends on the method used for the piecewise quadratic approximation to the circle. The error in the offset approximation stems only from the quadratic polynomial approximation of the circular arc, scaled by the offset radius d. Lee et al. [5] used five different circle approximation methods. Two of the five methods generate G^1-continuous circle approximations with quadratic Bézier curve segments. In the first method, the unit circle $U(s)$ is totally contained in the closed convex region bounded by the quadratic curve segments. The corresponding offset curve approximation completely over-estimates the exact offset curve. In the second method, the quadratic curve segments pass through both the interior and exterior of the unit circle $U(s)$. Therefore, the offset approximation curve both over and under-estimates the exact offset curve, while the approximation error is reduced by half from the over-estimating first method. We use this second method, referred to as *Lee* in the next section, for comparison with other methods.

In contrast, Lee et al. [15] approximated the reparametrization $s(t)$, while representing the circle $U(s)$ exactly by a rational quadratic curve. In this method, the error stems only from the inaccurate reparameterization function $s(t)$, which results in a mismatch in the parallel constraint of $C'(t) \parallel U'(s(t))$. To the authors' knowledge, this is the only offset approximation method for which the use of $\epsilon(t)$ is completely ineffective in the global error bound. The term $\epsilon(t)$ is always equal to zero. Lee et al. [15] measured the angular deviation of $U(s(t))$ from the exact offset direction $N(t)$ by using the following error function:

$$\epsilon_m(t) = \frac{\langle C(t) - C_d^a(t), C'(t)\rangle^2}{d^2 \, \|C'(t)\|^2}. \qquad (11)$$

The error is equal to zero if orthogonality is preserved. Otherwise, it is equal to $\cos^2\theta$, where θ is the angle between $U(s(t))$ and $C'(t)$.

Quantitative Comparisons

We consider how efficiently each method approximates the offset curve, given a prescribed tolerance. Several examples of Bézier and B-spline curves are given, both in polynomial and rational forms. All the methods (compared in this article) are implemented using the IRIT [16] solid modeling system that has been developed at the Technion, Israel, with some of the offset approximation methods implemented at POSTECH, Korea.

Methods under Comparison

We quantitatively compare the following methods:

- *Cob*: The simple method of Cobb [10] in which the control points are translated by the offset distance. This method always creates under-estimated offsets. (See Sidebar on Under and Over-Estimation.)

- *Elb*: An adaptive offset refinement approach that was suggested in Elber and Cohen [7]. Instead of subdividing the base curve, whenever the error is too large, the offset curve is refined to yield a better approximation (by using more control points). The error analysis of $\epsilon(t)$ is exploited to find better candidate locations for refinement. This method also under-estimates the offset curves.

- *Coq*: The enhancement suggested by Coquillart [11] that allows the exact offset representation of linear as well as circular segments.

- *Til*: The method of Tiller and Hanson [6] in which the edges of the control polygon, rather than the control points, are translated.

- *Klass*: The method of Klass [13] that fits a cubic Bézier curve to each offset curve segment to interpolate the boundary points and velocities of the exact offset curve.

- *Pham*: The method of Pham [14] that interpolates a sequence of finite sample points on the exact offset curve by a non-uniform piecewise cubic B-spline curve. (The original method of Pham [14] uses a uniform B-spline curve; however, we have modified the method.) Whenever the offset approximation error is larger than the prescribed tolerance, more sample offset points are used for a better fit.

- *Lst*: The global least squares approximation that fits a uniform piecewise cubic B-spline curve to the offset curve. Whenever the offset approximation error is larger than the prescribed tolerance, more control points are used for a better fit.

- *Hos*: The least squares method of Hoschek [1, 2] that fits a cubic Bézier curve to each offset curve segment. Whenever the error is larger than the tolerance, the base curve is subdivided into two subsegments and the offset approximation is repeated recursively.

- *Lee*: The approach suggested by Lee et al. [5] that approximates the curve of the convolution between $C(t)$ and the offset circle $d \cdot U(s)$ of radius d.

Traditionally, the offset approximation error has been measured only at finite sample points of $C(t)$ and $C_d^a(t)$. As previously mentioned, we adopt the symbolic approach of error estimation [7]. Therefore, we can provide an L_∞ global upper bound on the offset approximation error for each of the methods under comparison. The global error bound is derived by symbolically computing the error function $\epsilon(t)$ (in Equation (3)). Because of the convex hull property of the Bézier or NURBS representation of the scalar function $\epsilon(t)$, we can easily determine its upper bound as the maximum coefficient of the Bézier or NURBS basis functions.

Comparison Results and Remarks

Figures 5–6 show the results of offsetting (piecewise) quadratic curves. We compare the number of control points with respect to the accuracy of offset approximation. In these examples, the method of Tiller and Hanson [6] outperforms all the other methods even if the base curve has sharp corners with high curvature (Figure 6). This surprising result has never been reported in the literature. In fact, we have assumed that the least squares methods provide near optimal solutions to the offset approximation problem. However, the superior performance of Tiller and Hanson [6] tells us that this is not true, in general. At this moment, we have no clear explanation of the underlying geometric properties of this unusual phenomenon. Nevertheless, it is not difficult to point out at least two possible sources of the non-optimality in the current least squares methods:

- As discussed above, the least squares methods provide the optimal solutions in an L_2 norm, which may be quite different from the optimal solutions in an L_∞ norm.

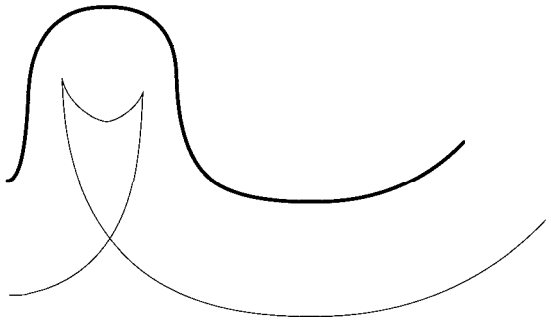
- The least squares optimization procedure solves an over-constrained problem, the solution of which depends on the distribution of finite sample points on the offset curve. In some degenerate cases, the least squares solution may have large variation depending on the distribution of data points.

Further investigations are required to eliminate these limitations, and this may advance the state-of-the-art of offset curve approximation.

Figures 7, 9, and 10 show other examples of offsetting (piecewise) cubic B-spline curves. Throughout the conducted tests, we have observed the following consistent results:

- The under-estimating offset approximation method, *Cob*, is doing quite poorly.
- The adaptive offset refinement approach, *Elb*, is better than *Cob*, especially when high precision is desired.
- In the case of offsetting (piecewise) quadratic curve segments, the simple method of Tiller and Hanson [6] outperforms all the other methods, especially when high precision is required.
- In the case of offsetting (piecewise) cubic curve segments, the least squares methods, *Lst* and *Hos*, perform much better than all the other methods, especially when high precision is required.
- In many examples, the local cubic B-spline interpolation method, *Pham*, has similar – and sometimes even better – performance to *Hos*. However, its performance deteriorates when the base curve has a radius of curvature similar to the offset radius.
- The only geometrical method that approaches the efficiency of the least squares methods is *Lee* followed not so closely by *Elb*.

For the case of offsetting (piecewise) cubic curves, the global least squares method, *Lst*, outperforms all the other methods, while it is closely followed by the local least squares method, *Hos*, and also by



ϵ	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
10^{-0}	8	8	8	8	30	8	8	8	33
10^{-1}	15	8	19	11	93	10	11	15	33
10^{-2}	41	29	37	21	132	22	39	35	45
10^{-3}	115	92	119	39	185	52	79	63	73
10^{-4}	363	313	357	71	267	130	161	131	97
10^{-5}	1191	948	1013	127	451	270	332	277	157

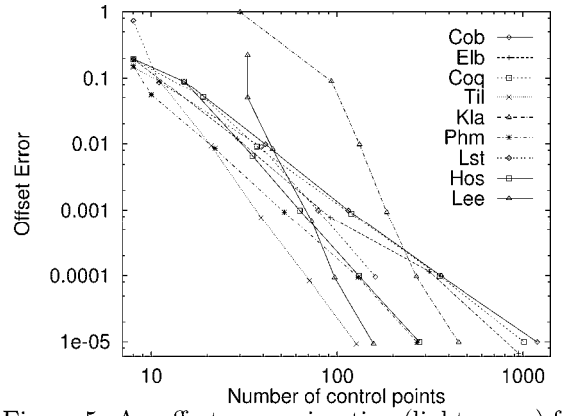
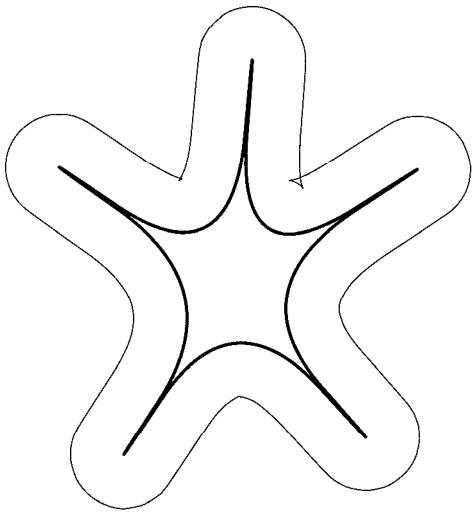


Figure 5: An offset approximation (light curve) for a quadratic polynomial B-spline curve with eight control points.

the local cubic B-spline interpolation method, *Pham*. Many practical situations require the production of local optimal solutions based only on the local data that is available. For example, for data storage saving, we can store only the subdivision locations of the curve, instead of all the control points that are generated. We then use the local methods to generate the control points (on the fly) by considering only local data. In this case, *Hos* and *Pham* are the methods of choice.

As discussed in the above observation, the performance of *Pham* is closely related to the radius of curvature of the base curve. When the radius of curvature is similar to the offset radius, the sample offset points are clustered together. The B-spline interpolation of these clustered points generates undulation, which is the main source of large approximation er-



ϵ	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
10^{-0}	12	12	29	73	21	12	30	12	93
10^{-1}	127	22	127	95	167	78	58	129	101
10^{-2}	223	162	219	155	239	140	362	185	169
10^{-3}	527	436	591	217	399	268	654	337	237
10^{-4}	1497	1316	1803	313	783	530	1234	659	397
10^{-5}	4763	3878	5587	505	1651	1162	1650	1367	681

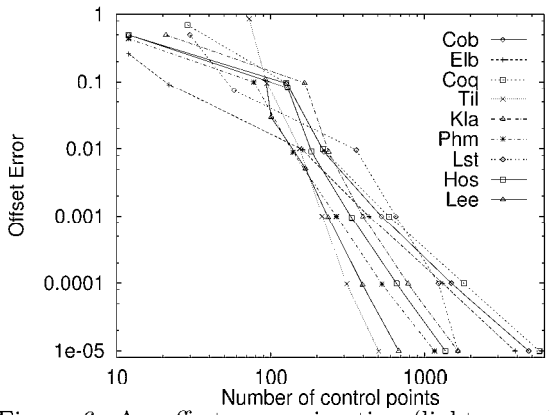
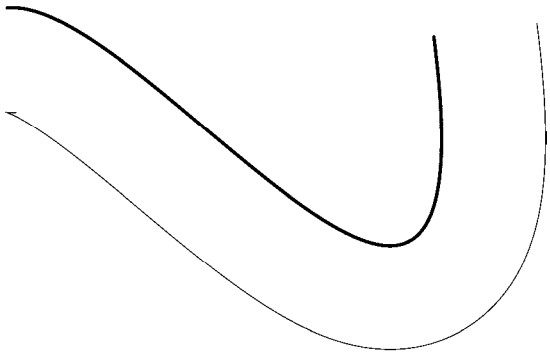


Figure 6: An offset approximation (light curve) for a quadratic polynomial with sharp corners.



ϵ	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
10^{-0}	4	4	4	7	4	4	4	4	15
10^{-1}	10	9	10	10	13	13	8	10	15
10^{-2}	31	22	25	22	22	19	17	22	36
10^{-3}	73	53	58	73	43	31	26	31	43
10^{-4}	226	156	175	226	76	79	42	64	78
10^{-5}	679	490	538	715	139	157	68	106	127

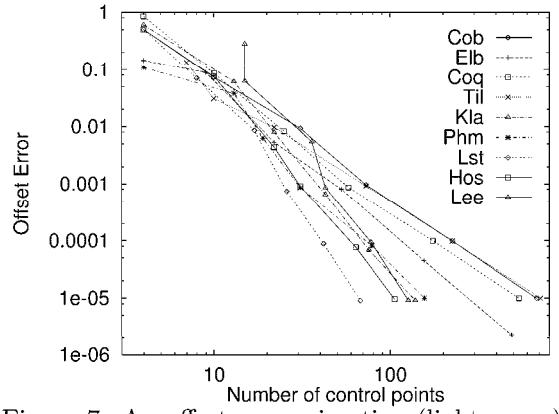
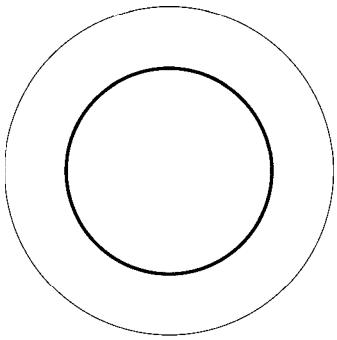


Figure 7: An offset approximation (light curve) for a polynomial cubic Bézier curve.

ror. In this case, it is better to use a smaller number of data points for the interpolation. Figures 11–12 exemplify this phenomenon by comparing the relative performances of different offset approximation methods. Given a fixed base curve, by increasing the offset radius gradually, we can observe that Pham's method has the worst relative performance near the offset distance which starts to develop cusps in the offset curve.

There is another source of undulation we have to consider in Pham's method. That is, the mismatch in speeds between the two curves, i.e., the base curve and the offset curve, also cause deterioration in the quality of the offset approximation. For the implementation of *Pham*, we use a non-uniform cubic B-spline curve, in which the data points of the offset inherit the knot values of the base curve points. When the offset data points are clustered, their knot values are much sparser compared with the offset curve length. This unnatural assignment of knot values generates undulation. Therefore, for a better offset approximation, it is also important to rearrange the knot values of the offset data points.

The superior performance (in the quadratic case) of the simple method, *Til*, suggests the possibility of improvement over the current least squares methods. This improvement may be achieved by resolving the limitations of least squares methods as discussed above. The limitation resulting from the L_2 norm seems more serious. To resolve this problem, we need to develop an efficient algorithm to compute and optimize the L_∞ norm of the offset approxima-



ϵ	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
10^{-0}	9	9	9	9	10	9	9	9	25
10^{-1}	17	13	9	9	17	9	9	9	25
10^{-2}	33	53	9	9	49	9	13	17	61
10^{-3}	129	69	9	9	113	47	24	49	73
10^{-4}	497	261	9	9	417	107	46	97	121
10^{-5}	1025	524	9	9	1345	221	98	209	229

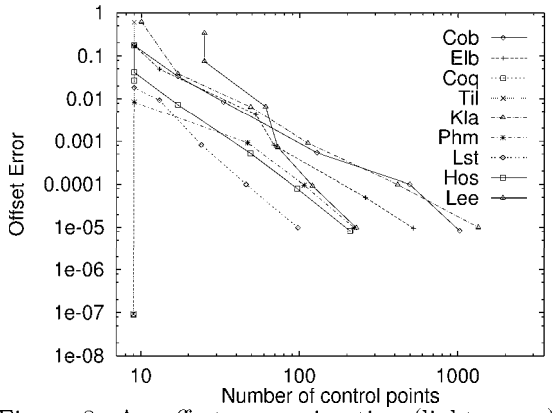


Figure 8: An offset approximation (light curve) for a rational quadratic B-spline circle with nine control points.

tion error; that is, the maximum of the error function $\epsilon(t)$ (in Equation (3)):

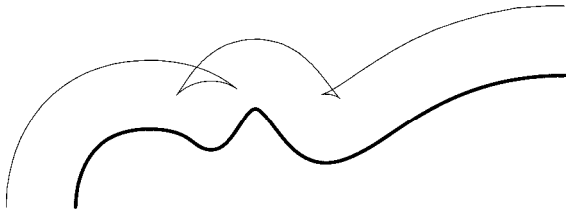
$$\max_t \left\{ \|C(t) - C_d^a(t)\|^2 - d^2 \right\}, \qquad (12)$$

or a more precise geometric distance measure based on the following Hausdorff metric:

$$\max \left(\max_t \min_s \left\{ \|C(s) - C_d^a(t)\|^2 - d^2 \right\}, \qquad (13) \right.$$
$$\left. \max_s \min_t \left\{ \|C(s) - C_d^a(t)\|^2 - d^2 \right\} \right)$$

where s is assumed to be a local perturbation of the parameter t.

Note that the method of Cobb [10] essentially models the L_∞ norm of Equation (12) in terms of the maximum and minimum magnitudes of the distance curve, $D(t) = C(t) - C_d^a(t)$, in Figures 2–4. Let's consider a variant of Cobb [10] which uses the least squares technique to optimize the offset distance at



ϵ	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
10^{-0}	13	13	13	13	386	13	13	13	99
10^{-1}	27	16	24	13	415	16	14	27	99
10^{-2}	69	67	60	51	418	67	37	57	127
10^{-3}	177	236	165	171	445	112	66	99	162
10^{-4}	543	380	525	546	496	181	109	183	246
10^{-5}	1746	1129	1518	1788	607	295	180	312	365

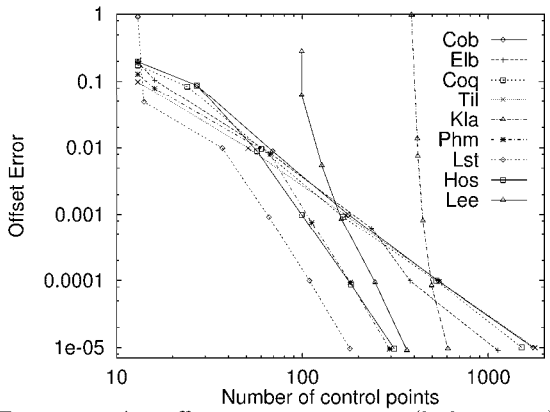
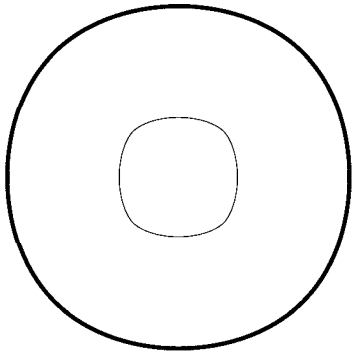


Figure 9: An offset approximation (light curve) for a cubic polynomial B-spline curve with 13 control points.

each control point so that the distance curve $D(t)$ is a best fit to the offset circle of radius d. This method measures the offset error in the L_∞ sense of Equation (12). (Note that the approximation of $D(t)$ to an offset circle still has the limitation of L_2 norm.) Although we have not provided all the details of *Lee* in this article, the method of Lee et al. [5] actually measures the offset approximation error under the L_∞ norm of Equation (13), which is more precise than the L_∞ norm of Equation (12). We expect that future offset approximation techniques (while incorporating these L_∞ norms into their optimization procedures) may provide more accurate results than the current least squares methods.

Conclusion

We have compared several contemporary offset approximation techniques for freeform curves in the plane. In general, the least squares methods per-



ϵ	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
10^{-0}	7	7	7	7	13	7	7	7	50
10^{-1}	9	7	13	7	13	7	7	9	57
10^{-2}	25	27	49	25	25	19	10	25	71
10^{-3}	97	143	145	97	49	52	19	49	99
10^{-4}	385	147	433	385	73	82	35	49	148
10^{-5}	1033	742	1537	1081	121	121	69	97	239

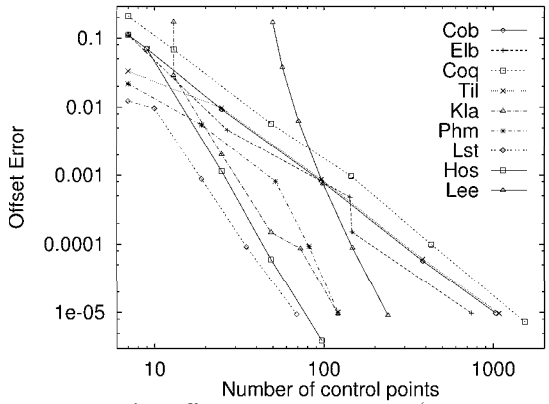
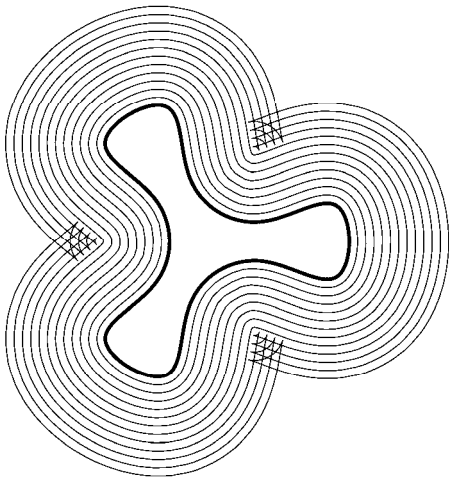


Figure 10: An offset approximation (light curve) for a cubic polynomial periodic B-spline curve with four control points.



d	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
0.1	531	483	417	519	217	198	181	237	288
0.2	765	660	477	759	253	258	217	279	330
0.3	909	867	723	927	307	333	217	297	351
0.4	1029	933	783	1029	325	333	253	315	358
0.5	1113	978	843	1125	325	396	253	327	379
0.6	1371	1197	867	1377	325	399	271	327	400
0.7	1431	1269	903	1449	325	366	307	345	442
0.8	1503	1344	921	1515	361	390	307	345	442
0.9	1569	1416	927	1569	361	423	325	363	463
1.0	1647	1491	945	1695	379	423	325	375	463
1.1	1749	1641	1095	1761	397	429	325	399	463
1.2	1821	1740	1401	1845	397	447	325	399	463

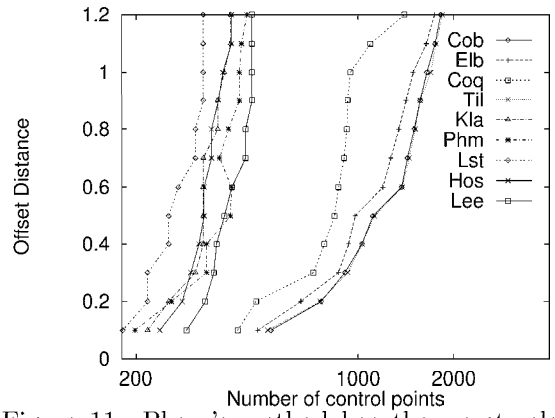
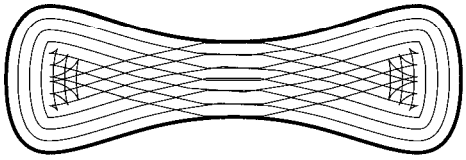


Figure 11: Pham's method has the worst relative performance for the offset distances between 0.4 and 0.6. Base curve is a cubic B-spline curve. Tolerance of 0.0001 is used for offset approximation error.

form very well. However, for the case of offsetting quadratic curves, the simple method of Tiller and Hanson [6] is the method of choice. Therefore, the least squares methods need further improvement to produce near-optimal solutions in all cases. Some of the current methods [5, 7, 10] have geometric representations of the offset approximation error (in certain L_∞ norms), whereas none of the current least squares methods have such geometric interpretation of their respective error bounds. We also pointed out two limitations of the current least squares methods: (i) the L_2 norm employed in these methods and (ii) the dependency on the finite sample points used in the optimization. The B-spline interpolation method also needs further investigation to eliminate the curve undulation resulting from the curve speed mismatch between the base curve and the offset curve. In this respect, there are still many ways to improve the current state-of-the-art of offset curve approximation. We hope that the experimental results reported in this article and the related remarks will serve as useful guidelines for future research.

References

[1] Hoschek, J., (1988), "Spline Approximation of Offset Curves," *Computer Aided Geometric Design*, Vol. 5, pp. 33–40.



d	Cob	Elb	Coq	Til	Kla	Phm	Lst	Hos	Lee
0.1	279	256	153	279	127	178	115	129	190
0.2	369	337	195	351	163	181	115	165	197
0.3	483	424	255	483	163	196	127	171	204
0.4	549	499	267	579	175	202	139	189	204
0.5	627	547	303	621	187	175	151	207	218
0.6	663	580	333	657	199	187	163	213	232

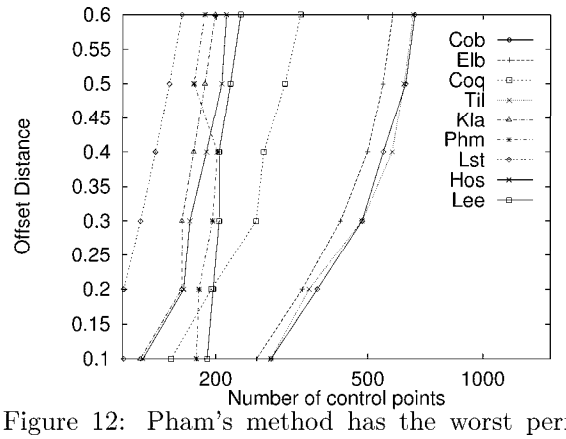


Figure 12: Pham's method has the worst performance for the offset distances around 0.4. Base curve is a cubic B-spline curve. Tolerance of 0.001 is used for offset approximation error.

[2] Hoschek, J., and Wissel, N., (1988), "Optimal Approximate Conversion of Spline Curves and Spline Approximation of Offset Curves," *Computer-Aided Design*, Vol. 20, No. 8, pp. 475–483.

[3] Farouki, R., and Neff, C., (1990), "Analytic Properties of Plane Offset Curves" & "Algebraic Properties of Plane Offset Curves," *Computer Aided Geometric Design*, Vol. 7, pp. 83–99 & pp. 101–127.

[4] Farouki, R., and Sakkalis, T., (1990), "Pythagorean Hodograph," *IBM J. Res. Develop.*, 34, pp 736-752.

[5] Lee, I.-K., Kim, M.-S., and Elber, G., (1995), "Planar Curve Offset Based on Circle Approximation," to appear in *Computer-Aided Design*.

[6] Tiller, W., and Hanson, E., (1984), "Offsets of Two Dimensional Profiles," *IEEE Computer Graphics & Application*, Vol. 4, pp. 36–46.

[7] Elber, G., and Cohen, E., (1991), "Error Bounded Variable Distance Offset Operator for Free Form Curves and Surfaces," *International Journal of Computational Geometry and Applications*, Vol. 1, No. 1, pp. 67–78.

[8] Elber, G., (1992). "Free Form Surface Analysis using a Hybrid of Symbolic and Numeric Computation," Ph.D. thesis, University of Utah, Computer Science Department.

[9] R. T. Farouki and V. T. Rajan. Algorithms For Polynomials In Bernstein Form. Computer Aided Geometric Design 5, pp 1-26, 1988.

[10] Cobb, B., (1984), *Design of Sculptured Surfaces Using The B-spline Representation*, Ph.D. thesis, University of Utah, Computer Science Department.

[11] Coquillart, S., (1987), "Computing Offset of B-spline Curves," *Computer-Aided Design*, Vol. 19, No. 6, pp. 305–309.

[12] Elber, G., and Cohen, E., (1992), "Offset Approximation Improvement by Control Points Perturbation." *Mathematical Methods in Computer Aided Geometric Design II*, Tom Lyche and Larry L. Schumaker, Academic Press, pp 229-237.

[13] Klass, R., (1983), "An Offset Spline Approximation for Plane Cubic Splines," *Computer-Aided Design*, Vol. 15, No. 4, pp. 297–299.

[14] Pham, B., (1988), "Offset Approximation of Uniform B-splines," *Computer-Aided Design*, Vol. 20, No. 8, pp. 471–474.

[15] Lee, I.-K., Kim, M.-S., and Elber, G., (1995), "Planar Curve Offset by Curve Tangent Matching," Technical Report, CS-CG-TR-95-007, Dept. of Computer Science, POSTECH, November, 1995.

[16] IRIT 6.0 User's Manual, February 1996, Technion.