

# Comparison of Three Bounding Regions with Cubic Convergence to Planar Freeform Curves

Jaewook Lee · Yong-Joon Kim · Myung-Soo Kim · Gershon Elber

**Abstract** We compare the relative performance of bounding regions generated by three different curve bounding methods with cubic convergence to planar freeform curves: spiral fat arcs (SFA) [4], bilens [17], and bounding circular arcs (BCA) [21]. For quantitative comparison, we consider three different criteria: geometric complexity (the number of circular arcs and line segments), construction time, and numerical stability. The BCA construction after one step refinement (producing four circular arcs) is almost comparable to the other two methods in geometric complexity: the spiral fat arcs with two circular arcs and two line segments, and the bilens with four circular arcs. In other comparison criteria, the BCA approach is more efficient and stable than the other two methods in producing a hierarchy of bounding regions that approximate a family of freeform planar curves within a given error bound.

**Keywords** Planar freeform curves · monotone spiral curve · bounding circular arcs · dynamic BVH

## 1 Introduction

Bounding volume hierarchy (BVH) is one of the most powerful spatial data structures for accelerating algorithms in computer graphics and geometric modeling [1,

7,8]. For the design of efficient geometric algorithms for freeform NURBS surfaces, the Coons BVH construction (based on Coons patch approximation [6]) is highly effective, often demonstrating interactive-speed performance in computing minimum and Hausdorff distances and convex hulls [12,14,15]. Coons patches approximate NURBS surfaces considerably more tightly and thus have a higher approximation order than other regular shapes such as spheres and boxes.

Based on the concept of fat-arcs originally proposed by Sederberg et al. [24], the  $G^1$ -biarc approximation has been used for the acceleration of geometric algorithms for freeform planar curves, and produced many fruitful results that are considerably more robust than previous solutions for non-trivial problems such as medial axis and Voronoi diagram construction and offset curve approximation and trimming [2,3,11,13,25]. The improved robustness is in some sense a direct consequence of higher approximation order – the round-off error accumulates slowly as the curve subdivision would terminate in an early stage of geometric computation.

The cubic convergence of biarc approximation [21] is the main factor in the improvement of computational efficiency and robustness. There are other curve approximation schemes with cubic convergence to planar freeform curves [4,17]. Thus it is important to compare their relative performance. In the proposal of fat arcs, Sederberg et al. [24] also made an extensive comparison against other conventional bounding regions (such as convex hull and bounding boxes) in terms of bounding area and construction speed. In many geometric applications, the thickness of the bounding region seems more relevant than area – the thickness is directly related to the Hausdorff distance error. In this paper, we measure the relative performance of different approximation methods in terms of computing speed and

---

Jaewook Lee

Department of Electrical and Computer Engineering, Seoul National University, Seoul 151-744, Korea

Yong-Joon Kim

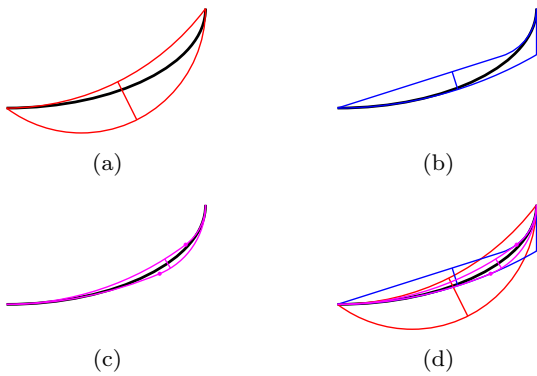
Computer Science Department, Technion, Haifa 32000, Israel

Myung-Soo Kim (Corresponding Author)

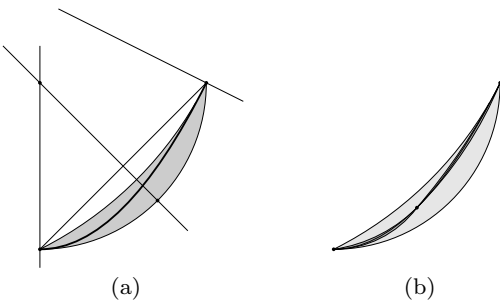
Department of Computer Science and Engineering, Seoul National University, Seoul 151-744, Korea

Gershon Elber

Computer Science Department, Technion, Haifa 32000, Israel



**Fig. 1** (a) Bounding circular arcs, (b) spiral fat arcs, (c) bilens, and (d) an overlap of three bounding regions.



**Fig. 2** (a) Bounding circular arcs for  $C(t) = (t, t^2)$ ,  $0 \leq t \leq 1$ , (b) tighter BCAs after one step refinement by subdividing at  $t = \frac{1}{2}$ .

numerical stability in bounding planar freeform curves within a given error bound.

Fat arcs are often generated in the form of fat biarcs [20, 26] as the  $G^1$ -biarc approximation has cubic convergence to planar curves. Nevertheless, it is time-consuming to measure the maximum deviation of biarcs from the given curve and thus to determine the thickness of fat arcs. In a recent work on trimming planar offset curves, Lee et al. [19] showed that the bounding circular arcs (BCA) of Meek and Walton [21] is considerably more efficient to generate and measure the error bound than fat arcs. In this paper, we compare the relative performance of the BCA bounding region against spiral fat arcs [4] (each constructed with two circular arcs and two line segments) and bilens [17] (with two biarcs, i.e., four circular arcs), all of which are planar curve approximation methods of cubic convergence.

Figure 1 shows an example of comparing three different bounding regions for the same curve, where the BCA bounding region is about twice thicker than spiral fat arcs and four times than bilens. It may sound that the BCA scheme is a loser among the three methods under comparison. Nevertheless, thanks to the cubic convergence rate, as long as the BCA construction is two times more efficient (in computing speed)

than the other two methods, the BCA construction for two half curves would generate bounding regions about four times thinner than spiral fat arcs and twice than bilens. The two-BCA approach makes sense for a fair comparison as two BCAs are represented with four arcs, the same number as for the bilens with four arcs. (Figure 2(b) illustrates the effect of a two-BCA construction; note that the two smaller BCAs can be constructed even without using the larger BCA for the whole curve.) In fact, the fat biarcs are thinner than BCA. On the other hand, BCA can be subdivided many times in the cycle where only a single fat biarc subdivision can be made. Thus the speed of construction is very important in the selection of an optimal bounding region.

Improving the numerical stability of each construction step is often more important than reducing the computing time – instability often leads to unreliable or even no solution at all. In the limiting case, bounding regions will be generated by tiny circular arcs and line segments meeting each other almost tangentially. In this respect, the BCA construction is more stable than the other two methods under comparison as it is based on the intersection of lines (see Figure 2(a)). On the other hand, spiral fat arcs are constructed by (i) computing a tangent line from a curve endpoint (of lower curvature) to the osculating circle at the other endpoint and (ii) intersecting the tangent line at an endpoint (of higher curvature) against the osculating circle at the other endpoint. Moreover, the bilens construction is based on intersecting a certain circle (formed by all possible joint locations for biarcs interpolating the given  $G^1$  data at both endpoints [25]) with two osculating circles at both endpoints. In the degenerate case of approximating an almost circular curve with bilens, one may end up with intersecting almost identical circles all the time, which can be highly unstable numerically.

The rest of this paper is organized as follows. In Section 2, we briefly review related previous work. Section 3 presents elementary construction methods for three bounding regions, each applied to a simple example  $C(t) = (t, t^2)$ ,  $0 \leq t \leq 1$ , so that the readers may get some intuitive understanding on the operation counts and numerical stability involved in each method. Some experimental results are reported in Section 4, and finally, Section 5 concludes the paper.

## 2 Related Work

For freeform curves and surfaces, the convex hulls of their control polygons and polyhedra have been conventionally used for the design of geometric algorithms.

Sederberg et al. [24] pointed out the limitation of convex hull as a bounding volume for freeform geometric models and suggested fat arcs as a bounding region that is considerably tighter than convex hulls.

It is quite amazing that the important concept of fat arcs, originally proposed by Sederberg et al. [24] in 1989, has been around in the research area of geometric modeling and processing, for more than 25 years. This is even a longer history than some of the early developments of bounding volume hierarchies (mostly in the mid 90s) such as sphere tree [9,22,23], BOXTREE [5], AABB (Axis-Aligned Bounding Box) tree [27], OBB (Oriented Bounding Box) tree [8], k-DOP (k-Discrete Oriented Polytope) [16], SSV (Swept Sphere Volume) tree [18], etc. Nevertheless, it is a relatively recent trend that fat arcs are extensively used in the acceleration of geometric algorithms for freeform planar curves [2,3,10,13].

The concept of bounding circular arcs (BCA) also has a relatively long history of 20 years. In 1995, Meek and Walton [21] introduced the concept of BCA as a handy tool for proving the cubic convergence of  $G^1$  biarc approximation of planar B-spline curves. To the best of our knowledge, this bounding volume has never been used for the purpose of accelerating geometric algorithms until Lee et al. [19] first used BCA as a tool for speeding up the offset trimming algorithm for planar freeform curves. We believe that BCA has a great potential in accelerating many other geometric algorithms for planar curves, which is the main motivation of the current work.

Barton and Elber [4] introduced spiral fat arcs (SFA) for bounding planar freeform curves more tightly than fat arcs. The inner boundary consists of a segment of the osculating circle at the endpoint of maximum curvature and a tangent line segment of the circle to the other endpoint. The line segment interpolates the position but not the tangent direction of the curve at the other endpoint. The outer boundary interpolates the positions and tangent directions at both endpoints. The construction starts with the tangent line at the endpoint of maximum curvature and the osculating circle at the other endpoint of minimum curvature, and intersects them to get the junction point, where the outer boundary has the position continuity, but no  $G^1$ -continuity in general.

The bilens recently developed by Kumosenko [17] can resolve the  $G^1$ -discontinuity of the spiral fat arcs using contact circles at the endpoints and making them smoothly connected with  $G^1$ -continuity to the osculating circles at the other endpoints. The bilens construction generates the tightest bounding regions than any other methods based on arcs and biarcs. Nevertheless,

the bilens construction takes more computing time than others. Thus, we need to make detailed comparisons among different curve approximation methods, which is the main goal of this paper.

### 3 Construction Methods for Bounding Regions

For the construction of three different bounding regions under comparison, in a preprocessing step, we need to subdivide a planar freeform curve  $C(t) = (x(t), y(t))$ ,  $0 \leq t \leq 1$ , into  $x$ - and  $y$ -monotone spiral curve segments. This can be done by computing all solutions of  $x'(t) = 0$  (i.e.,  $x$ -extremes),  $y'(t) = 0$  ( $y$ -extremes),  $x'(t)y''(t) - x''(t)y'(t) = 0$  (inflections), and  $\kappa'(t) = 0$  (curvature extremes), where  $\kappa(t)$  denotes the curvature of  $C(t)$ :

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

Thus, in the rest of this paper, we assume that  $C(t)$ ,  $0 \leq t \leq 1$ , is a planar freeform curve which is monotone with respect to coordinate axes, tangent angle (as being an inflection-free curve), and curvature.

For the design of efficient geometric algorithms for planar curves, we usually organize a simple global hierarchy of bounding boxes. Each monotone spiral curve segment  $C(t)$  is first contained in an axis-aligned bounding box (AABB) that is determined by the two end points  $C(0)$  and  $C(1)$ . A family of such planar curves can then be organized hierarchically in an AABB tree which is constructed in a bottom-up fashion starting with AABBs, each bounding one monotone spiral curve  $C(t)$ . Each leaf node of the AABB tree contains a single curve  $C(t)$  which is then represented in a separate bounding volume hierarchy (BVH) using each of the bounding regions under comparison in this paper.

We review the construction for each bounding region in an elementary manner. The purpose of this presentation style is to make the operation counts and the associated numerical stability clear as they have direct consequences to the performance of each construction algorithm. Geometric computations on circles are all carried out by constructing points, lines and vectors instead of algebraic manipulations. For the sake of explicit illustration of concrete examples of points, lines, vectors, circles, and distances, we employ a simple example:  $C(t) = (t, t^2)$ ,  $0 \leq t \leq 1$ , for explaining the detailed construction steps of each bounding region.

Note that  $C(t)$  is a planar quadratic curve segment which is monotone along  $x$  and  $y$ -directions and free of

inflection and curvature extreme points on the interval  $[0, 1]$ . The curvature function  $\kappa(t)$  is given as follows:

$$\kappa(t) = \frac{2}{(1 + 4t^2)^{3/2}},$$

which is monotonically decreasing along the curve. The osculating circle  $O$  of  $C(t)$  at the endpoint  $C(0) = (0, 0)$  of maximum curvature has radius  $1/\kappa(0) = \frac{1}{2}$  and center  $(0, \frac{1}{2})$ :

$$x^2 + \left(y - \frac{1}{2}\right)^2 = \frac{1}{4}.$$

As the curve  $C(t)$ ,  $0 \leq t \leq 1$ , has the maximum curvature at  $C(0)$ , the whole curve segment is contained in the exterior of the osculating circle:

$$t^2 + \left(t^2 - \frac{1}{2}\right)^2 = t^2 + t^4 - t^2 + \frac{1}{4} = t^4 + \frac{1}{4} \geq \frac{1}{4},$$

where the equality holds only at the curve endpoint  $C(0)$ . Similarly, the osculating circle  $\hat{O}$  at the other endpoint  $C(1)$  has radius  $1/\kappa(1) = \frac{5\sqrt{5}}{2}$  and center  $(-4, \frac{7}{2})$ , and completely contains the curve  $C(t)$  in its interior.

### 3.1 Bounding circular arcs

Starting from the osculating circle  $O = \bar{O}_{\frac{1}{2}}$  with radius  $\frac{1}{2}$  at  $C(0)$ , as we increase the radius of the contact circle  $\bar{O}_r$  at  $C(0)$  from  $r = \frac{1}{2}$  to  $r = \frac{1}{2} + \Delta r$ , the enlarged circle  $\bar{O}_{\frac{1}{2} + \Delta r}$  will intersect the curve  $C(t)$  transversally at  $C(\sqrt{2\Delta r})$  (and tangentially at  $C(0)$ ). When the radius is increased up to  $r = \frac{1}{2} + \Delta r = 1$ , the contact circle  $\bar{O}_1$  with the following implicit equation:

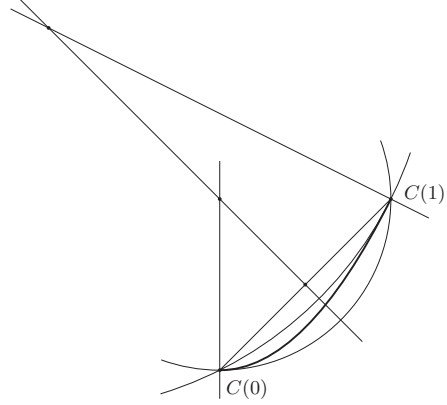
$$x^2 + (y - 1)^2 = 1,$$

will pass through the other endpoint  $C(1)$  and completely contains the whole curve  $C(t)$ ,  $0 \leq t \leq 1$ , in its interior:

$$t^2 + (t^2 - 1)^2 = t^2 + t^4 - 2t^2 + 1 = t^2(t^2 - 1) + 1 \leq 1,$$

where the equality holds only at the two curve endpoints  $C(0)$  and  $C(1)$ .

The contact circle  $\bar{O}_1$  is called the outer bounding circle of  $C(t)$  (see Figure 3). As the circle is in tangential contact with the curve  $C(t)$  at  $C(0)$ , the circle center  $(0, 1)$  is located on the normal line of  $C(t)$  at  $C(0) = (0, 0)$  with an implicit equation:  $x = 0$ . Moreover, the center  $(0, 1)$  is also located on the bisector line:  $x + y = 1$ , between two curve endpoints  $C(0) = (0, 0)$  and  $C(1) = (1, 1)$ . This is because the line segment  $\overline{C(0)C(1)}$  is a chord to the circle  $\bar{O}_1$  and thus the circle center should be located on the bisector line of the chord.



**Fig. 3** Construction of inner and outer bounding circles.

Similarly, starting with the osculating circle  $\hat{O}$  of  $C(t)$  at the other endpoint  $C(1)$ , we can reduce the radius of the contact circle while maintaining tangential contact at  $C(1)$  until the circle passes through  $C(0)$ . The result is the inner bounding circle of  $C(t)$ . The center of the inner bounding circle can be constructed by intersecting the bisector line:  $x + y = 1$  with the normal line:  $x + 2y = 3$  of  $C(t)$  at  $C(1) = (1, 1)$ , which produces  $(-1, 2)$ . Consequently, the inner bounding circle is implicitly defined as

$$(x + 1)^2 + (y - 2)^2 = 5.$$

Note that the curve  $C(t)$  is in the exterior of this circle:

$$\begin{aligned} & (t + 1)^2 + (t^2 - 2)^2 \\ &= t^2 + 2t + 1 + t^4 - 4t^2 + 4 \\ &= t^4 - 3t^2 + 2t + 5 = t(t - 1)^2(t + 2) + 5 \geq 5, \end{aligned}$$

where the equality holds only at the two endpoints  $C(0)$  and  $C(1)$ .

The distance between the two centers  $(-1, 2)$  and  $(0, 1)$  of bounding circles is  $\sqrt{2}$ , which means that the distance from the center  $(0, 1)$  (of the outer bounding circle) to the inner bounding circle is  $\sqrt{5} - \sqrt{2}$ . Consequently, the thickness of the BCA bounding region can be computed as  $1 + \sqrt{2} - \sqrt{5}$ . Though the center and two endpoints of a circular arc can determine the circular arc uniquely, we need to compute the circle radius for the purpose of measuring the upper bound of the curve approximation error. As to be discussed in the following two subsections, the circle radius information is also needed for the other two methods – the radius is used in the construction of junction points that appear in the middle of inner and outer boundaries of their bounding regions.

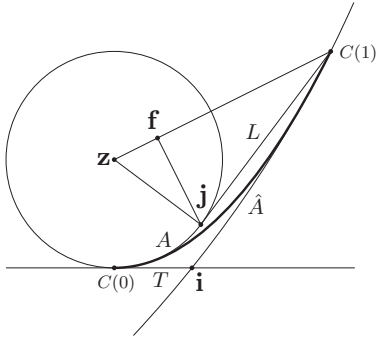
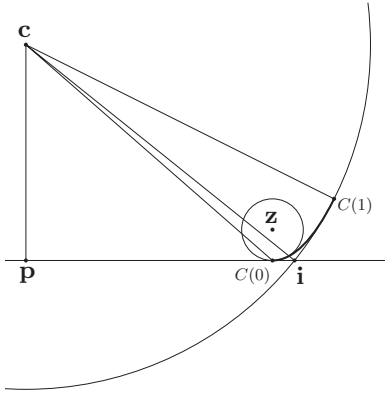


Fig. 4 Construction of spiral fat arcs.


 Fig. 5 Construction of the intersection point  $\mathbf{i}$ .

### 3.2 Spiral fat arcs

The construction of spiral fat arcs starts with the osculating circle  $O$  at the endpoint  $C(0) = (0, 0)$ , where the curvature is maximum. The osculating circle  $O$  has its center  $\mathbf{z} = (0, \frac{1}{2})$  and radius  $r = \frac{1}{2}$ . We can draw a tangent line segment  $L$  from the other endpoint  $C(1)$  to the osculating circle  $O$  (see Figure 4). Let  $d$  denote the distance from  $C(1) = (1, 1)$  to the center  $\mathbf{z} = (0, \frac{1}{2})$ :  $d = \sqrt{1^2 + (1 - \frac{1}{2})^2} = \frac{\sqrt{5}}{2}$ , then the tangent line segment  $L$  has its length:  $l = \sqrt{d^2 - r^2} = 1$ .

The inner boundary of the spiral fat arcs consists of a circular arc  $A$  that starts from  $C(0)$ , moves along the osculating circle  $O$  and ends at a tangential junction point  $\mathbf{j}$ , where the line segment  $L$  has a tangential contact with the osculating circle  $O$ . As shown in Figure 4, three points  $\mathbf{z}$ ,  $\mathbf{j}$ ,  $C(1)$  form a triangle, with a right angle at  $\mathbf{j}$ . When we draw an orthogonal line from  $\mathbf{j}$  to the footpoint  $\mathbf{f}$  on the line segment connecting  $\mathbf{z}$  and  $C(1)$ , the three points  $\mathbf{j}$ ,  $\mathbf{f}$ ,  $C(1)$  also form a right triangle. The footpoint  $\mathbf{f}$  and the junction point  $\mathbf{j}$  can be computed using the similarity relation between the two triangles  $\Delta(\mathbf{z}\mathbf{j}C(1))$  and  $\Delta(\mathbf{j}\mathbf{f}C(1))$ :

$$\mathbf{f} = C(1) + \frac{l^2}{d^2} \cdot (\mathbf{z} - C(1))$$

$$= (1, 1) + \frac{4}{5} \left( -1, -\frac{1}{2} \right) = \left( \frac{1}{5}, \frac{3}{5} \right),$$

$$\begin{aligned} \mathbf{j} &= \mathbf{f} + \frac{lr}{d^2} \cdot (\mathbf{z} - C(1))^\perp \\ &= \left( \frac{1}{5}, \frac{3}{5} \right) + \frac{2}{5} \left( \frac{1}{2}, -1 \right) = \left( \frac{2}{5}, \frac{1}{5} \right). \end{aligned}$$

where  $\mathbf{v}^\perp$  denotes the rotation of a vector  $\mathbf{v}$  by angle  $90^\circ$ .

The outer boundary of the spiral fat arcs consists of a tangent line segment  $T$  at  $C(0) = (0, 0)$ , connected through a transversal intersection point  $\mathbf{i}$  to a circular arc  $\hat{A}$  lying on the osculating circle  $\hat{O}$  at the other endpoint  $C(1)$ . The osculating circle  $\hat{O}$  has its center  $\mathbf{c} = (-4, \frac{7}{2})$  and radius  $\hat{r} = \frac{5\sqrt{5}}{2}$ . As shown in Figure 5, the construction of the intersection point  $\mathbf{i}$  proceeds as follows: (i) the center  $\mathbf{c}$  of  $\hat{O}$  is first projected to a point  $\mathbf{p}$  on the tangent line  $T$ , and (ii) the intersection point  $\mathbf{i}$  is reached by moving from  $\mathbf{p}$  along the tangent direction  $C'(0) = (1, 0)$  by a distance  $\sqrt{\hat{r}^2 - \hat{d}^2}$ , where  $\hat{d}$  is the distance from  $\mathbf{c}$  to  $\mathbf{p}$ .

The projection to  $\mathbf{p}$  can be computed by moving from  $\mathbf{c}$  along a displacement that is obtained by removing the tangential component along  $C'(0) = (1, 0)$  from the difference vector  $C(0) - \mathbf{c} = (4, -\frac{7}{2})$ :

$$\begin{aligned} \mathbf{p} &= \mathbf{c} + (C(0) - \mathbf{c}) - \frac{\langle C(0) - \mathbf{c}, C'(0) \rangle}{\langle C'(0), C'(0) \rangle} C'(0) \\ &= (0, 0) - (4, 0) = (-4, 0). \end{aligned}$$

Now, with the distance  $\hat{d} = \frac{7}{2}$  and the radius  $\hat{r} = \frac{5\sqrt{5}}{2}$ , we can compute the intersection point  $\mathbf{i}$  by moving from  $\mathbf{p}$  along the direction  $C'(0) = (1, 0)$  by the distance  $\sqrt{\hat{r}^2 - \hat{d}^2} = \sqrt{19}$ :

$$\begin{aligned} \mathbf{i} &= \mathbf{p} + \frac{\sqrt{\hat{r}^2 - \hat{d}^2}}{\|C'(0)\|} \cdot C'(0) \\ &= (-4, 0) + \sqrt{19}(1, 0) = (\sqrt{19} - 4, 0). \end{aligned}$$

It is a bit tricky to measure the thickness of an SFA region. In the example of Figure 4, the thickness is given as the distance from the intersection point  $\mathbf{i}$  to the inner osculating circle  $O$ . But, this is not always the case. In some other examples, the thickness may also be realized as the maximum deviation between the inner tangent line  $L$  and the outer circular arc on the osculating circle  $\hat{O}$ . Nevertheless, the thickness will never occur between the two osculating circular arcs, which greatly simplifies the case analysis.

### 3.3 Bilens

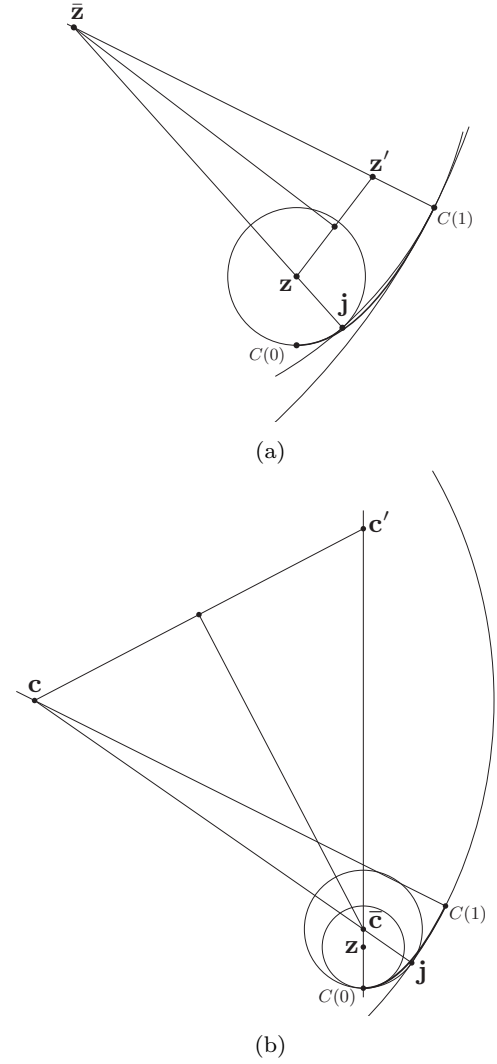
There is one parameter family of biarcs that interpolate the  $G^1$ -conditions (i.e., positions and tangent directions) of  $C(t)$  at both endpoints  $C(0)$  and  $C(1)$ . Among

all these biarcs, there are two special ones in the sense that they bound the monotone spiral curve  $C(t)$  most tightly from the inner and outer sides of the curve  $C(t)$ . We first consider the tightest biarc that bounds  $C(t)$  from the inner side. Again we start with the osculating circle  $O$  at the endpoint  $C(0)$ . A circular arc  $A$  of  $O$  that starts from  $C(0)$  should be connected smoothly at a junction point  $\mathbf{j}$  to another arc  $\bar{A}$  on a circle  $\bar{O}$  that tangentially touches the curve  $C(t)$  at the other endpoint  $C(1)$ . (Note that the contact circle  $\bar{O}$  is different from the osculating circle  $\hat{O}$  of  $C(t)$  at  $C(1)$ .) We need to determine the locations of the junction point  $\mathbf{j}$  and the center  $\bar{\mathbf{z}}$  of the tangent circle  $\bar{O}$ .

Note that the center  $\bar{\mathbf{z}}$  is located on the normal line:  $x + 2y = 3$  at  $C(1) = (1, 1)$  as we have  $C'(1) = (1, 2)$ . The osculating circle  $O$  is contained in a larger circle  $\bar{O}$  with a tangential contact at the junction point  $\mathbf{j}$ , where their arcs  $A$  and  $\bar{A}$  are smoothly connected with  $G^1$ -continuity. This means that the distance between  $\bar{\mathbf{z}}$  and  $\mathbf{j}$  is the same as the radius of the contact circle  $\bar{O}$ . But the locations of  $\bar{\mathbf{z}}$  and  $\mathbf{j}$  are yet unknown. We describe a simple geometric algorithm for computing them (also see Figure 6).

The radius of  $\bar{O}$  is the same as the distance between  $\bar{\mathbf{z}}$  and  $C(1)$ , which is also the distance between  $\bar{\mathbf{z}}$  and  $\mathbf{j}$ . The osculating circle  $O$  (contained inside  $\bar{O}$  with a tangential contact at  $\mathbf{j}$ ) has its center  $\mathbf{z} = (0, \frac{1}{2})$  located on the line segment connecting  $\bar{\mathbf{z}}$  and  $\mathbf{j}$  (as the two circles have a tangential contact at  $\mathbf{j}$ ). Though the location of  $\mathbf{j}$  is unknown, the distance between  $\mathbf{z}$  and  $\mathbf{j}$  is given as the radius  $\frac{1}{2}$  of the osculating circle  $O$ . Now, using this distance information, we construct a point  $\mathbf{z}' = (1 - \frac{\sqrt{5}}{5}, 1 + \frac{\sqrt{5}}{10})$  on the line segment connecting  $\bar{\mathbf{z}}$  and  $C(1)$  so that the distance between  $\mathbf{z}'$  and  $C(1)$  is the same as the radius  $\frac{1}{2}$  of the osculating circle  $O$ . (This can be done by moving from  $C(1)$  towards  $\bar{\mathbf{z}}$  by the radius  $\frac{1}{2}$ .) Then the center  $\bar{\mathbf{z}}$  of the contact circle  $\bar{O}$  is located on the bisector line between  $\mathbf{z} = (0, \frac{1}{2})$  and the new point  $\mathbf{z}' = (1 - \frac{\sqrt{5}}{5}, 1 + \frac{\sqrt{5}}{10})$ . The location can be computed as  $\bar{\mathbf{z}} = (-\frac{1+\sqrt{5}}{2}, \frac{7+\sqrt{5}}{4})$  by intersecting the bisector line:  $ax + by = c$ , with the normal line:  $x + 2y = 3$ , at  $C(1) = (1, 1)$ .

The outer bounding biarc can be constructed in a similar way by computing the center  $\bar{\mathbf{c}}$  of a tangent circle at  $C(0)$  that touches the osculating circle  $\hat{O}$  at  $C(1)$  from inside. The osculating circle  $\hat{O}$  has radius  $\frac{5\sqrt{5}}{2}$  and center  $\mathbf{c} = (-4, \frac{7}{2})$ . Along the normal line:  $x = 0$ , at  $C(0)$ , we construct a new point  $\mathbf{c}' = (0, \frac{5\sqrt{5}}{2})$  which is located at distance  $\frac{5\sqrt{5}}{2}$  (= the radius of  $\hat{O}$ ) from the endpoint  $C(0)$ . By intersecting the normal line  $x = 0$  with the bisector line between  $\mathbf{c}$  and  $\mathbf{c}'$ , we can



**Fig. 6** Construction of bilens: (a) inner bounding biarc, and (b) outer bounding biarc.

compute the location of  $\bar{\mathbf{c}}$ , which is the center of the tangent circle at  $C(0)$ . The junction point is located on the line connecting  $\mathbf{c}$  and  $\bar{\mathbf{c}}$ .

The thickness for a bilens bounding region is always realized as the maximum deviation between the two contact circular arcs other than those from the two osculating circles [17]. Consequently, the thickness computation can be done in a way similar to the BCA case.

## 4 Comparison of Relative Performance

### 4.1 Geometric complexity

When we subdivide a curve  $C(t)$  into two pieces, in all three methods, the point of subdivision is a common new endpoint to both the left and right subcurves. The BCA method thus computes one curve point and four

centers of bounding circles for both subcurves. On the other hand, the bilens method constructs five points and eight circle centers, and the spiral fat arcs method generates five points, four circle centers, and four line segments. In geometric complexity, the BCA approach needs to store four points and four lines less than the SFA and four points and four circles less than the bilens.

#### 4.2 Construction timing

Each time a monotone spiral curve  $C(t) = (x(t), y(t))$  is subdivided at the midpoint  $C(\frac{1}{2})$ , we need to evaluate the midpoint location  $C(\frac{1}{2}) = (x(\frac{1}{2}), y(\frac{1}{2}))$ , and the normal line at the midpoint by computing  $C'(\frac{1}{2}) = (x'(\frac{1}{2}), y'(\frac{1}{2}))$  and representing the normal line  $ax + by + c = 0$  using  $a = y'(\frac{1}{2})$ ,  $b = -x'(\frac{1}{2})$ , and  $c = x'(\frac{1}{2})y(\frac{1}{2}) - x(\frac{1}{2})y'(\frac{1}{2})$ . This part is common for all three different methods. For an efficient computation of  $C(t_i)$  and  $C'(t_i)$ , at all recursive subdivision parameters  $t_i = i/2^H$ , for  $i = 0, \dots, 2^H$ , where  $H$  is the maximum level of subdivision, we use precomputed basis function values  $B_k^d(t_i)$ , for  $k = 0, \dots, d$ . Thus, for a cubic Bézier curve,  $C(t_i)$  and  $C'(t_i)$  can be computed using four and three scalar-vector multiplications, respectively. This means that one normal line can be constructed using 16 multiplications and some additional arithmetic operations.

For the BCA construction, the normal line is first intersected with the bisector line  $B_0$  between  $C(0)$  and  $C(\frac{1}{2})$ , and then with the bisector line  $B_1$  between  $C(\frac{1}{2})$  and  $C(1)$ . Each bisector line  $B_j$  is also intersected with the corresponding normal line at  $C(j)$ , for  $j = 0, 1$ . In summary, we can compute the four centers of four bounding circles for the two subdivided curve segments, by evaluating two bisector lines and four line-line intersections. The thickness of each BCA bounding region can be determined by computing the radii of inner and outer bounding circles and the distance between their centers.

On the other hand, each subdivision of the bilens method computes one curvature, one vector normalization (for computing the unit vector  $C'(\frac{1}{2})^\perp / \|C'(\frac{1}{2})\|$ ), and one scalar-vector multiplication for constructing the center of the osculating circle at the curve subdivision point. After that, for the construction of four contact circles (such as  $\bar{O}$ ), we need to compute four auxiliary points (such as  $\mathbf{z}'$ , each using a scalar-vector multiplication), four bisector lines, four line-line intersections, and four radii of contact circles (each radius as the distance between the center and a point on the circle). In addition to these, the four junction points (such as  $\mathbf{j}$ ) are generated by computing (i) four unit direction vectors (each from the center of a contact circle toward

the junction point), and (ii) the four junction points (each by adding to the center the result of a scalar-vector multiplication). Moreover, two thickness computations are also needed for both the left and right bilens bounding regions. Now, it is clear that the BCA subdivision is more efficient than the bilens counterpart.

The SFA method also starts each subdivision by constructing a new osculating circle. Two inner junction points are then computed using two squared distances (such as  $d^2$ ), two distances (such as  $l$  using square roots), and four scalar-vector multiplications. Two outer junction points are also constructed using four vector normalizations, two inner products, two square roots, and four scalar-vector multiplications. Moreover, the thickness of two new SFA regions should be decided, which requires two point-to-circle projections and two point-to-line projections.

#### 4.3 Numerical stability

The BCA construction becomes unstable when the curvature drops to zero, namely, when the curve becomes flat to be almost like a line segment. In this case, the other two methods also have serious problem in stability – it is difficult to deal with osculating circles of almost infinite radius. The BCA approach may handle the almost linear case by replacing the inner bounding circle with the line connecting two endpoints and the outer bounding circle with the circle containing all control points of the curve  $C(t)$ .

In the construction of bilens bounding regions, a degenerate case is when the given curve  $C(t)$  has a constant curvature, namely, when the curve is a circular arc. In the almost-circular case, the two points  $\mathbf{z}$  and  $\mathbf{z}'$  get closer to an almost identical location, which makes the construction of their bisector line unstable. This case will eventually occur to all curve locations as we keep subdividing the given curve  $C(t)$  up to a certain level. When the difference  $\kappa(1) - \kappa(0)$  between the two extreme curvatures becomes smaller than a given tolerance  $\epsilon$ , we switch to the BCA approach, which produces more stable results.

The SFA method is quite stable even in the almost circular case as the construction is mainly based on moving points to well-defined directions. Nevertheless, there will be tiny line segments generated as a result of the degenerate SFA construction. These tiny pieces are still a potential source of system failure and an obvious source of degradation when the SFA tree is actually used for some specific geometric computations. Thus, we also need to switch to the BCA approach in the degenerate case.

## 5 Experimental Results

We have implemented the construction methods for three different bounding regions in C++ on an Intel Core i7-4770 3.4GHz PC with a 16GB main memory. The comparison was made on eight planar B-spline curves as shown in Figure 7. These are the same as the test examples of Kim et al. [13], which are uniform cubic B-spline curves defined in the normalized window of  $[-1, 1] \times [-1, 1]$ . In a preprocessing step, we have subdivided each of them into cubic Bézier curves and further into  $x$ - and  $y$ -monotone, inflection-free, and curvature monotone spiral curve segments. The numbers of cubic Bézier curves and monotone spiral pieces are reported in the first two rows of Table 1. (Note that the numbers are slightly different from those of Kim et al. [13], where spiral curves are not segmented into  $x$ - and  $y$ -monotone pieces.)

In this experiment, we have considered the construction of BVH trees based on three different bounding regions: BCA, Bilens, and SFA. The leaf nodes of each BVH tree approximate the original input curves within Hausdorff distance error bounds ( $\epsilon = 10^{-5}, 10^{-8}$  and  $10^{-10}$ ). The comparison results (computing time measured in *ms* and the number of circular arcs generated) are reported in Table 1. The computing time reported here includes none of the preprocessing time for the monotone spiral segmentation.

The construction of a BVH tree is the result of building various different components of the data structure, where the bounding regions form the major part, but not all. The construction of all other common components contributes almost equally to all three different methods. Thus, the timing results reported in Table 1 can be interpreted as an amortized performance of the bounding region generation together with all other construction components. The high-precision cases (with approximation errors  $\epsilon = 10^{-8}$  and  $10^{-10}$ ) report more reliable relative timing performance among the three different methods, where the BCA is about 1.5 times faster than the Bilens, and the Bilens is about twice faster than the SFA.

The number of circular arcs generated by each construction method is a good indicator to the total number of bounding regions generated for the BVH tree, but not to the amount of memory space for data storage. The number of BCA or SFA bounding regions is exactly the half of the number of circular arcs, whereas the number of bilens bounding regions is exactly one quarter of the number of circular arcs. The amount of memory space is a bit more complicated to estimate. The node size of each BVH tree is dependent on various optimization criteria for a specific algorithm. Nev-

ertheless, a rough estimation of memory space can be made by multiplying a certain weighting factor to the number of circular arcs in each method.

## 6 Conclusions

We have examined and showed that the BCA bounding region is an efficient and fairly stable way of generating a hierarchy of bounding regions for planar freeform curves. To demonstrate the effectiveness of the BCA tree, we have made comparison among three different curve approximation methods (BCA, Bilens, SFA) with cubic convergence to planar curves. The performance improvement of BCA over the other two recent methods was reported in terms of geometric data complexity, computational efficiency, and numerical stability, in each of which the BCA tree has relative advantages.

## Acknowledgments

This work was supported in part by the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme FP7/2007-2013/ under REA grant agreement PIAP-GA-2011-286426, in part by the Israel Science Foundation (Grant No.278/13), and also in part by the Korean MCST and KOCCA in the CT R&D Program 2014 (No. R2014060001), and in part by NRF Research Grants (No. 2013R1A1A2010085).

## References

1. T. Akenine-Möller, E. Hains, N. Hoffman: *Real-Time Rendering*, A.K. Peters, Natick, MA, 3rd Ed., 2008.
2. O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, M. Oberneder, B. Jüttler: Medial axis computation for planar free-form shapes, *Computer-Aided Design*, Vol. 41, No. 5, pp. 339–349, 2009.
3. O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, M. Rabl.: Divide-and-conquer algorithms for Voronoi diagrams revisited. *Computational Geometry: Theory and Applications*, Vol. 43, No. 8, pp. 688–699, 2010.
4. M. Barton and G. Elber: Spiral fat arcs – Bounding regions with cubic convergence. *Graphical Models*, Vol. 73, No. 2, pp. 50–57, 2011.
5. G. Barequet, B. Chazelle, L. Guibas, J. Mitchell, A. Tal: BOXTREE: A hierarchical representation for surfaces in 3D. *Computer Graphics Forum*, Vol. 15, No. 3, pp. 387–396, 1996.
6. S. Coons. *Surfaces for Computer-Aided Design*, Technical report, MIT, 1964. Available as AD 663 504 from the National Technical Information Service, Springfield, VA, 22161.
7. C. Ericson: *Real-Time Collision Detection*, Morgan Kaufmann, San Francisco, CA, 2005.



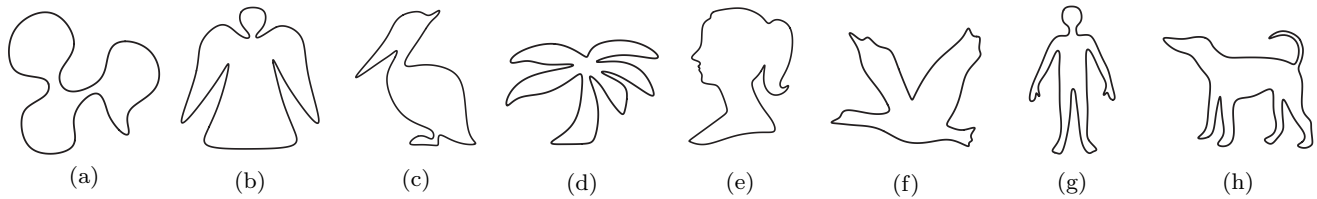


Fig. 7 Eight test examples of cubic B-spline planar curves.

Input Curves	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
#Cubic Bezier Curves	28	34	39	44	46	50	66	71
#Spiral Segments	75	85	105	115	123	132	169	182
	Time (ms)							
Precision: $10^{-5}$								
BCA	0.155	0.220	0.178	0.244	0.228	0.207	0.204	0.298
Bilens	0.266	0.286	0.321	0.386	0.348	0.355	0.368	0.480
SFA	0.485	0.552	0.550	0.665	0.561	0.621	0.736	0.817
Precision: $10^{-8}$								
BCA	1.400	1.820	1.721	2.350	1.934	2.096	2.547	2.784
Bilens	2.203	2.649	2.629	3.536	2.903	3.215	3.743	4.284
SFA	4.503	5.371	5.026	6.660	5.705	6.389	6.988	8.196
Precision: $10^{-10}$								
BCA	6.438	8.248	8.436	10.685	9.077	9.729	11.688	12.732
Bilens	9.849	12.675	11.684	16.486	13.460	14.819	17.644	19.489
SFA	21.259	25.043	23.701	32.169	28.824	28.988	34.888	38.600
	Space (#Arcs)							
Precision: $10^{-5}$								
BCA	1446	1774	1694	2298	1886	2060	2410	2700
Bilens	2150	2604	2430	3452	2718	2928	3514	3936
SFA	1898	2258	2114	2830	2370	2512	2970	3376
Precision: $10^{-8}$								
BCA	14998	19266	18234	24046	20366	22272	26774	29288
Bilens	24288	29426	28542	38568	31646	34918	41078	45680
SFA	22822	26926	26486	35430	29286	32168	37354	42452
Precision: $10^{-10}$								
BCA	69010	87902	86074	113854	95066	103404	124466	136340
Bilens	111246	142092	131024	179914	151122	163774	196882	218752
SFA	109026	137118	126134	174026	144386	157260	187602	208792

Table 1 Computing time and the number of circular arcs generated for the eight examples of Figure 7.

8. S. Gottschalk, M. Lin, D. Manocha: OBB-tree: a hierarchical structure for rapid interference detection. *Computer Graphics (SIGGRAPH 1996)*, pp. 171–180, 1996.
9. P. Hubbard: Collision detection for interactive graphics applications. *IEEE Trans. on Visualization and Computer Graphics*, Vol. 1, No. 3, pp. 218–230, 1995.
10. Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, and G. Elber. Precise Hausdorff distance computation for planar freeform curves using biarcs and depth buffer. *The Visual Computer*, Vol. 26, Nos 6–8, pp. 1007–1016, 2010.
11. Y.-J. Kim, J. Lee, M.-S. Kim, and G. Elber. Efficient convex hull computation for planar freeform curves. *Computers & Graphics*, Vol. 35, No. 3, pp. 698–705, 2011.
12. Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, and G. Elber. Coons BVH for freeform geometric models. *ACM Transactions on Graphics*, Vol. 30, No. 6, Article 169, SIGGRAPH Asia 2011.
13. Y.-J. Kim, J. Lee, M.-S. Kim, and G. Elber. Efficient offset trimming for planar rational curves using biarc trees. *Computer Aided Geometric Design*, Vol. 29, No. 7, pp. 555–564, 2012.
14. Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, and G. Elber. Efficient Hausdorff distance computation for freeform geometric models in close proximity. *Computer-Aided Design*, Vol. 45, No. 2, pp. 270–276, 2013.
15. Y.-J. Kim, M.-S. Kim, and G. Elber. Precise convex hull computation for freeform models using a hierarchical Gauss map and a Coons bounding volume hierarchy. *Computer-Aided Design*, Vol. 46, No. 2, pp. 252–257, 2014.
16. J. Klosowski, M. Held, J. Mitchell, H. Sowizral, K. Zikan: Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Trans. on Visualization and Computer Graphics*, Vol. 4, No. 1, pp. 21–37, 1998.
17. A. Kumosenko. Biarcs and bilens. *Computer Aided Geometric Design*, Vol. 30, No. 3, pp. 310–330, 2013.
18. E. Larsen, S. Gottschalk, M.C. Lin, D. Manocha. Fast proximity queries using swept sphere volumes. Technical Report TR99-018, Dept. of Computer Science, UNC, 1999.

19. J. Lee, Y.-J. Kim, M.-S. Kim, and G. Elber. Efficient off-set trimming for deformable planar curves using a dynamic hierarchy of bounding circular arcs. *Computer-Aided Design*, Vol. 58, pp. 248–255, 2015.
20. Q. Lin and J.G. Rokne. Approximation by fat arcs and fat biarcs. *Computer-Aided Design*, Vol. 34, No. 13, pp. 969–979, 2002.
21. D. Meek and D. Walton. Approximating smooth planar curves by arc splines. *J. of Computational and Applied Mathematics*, Vol. 59, No. 2, pp. 221–231, 1995.
22. I. Palmer, R. Grimsdale: Collision detection for animation using sphere-trees. *Computer Graphics Forum*, Vol. 14, No. 2, pp. 105–116, 1995.
23. S. Quinlan. Efficient distance computation between non-convex objects. *IEEE International Conference on Robotics and Automation*, pp. 3324–3329, 1994.
24. T.W. Sederberg, S.C. White, and A.K. Zundel. Fat arcs: A bounding region with cubic convergence. *Computer Aided Geometric Design*, Vol. 6, No. 3, pp. 205–218, 1989.
25. Z. Sir, R. Feichtinger, B. Jüttler: Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. *Computer-Aided Design*, Vol. 38, No. 6, pp. 608–618, 2006.
26. J.-H. Yong, X. Chen, J.-C. Paul: An example on approximation by fat arcs and fat biarcs. *Computer-Aided Design*, Vol. 38, No. 5, pp. 515–517, 2006.
27. G. van den Bergen: Efficient collision detection of complex deformable models using AABB trees. *J. of Graphics Tools*, Vol. 2, No. 4, pp. 1–14, 1997.



**Jaewook Lee** is a BS student in the Department of Electrical and Computer Engineering, Seoul National University, South Korea. His research interests are in geometric modeling and processing.

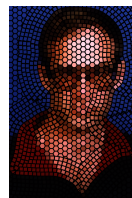


**Yong-Joon Kim** is a postdoctoral research fellow in the department of computer science, Technion, Israel. His research interests are computer aided geometric design, real-time 3D geometry processing and computer graphics. He received B.S. degree in mathematics education and Ph.D. degree in computer science from Seoul national university in 2008 and 2012, respectively.



**Myung-Soo Kim** received the BS and MS degrees from Seoul National University in 1980 and

1982, respectively, and the MS degree in applied mathematics and the MS and PhD degrees in computer science from Purdue University in 1985, 1987, and 1988, respectively. He is currently a professor of the Department of Computer Science and Engineering, Seoul National University, South Korea. His research interests are in computer graphics and geometric modeling. From 1988 to 1998, he was with the Department of Computer Science, POSTECH, Korea. He serves on the editorial boards of the journals *Computer-Aided Design*, *Computer Aided Geometric Design*, and the *International Journal of Shape Modeling*. Together with G. Farin and J. Hoschek, he edited the *Handbook of Computer Aided Geometric Design* (Amsterdam: North-Holland, 2002). In 2010, he served as a technical program co-chair of the ACM Symposium on Solid and Physical Modeling, Haifa, Israel. In 2011, he also served as a program co-chair of the SIAM Conference on Geometric and Physical Modeling, Orlando, USA.



**Gershon Elber** is a professor in the Computer Science Department, Technion, Israel. His research interests span computer aided geometric designs and computer graphics. Prof. Elber received a BSc in computer engineering and an MSc in computer science from the Technion, Israel in 1986 and 1987, respectively, and a PhD in computer science from the University of Utah, USA, in 1992. He is a member of SIAM and the ACM.

Prof. Elber has served on the editorial board of the *Computer Aided Design*, *Computer Graphics Forum*, *The Visual Computer*, *Graphical Models*, and the *International Journal of Computational Geometry & Applications* and has served in many conference program committees including *Solid Modeling*, *Shape Modeling*, *Geometric Modeling and Processing*, *Pacific Graphics*, *Computer Graphics International*, and *Siggraph*. Prof. Elber was one of the paper chairs of *Solid Modeling 2003* and *Solid Modeling 2004*, one of the conference chairs of *Solid and Physical Modeling 2010*, and the chair of *GDM 2014*. He has published over 200 papers in international conferences and journals and is one of the authors of a book titled "Geometric Modeling with Splines - An Introduction". Prof. Elber received the John Gregory Memorial Award, 2011, in "Appreciation for Outstanding Contributions in Geometric Modeling".