Skeleton-based Parametric 2D Region Representation: Disk B-spline Curves

Zhongke Wu, Xingce Wang, Shaolong Liu
School of Artificial Intelligence
Beijing Normal University
Beijing 100875, China

Quan Chen, Hock Soon Seah
School of Computer Science and Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798

Feng Tian
Faculty of Science and Technology
Bournemouth University, BH12 5BB, UK

***Abstract*—The skeleton, or medial axis, is an important attribute of 2D shapes. The disk B-spline curve (DBSC) is a skeleton-based parametric freeform 2D region representation, which is defined in B-spline form. The DBSC describes not only a 2D region, which is suitable for describing heterogeneous materials in the region, but also the center curve (skeleton) of the region explicitly, which is suitable for animation, simulation and recognition. In addition to being useful for error estimation of the B-spline curve, the DBSC can be used in designing and animating freeform 2D regions. Despite increasing DBSC applications, its theory and fundamentals have not been thoroughly investigated. In this paper, we discuss several fundamental properties and algorithms, such as the de Boor algorithm for DBSCs. We first derive the explicit evaluation and derivatives formulas at arbitrary points of a 2D region (interior and boundary) represented by a DBSC and then provide heterogeneous object representation. We also introduce modeling and interactive heterogeneous object design methods for a DBSC, which consolidates DBSC theory and supports its further applications.**

Index Terms: Disk B-Spline Curves; Skeleton; 2D Region; B-spline curve; Heterogeneous Object Representation;

■ **IN THE COMPUTER GRAPHICS COMMUNITY**, 2D region representation is fundamental and widely used in 2D animation, nonphotorealistic rendering (NPR) and geographical information systems. Discrete points and linear and parametric outline representations are most commonly used for 2D region representation. In [1], the interval Bézier curve was proposed to bound errors, while the disk Bézier curves were discussed in [2]. The disk Bézier curves are similar to the interval Bézier curve [1] but use disks instead of rectangles. Lin and Rokne [2] tried to assemble a disk B-spline by imposing conditions on two disk Bézier curves. They confined it to a special case where $r_0 = r_1 = \cdots = r_n = r = const$. Even for this special case, the conditions were complicated and difficult, if not impossible, to extend to higher degree cases. The two representations above are only regarded as parametric curves with error tolerances. We generalized the disk Bézier curve to a disk B-spline curve (DBSC) by directly defining the disk B-spline [3][4], which automatically maintains the $p-1$ (p is the degree of the curve) continuity between two curve pieces and the infinite continuity inside each piece. We used DBSC to directly represent and model freeform 2D regions instead of using raster images [5] or their outlines [6].

The skeleton is an object's important attribute. However, for almost all object representations, their skeletons are not directly represented but computed. However, in general, the computation of the medial axis transformation (MAT) is extremely complicated. In a DBSC, the skeleton is explicitly represented. Therefore, a DBSC can describe both the freeform 2D region and its skeleton. Given these advantages, DBSCs have been used to represent strokes in sketching [7], brushstrokes [8], Chinese calligraphy [9], 2D animation [10][11][12], and nonphotorealistic rendering [13]. A few papers have discussed the use of DBSC algorithms for extension [14], blending [15], fitting [16], intersection [17] and physically based simulations [18].

Heterogeneous object (HE object) modeling is paramount for natural object representation [19], such as tissue modeling, physically based anisotropic simulation [20], and additive manufacturing [21]. However, a 2D region is usually represented by its contours. Therefore, it only describes homogeneous objects. As DBSC can describe arbitrary points in a 2D region, it is suitable for representing HE objects. With its elegant mathematical definition, a DBSC is endowed with preeminent properties that make geometric operations of models based on DBSCs easy to implement. Therefore, DBSC representation has been applied to various areas, such as nonphotorealistic rendering [3], 2D modeling and animation in computer-assisted cel animation [4]. However, to the best of our knowledge, the theory and fundamentals of DBSCs have not yet been completely investigated. This is the focus of this paper. In addition to the fundamental properties of DBSCs, we derive explicit evaluation and derivative formulas for DBSCs, which include those on the boundary and arbitrary points inside a DBSC. We also provide basic modeling algorithms based on a DBSC, such as interpolation, approximation, deformation. Furthermore, we provide a representation method for HE objects in a DBSC and present an interactive design approach for HE objects. This paper provides the following contributions:

1) We provide a comprehensive discussion of fundamental DBSC properties and algorithms (Sec.2), such as the de Boor algorithm for DBSCs. And we derive explicit evaluations and derivative formulas on the boundary and at arbitrary points inside a 2D region represented by a DBSC (Sec.3). These benefit further developments and applications of DBSC.
2) We provide HE object representation on a DBSC and develop an interactive HE object design method. This lays the foundation for HE object simulation and additive manufacturing using a DBSC (Sec.4).

FUNDAMENTALS OF A DBSC

This section introduces the definitions, fundamental properties and algorithms for a DBSC.

Definitions of a DBSC Disk Geometry

Disk Geometry

A disk in the plane is defined as $< C; r > \equiv \{x \in R^2 \mid \|x - C\| \leq r, C \in R^2, r \in R^+\}$. Here C is the center of the disk and r is the radius. For disks, the following operations are defined:

- For an arbitrary real number, a $< C, r >= \langle aC, |a|r\rangle$.
- $< C_1, r_1 > + < C_2, r_2 >=< C_1 + C_2, r_1 + r_2\rangle$.

Therefore, any finite summation can be defined as: $\sum\limits_{i=1}^{n} a_i \langle C_i, r_i\rangle = \sum\limits_{i=1}^{n} \langle a_i C_i, |a_i|\, r_i\rangle = \left\langle \sum\limits_{i=1}^{n} a_i C_i, \sum\limits_{i=1}^{n} |a_i|\, r_i \right\rangle$. For more information on disk geometry, please refer to [2].

DBSC Definition

Let $N_{i,p}(t)$ be the i-th B-spline basis of degree p with knot vector $[t_0, \ldots, t_m] = \{\underbrace{a, \ldots, a}_{p}, t_{p+1}, \ldots, t_{m-p-1}, \underbrace{b, \ldots, b}_{p}\}$, where $m = n + p + 1$. Then, the disk B-spline Curve (DBSC) is defined as:

$$< D > (t) = \sum_{i=0}^{n} N_{i,p}(t) < P_i; r_i > \tag{1}$$

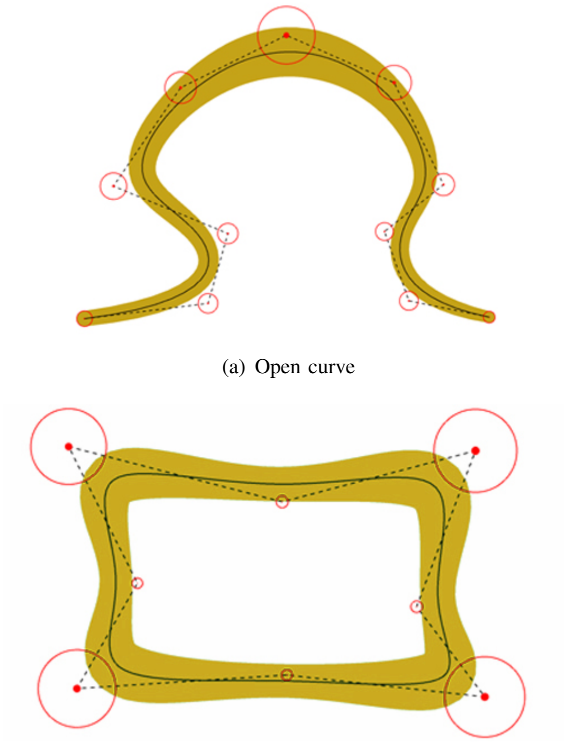
where P_i are the control points and r_i are the control radii. The DBSC represents a 2D region. For the detailed properties on a B-spline basis, please refer to [22]. As

$$\begin{aligned} &< D > (t) = \sum_{i=0}^{n} N_{i,p}(t) < P_i; r_i > \\ &= \sum_{i=0}^{n} < N_{i,p}(t)P_i; N_{i,p}(t)r_i > \\ &=< \sum_{i=0}^{n} N_{i,p}(t)P_i; \sum_{i=0}^{n} N_{i,p}(t)r_i > \end{aligned}$$

DBSC can be viewed as two parts: one is the center curve $c(t) = \sum_{i=0}^{n} N_{i,p}(t)P_i$, which is a B-spline curve, and the other is the radius function $r(t) = \sum_{i=0}^{n} N_{i,p}(t)r_i$, which is a B-spline scalar function. Owing to the perfect symmetry property of disks, the curve $c(t)$ constructed from the centers of disks is exactly the skeleton of the 2D region represented by the DBSC, as shown in Figure 1. Most properties and algorithms can be obtained by applying the B-spline curve and function to these two parts of the disk B-spline curve.

Properties

Affine invariance: An affine transformation is applied to the DBSC by applying it to the control points and radii. The affine invariance property for B-spline curves is derived from the partition property of unity of $N_{i,p}(t)$, i.e.,$\sum\limits_{i=0}^{n} N_{i,p}(t) = 1$.



(a) Open curve

(b) Closed curve

Figure 1: A disk B-spline curve: the 2D region (yellow) is represented by the DBSC; the curve (black) is the center curve; the polyline (dot line) and the circles (red) represent the control polygons and radii.

Differentiability: A disk B-spline curve of degree p is at least C^{p-k} at a knot of multiplicity k and infinitely differentiable in the interior of knot intervals.

Local modification scheme: Moving P_i or modifying r_i changes $< D > (t)$ only in the interval of $[\boldsymbol{u}_i, \boldsymbol{u}_{i+p+1}]$.

Convex hull property: The region represented by a disk B-spline is contained by the convex hull of those circles of radius r_i centered at P_i.
As $N_{i,p}(t) \geq 0$ and $\sum_{i=0}^{n} N_{i,p}(t) = 1$, for $\forall t$, $< \boldsymbol{D} > (\boldsymbol{t}) \subseteq conv\,\{< P_i; r_i >\}_{i=0}^{n}$. Here, $conv$ denotes a convex hull.

Topological property: From a topological viewpoint, the 2D region (white) represented by a DBSC without self-intersection is homotopic to its center curve (black), as shown in Figure 1. However, if self-intersection exists, the topological property is not maintained, as shown in

Figure 2.

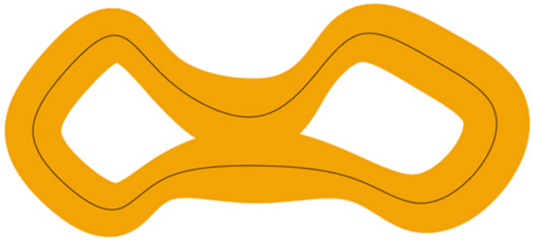


Figure 2: The center curve (black) is not homotopic to the 2D region (yellow) when there exist self-intersections.

Evaluation and Derivatives of DBSC

For a DBSC, most algorithms can be implemented by applying corresponding algorithms in the B-spline curve and function to the center curve and radius function, such as degree elevation, knot insertion, shape modification, transformation, etc. The following will discuss the evaluations and derivatives of a DBSC by extending the de Boor Algorithm to a DBSC.

De Boor Algorithm for DBSC: The de Boor Algorithm for a B-spline can be extended by applying it to the center curve and radius of a DBSC. Suppose a DBSC $< D > (t) = \sum_{i=0}^{n} N_{i,p}(t) < P_i; r_i >$, with knot vector $[t_0, \dots, t_m] = \{\underbrace{a, \dots, a}_{p}, t_{p+1}, \dots,$ $t_{m-p-1}, \underbrace{b, \dots, b}_{p}\}$, where $m = n + p + 1$.

Let $t \in [t_j, t_{j+1}]$, $\mathrm{j} = 0, 1, \dots, m-1$,

$$\lambda_i^k(t) = \frac{t - t_i}{t_{i+p-k} - t_i}, k = 1, 2, \dots, p-1 \quad (2)$$

$$P_i^k(\mathrm{t}) \\ = \begin{cases} P_i & , \; k = 0 \\ \lambda_i^k(t) P_i^{k-1}(t) + \left(1 - \lambda_i^k(t)\right) P_{i-1}^{k-1}(t), & \\ & k = 1, 2, \dots, p-1 \end{cases} \quad (3)$$

$$r_i^k(t) \\ = \begin{cases} r_i & , \; k = 0 \\ \lambda_i^k(t) r_i^{k-1}(t) + \left(1 - \lambda_i^k(t)\right) r_{i-1}^{k-1}(t), & \\ & k = 1, 2, \dots, p-1 \end{cases} \quad (4)$$

$\langle D \rangle (t) = \sum_{i=j-p+k+1}^{j} N_{i,p-k}(t) \langle P_i^k, r_i^k \rangle$, $k = 1, 2, \dots, p-1$. When $k = p-1$, $\langle D \rangle (t) = N_{j,1}(t) \left\langle P_j^{p-1}, r_j^{p-1} \right\rangle = \left\langle P_j^{p-1}, r_j^{p-1} \right\rangle$.

If these disks $\left\langle P_j^k, r_j^k \right\rangle, k = 0, 1, \dots, p$ and $\left\langle P_{j-k}^{p-k}, r_{j-k}^{p-k} \right\rangle, k = 0, 1, \dots, p-1$ are regarded as new control disks in the interval, DBSC is then subdivided at the parameter t.

DBSC Derivatives: The derivatives of a DBSC with respect to the parameter t can be computed as:

$$\begin{aligned} &< D' > (t) \\ &= \sum_{i=0}^{n} N'_{i,p}(t) < P_i; r_i > \\ &= p \left(\sum_{i=0}^{n-1} N_{i+1,p-1}(t) \frac{< P_{i+1}; r_{i+1} >}{t_{i+p+1} - t_{i+1}} \right) \\ &- p \left(\sum_{i=0}^{n} N_{i+1,p-1}(t) \frac{\langle P_i; r_i \rangle}{t_{i+p+1} - t_{i+1}} \right) \\ &= p \left(\sum_{i=0}^{n} N_{i+1,p-1}(t) \frac{< P_{i+1}; r_{i+1} > - < P_i; r_i >}{t_{i+p+1} - t_{i+1}} \right) \\ &= p \left(\sum_{i=0}^{n-1} N_{i+1,p-1}(t) \left\langle \frac{P_{i+1} - P_i}{t_{i+p+1} - t_{i+1}}; \frac{r_{i+1} - r_i}{t_{i+p+1} - t_{i+1}} \right\rangle \right) \\ &= p \left(\sum_{i=0}^{n-1} N_{i+1,p-1}(t) < Q_i; R_i > \right) \end{aligned} \quad (5)$$

here $Q_i = \frac{P_{i+1} - P_i}{t_{i+p+1} - t_{i+1}}; R_i = \frac{r_{i+1} - r_i}{t_{i+p+1} - t_{i+1}}$. Higher derivatives can be achieved similarly.

Other algorithms, such as degree elevation and knot insertion, can be obtained by applying the corresponding algorithms of the B-spline curve and function to the skeleton curve and radius function of the DBSC, respectively.

EVALUATION OF BOUNDARY AND INTERIOR

We will give implicit method and explicit formulas on boundary and interior.

Implicit Evaluations

To compute the boundary of a region represented by a DBSC, we can regard the boundary as the envelope of a one-parameter family of circles. For any t, with the center at $x(t) = \sum_{i=0}^{n} N_{i,p}(t) x_i$ and $y(t) = \sum_{i=0}^{n} N_{i,p}(t) y_i$, the radius $r(t) = \sum_{i=0}^{n} N_{i,p}(t) r_i$; then

$$(x - x(t))^2 + (y - y(t))^2 - (r(t))^2 = 0 \quad (6)$$

According to the envelope theorem [23],

$$\begin{cases} F(x,y,t)=0 \\ \frac{\partial F(x,y,t)}{\partial t}=0 \end{cases}, \text{ we have}$$

$$(x-x(t))x'(t)+(y-y(t))y'(t)+r(t)r'(t)=0 \tag{7}$$

For any t, we can compute $x(t)$, $y(t)$, $r(t)$ and their derivatives $x'(t)$, $y'(t)$, $r'(t)$ according to de Boor Algorithms for B-spline curves.

Let $X = x-x(t), Y = y-y(t), C = r(t), D = x'(t), E = y'(t)$, and $F = r'(t)$; we can obtain the following equation group of a circle and line intersection: $\begin{cases} X^2+Y^2=C^2 \\ DX+EY=-CF \end{cases}$ By solving the equation group, we obtain

$$\begin{cases} X=\frac{-CDF\pm CE\sqrt{D^2+E^2-F^2}}{D^2+E^2} \\ Y=\frac{-CEF\mp CD\sqrt{D^2+E^2-F^2}}{D^2+E^2} \end{cases} \tag{8}$$

where $D^2+E^2-F^2 \geq 0$, i.e., $(x'(t))^2+(y'(t))^2 \geq (r'(t))^2$.

For a point inside the 2D region represented by a DBSC, we use a parameter u that varies linearly from the corresponding point on the skeleton to the boundary, i.e., when $u=0$, the point is on skeleton; $u=1$, the point is on the boundary, and $0<u<1$, the point is between the skeleton point and the boundary point. Therefore, equation 6 becomes $(x-x(t))^2+(y-y(t))^2-u^2(r(t))^2=0$, equation 7 becomes $(x-x(t))x'(t)+(y-y(t))y'(t)+u^2r(t)r'(t)=0$. Similarly, through solving the equation group, we obtain the interior point.

Explicit Boundary Evaluations

Here, we derive explicit formulas for computing points on the boundary of a DBSC and their differentials.

Computing points on the boundary of DBSC: The boundary can be explicitly represented by rewriting equation 8 as follows:

$$\begin{aligned} b^{\pm}(\mathrm{t}) = & \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \\ & + \frac{r}{x'^2+y'^2}\left[\pm\sqrt{x'^2+y'^2-r'^2}\begin{pmatrix} -y' \\ x' \end{pmatrix}\right. \\ & \left. -r'\begin{pmatrix} x' \\ y' \end{pmatrix}\right] \end{aligned} \tag{9}$$

Here $b^{\pm}$ represents the upper boundary and the lower boundary.

When a DBSC is open and the radius at its end is not zero, the end part is an arc whose center is the end point of the center curve. The start and end points on the arc can then be computed from the envelope computation above. Therefore, any point on the boundary of the disk B-spline region can be obtained.

Boundary differentials: Applying the derivative to equation 9 gives the tangents on the boundary as follows:

The upper part:

$$\begin{aligned} b'^{+} = & \Bigg(1-\tfrac{r'^2}{x'^2+y'^2}-r\sqrt{1-\tfrac{r'^2}{x'^2+y'^2}}\,\tfrac{y''^3x'-x''^3y'}{(x'^2+y'^2)^2} \\ & -r\tfrac{r''\left(x'^2+y'^2\right)-r'\left(x'x''+y'y''\right)}{(x'^2+y'^2)^2}\Bigg)\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} \\ & +\Bigg(\tfrac{r'\sqrt{x'^2+y'^2-r'^2}}{x'^2+y'^2}+\tfrac{rr'\left(y''^3x'-x''^3y'\right)}{(x'^2+y'^2)^{\frac{5}{2}}} \\ & +\tfrac{2rr'^2\left(x'x''+y'y''\right)-r'r''\left(x'^2+y'^2\right)}{(x'^2+y'^2)^2\sqrt{x'^2+y'^2-r'^2}}\Bigg)\begin{pmatrix} -y'(t) \\ x'(t) \end{pmatrix} \end{aligned} \tag{10}$$

The lower part:

$$\begin{aligned} b'^{-} = & \Bigg(1-\tfrac{r'^2}{x'^2+y'^2}+r\sqrt{1-\tfrac{r'^2}{x'^2+y'^2}}\,\tfrac{y''^3x'-x''^3y'}{(x'^2+y'^2)^2} \\ & -r\tfrac{r''\left(x'^2+y'^2\right)-r'\left(x'x''+yy''\right)}{(x'^2+y'^2)^2}\Bigg)\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} \\ & +\Bigg(\tfrac{-r'\sqrt{x'^2+y'^2-r'^2}}{x'^2+y'^2}+\tfrac{rr'\left(y''^3x'-x''^3y'\right)}{(x'^2+y'^2)^{\frac{5}{2}}} \\ & -\tfrac{2rr'^2\left(x'x''+y'y''\right)-r'r''\left(x'^2+y'^2\right)}{(x'^2+y'^2)^2\sqrt{x'^2+y'^2-r'^2}}\Bigg)\begin{pmatrix} -y'(t) \\ x'(t) \end{pmatrix} \end{aligned} \tag{11}$$

The normal can be obtained by rotating the tangent at 90°. Figure 3 illustrates the tangents and normals on the boundary of a DBSC.

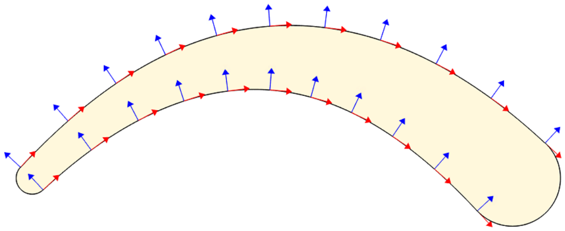


Figure 3: Illustration of tangents(red) and normals(blue) on the boundary of a DBSC.

Explicit Evaluation of Arbitrary Points inside a DBSC

Here, we derive explicit formulas for computing arbitrary points on a DBSC and their differentials.

Computing points inside a DBSC: We denote an arbitrary point inside the 2D region of a DBSC as $P(t,u)$, where t is the parameter of the DBSC, and u is the parameter that represents the point that varies linearly from the corresponding point on the skeleton to the boundary. Then, an arbitrary point inside a DBSC can be represented as

$$\begin{aligned}&\mathrm{P}^{\pm}(\mathrm{t},\mathrm{u}) = \begin{pmatrix} \mathrm{x(t)} \\ \mathrm{y(t)} \end{pmatrix}\\ &+\tfrac{\mathrm{ur}}{\mathrm{x}'^2+\mathrm{y}'^2}\left[\pm\sqrt{\mathrm{x}'^2+\mathrm{y}'^2-\mathrm{u}^2\mathrm{r}'^2}\begin{pmatrix} -\mathrm{y}' \\ \mathrm{x}' \end{pmatrix}\right.\\ &\left.-\,\mathrm{ur}'\begin{pmatrix} \mathrm{x}' \\ \mathrm{y}' \end{pmatrix}\right]\end{aligned} \tag{12}$$

Here, $\mathrm{P}^{\pm}(\mathrm{t},\mathrm{u})$ describes arbitrary points inside the 2D region represented by a DBSC as shown in Figure 4.

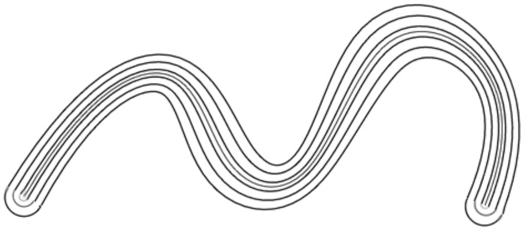


Figure 4: The boundary and variable offsets.

Differentials of Arbitrary Points inside DBSC: As a DBSC represents a 2D region, we can compute the properties of arbitrary points inside the region. Here, we provide the partial differentials according to parameters t and u. The partial differential of parameter t obtained by applying the derivative to equation 12 is as follows:

The upper part:

$$\begin{aligned}P_t'^{+} &= \Bigg(1-\tfrac{r'^2}{x'^2+y'^2}-\mathrm{ur}\sqrt{1-\tfrac{u^2r'^2}{x'^2+y'^2}}\,\tfrac{y''^3x'-x''^3y'}{(x'^2+y'^2)^2}\\ &-ur\tfrac{u^2r''\left(x'^2+y'^2\right)-ur'\left(x'x''+y'y''\right)}{(x'^2+y'^2)^2}\Bigg)\begin{pmatrix} x' \\ y' \end{pmatrix}\\ &+\Bigg(\tfrac{ur'\sqrt{x'^2+y'^2-u^2r'^2}}{x'^2+y'^2}+\tfrac{u^2rr'\left(y''^3x'-x''^3y'\right)}{(x'^2+y'^2)^{\frac{5}{2}}}\\ &+\tfrac{2u^3rr'^2\left(x'x''+y'y''\right)-u^2r'r''\left(x'^2+y'^2\right)}{(x'^2+y'^2)^2\sqrt{x'^2+y'^2-u^2r'^2}}\Bigg)\begin{pmatrix} -y' \\ x' \end{pmatrix}\end{aligned} \tag{13}$$

The lower part:

$$\begin{aligned}P_t'^{-} &= \Bigg(1-\tfrac{u^2r'^2}{x'^2+y'^2}+\mathrm{r}\sqrt{1-\tfrac{u^2r'^2}{x'^2+y'^2}}\,\tfrac{y''^3x'-x''^3y'}{(x'^2+y'^2)^2}\\ &-ur\tfrac{ur''\left(x'^2+y'^2\right)-ur'\left(x'x''+y'y''\right)}{(x'^2+y'^2)^2}\Bigg)\begin{pmatrix} x' \\ y' \end{pmatrix}\\ &+\Bigg(\tfrac{-ur'\sqrt{x'^2+y'^2-u^2r'^2}}{x'^2+y'^2}+\tfrac{u^2rr'\left(y''^3x'-x''^3y'\right)}{(x'^2+y'^2)^{\frac{5}{2}}}\\ &-\tfrac{2u^3rr'^2\left(x'x''+y'y''\right)-u^2r'r''\left(x'^2+y'^2\right)}{(x'^2+y'^2)^2\sqrt{x'^2+y'^2-u^2r'^2}}\Bigg)\begin{pmatrix} -y' \\ x' \end{pmatrix}\end{aligned} \tag{14}$$

The partial differential with respect to the parameter u can be computed by applying the derivative to equation 12 at u:

$$\begin{aligned}P_u^{\pm} &= \tfrac{r}{x'^2+y'^2}\left[\pm\sqrt{x'^2+y'^2-u^2r'^2}\begin{pmatrix} -y' \\ x' \end{pmatrix}\right.\\ &\left.-ur'\begin{pmatrix} x' \\ y' \end{pmatrix}\right]-\tfrac{ur}{x'^2+y'^2}\\ &\left[\pm\tfrac{ur'^2}{\sqrt{x'^2+y'^2-u^2r'^2}}\begin{pmatrix} -y' \\ x' \end{pmatrix}+r'\begin{pmatrix} x' \\ y' \end{pmatrix}\right]\\ &= \tfrac{\pm r\left(x'^2+y'^2\ -2u^2r'^2\right)}{(x'^2+y'^2)\sqrt{x'^2+y'^2-u^2r'^2}}\begin{pmatrix} -y' \\ x' \end{pmatrix}-\tfrac{2urr'}{x'^2+y'^2}\begin{pmatrix} x' \\ y' \end{pmatrix}\end{aligned} \tag{15}$$

According to equations 14 and 15, we can obtain partial differentials of arbitrary points inside a DBSC, as shown in Figure 5.

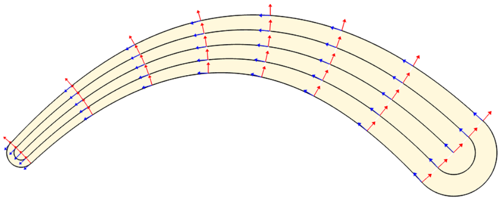


Figure 5: Illustration of Partial differentials of arbitrary points inside DBSC (red:along parameter t, and blue along parameter u).

MODELING METHODS

In geometric modeling, the three fundamental modeling methods are interpolation, approximation, and deformation. Here, we introduce these methods to a DBSC.

Interpolation

We can use the cubic B-spline interpolation method to interpolate the center curve and the B-spline scalar function method to interpolate the radius in a disk B-spline curve.

Given a group of points Q_i, $i = 0, \ldots, n$ on the central curve and their corresponding radii d_i, a disk B-spline curve whose center curve passes these data points Q_i and whose control radius is d_i can be obtained by interpolation. The knot vector is determined through its accumulated chord length

$$\begin{cases} t_0 = 0 \\ t_i = t_{i-1} + |Q_i - Q_{i-1}|, i = 1, 2, \ldots, n \end{cases}$$

and knot vector

$$T = [t_0, t_0, t_0, t_0, t_1, t_2, \ldots, t_{n-1}, t_n, t_n, t_n, t_n] \tag{16}$$

The interpolated DBSC is

$$D(t) = \sum_{i=0}^{n+2} N_{i,3}(t) \langle \mathrm{P}_i, r_i \rangle$$

Then,

$$\begin{cases} \sum\limits_{i=0}^{n+2} N_{i,3}(t_j) \mathrm{P}_i = Q_j \\ \sum\limits_{i=0}^{n+2} N_{i,3}(t_j) r_i = d_j \end{cases}, j = 0, 1, 2, \ldots, n$$

Through adding end constraint conditions, the DBSC can be obtained by solving the linear equation group. The curve can be either closed or open. Figure 6 shows a few examples.

Approximation

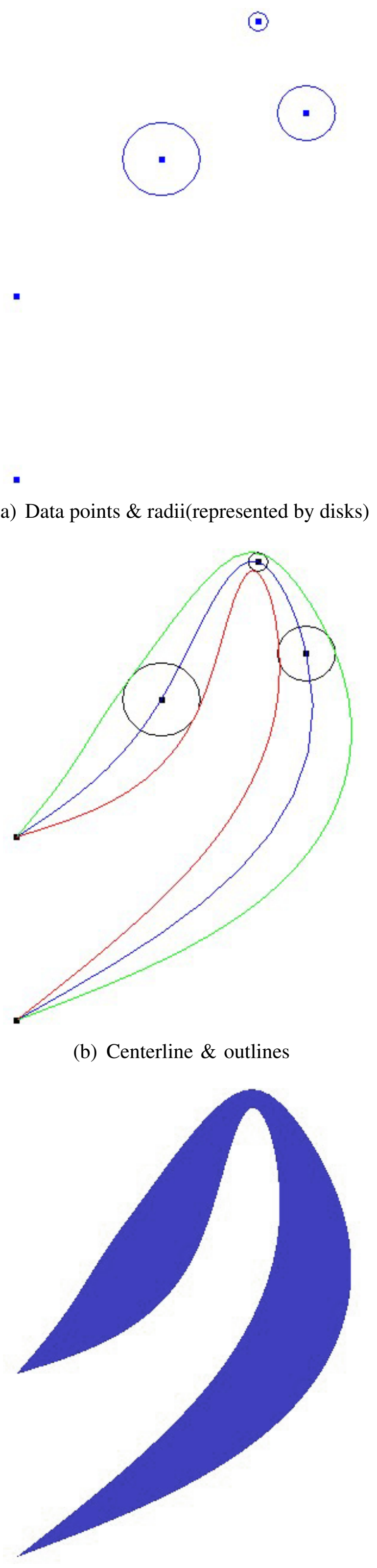
Here, we can use the cubic B-spline curve approximation method and B-spline scalar function approximation method to obtain the disk B-spline curve approximation.

Given a group of points Q_i, $i = 0, \ldots, m$, and their corresponding radii d_i, a DBSC whose center curve approximates these data points Q_i and whose radii are d_i can be obtained by approximation.

The knot vector is determined same as equation 16. The approximated DBSC $D(t) = \sum_{i=0}^{n} N_{i,3}(t) \langle \mathrm{P}_i, r_i \rangle$. Then, the approximation is regarded as the following optimization problem:

$$f = \min_{p_i, r_i} \sum_{j=0}^{m} \left(\left\| \sum_{i=0}^{n} N_{i,3}(t_j) P_i - Q_j \right\|^2 + \left\| \sum_{i=0}^{n} N_{i,3}(t_j) r_i - d_j \right\|^2 \right)$$

By solving the optimization, the DBSC is obtained. Figure 7 is an example illustrating the approximation result.



(a) Data points & radii(represented by disks)

(b) Centerline & outlines

(c) Region

Figure 6: Interpolation

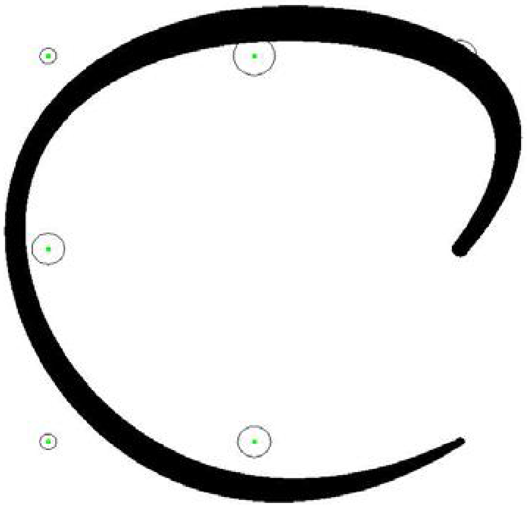
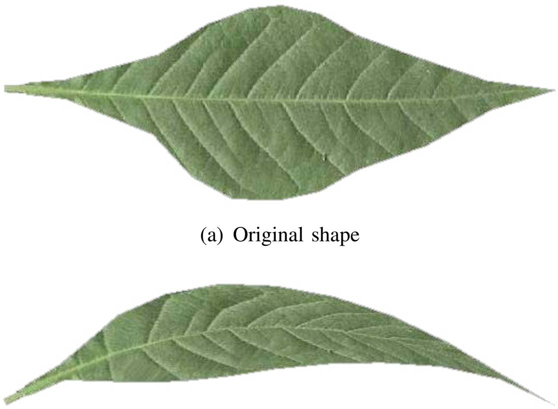


Figure 7: Approximation

Deformation

A new shape can be generated by deformation to a disk B-spline curve by applying deformation to its center curve and scaling its control radius function, as shown in Figure 8.



(a) Original shape

(b) Deformed shape

Figure 8: Deformation

Note that the deformation here is only based on geometry. As a DBSC can describe HE objects, physically anisotropic simulations can be implemented in the future.

HETEROGENEOUS OBJECT MODEL

As a DBSC describes a 2D region, we can define different attributes at each point, i.e., we can represent HE objects on a DBSC.

HE object representation

A 2D region is usually represented by its contour, which only describes homogeneous objects. However, as a DBSC can represent arbitrary points in a 2D region, it is suitable to describe natural objects with various heterogeneous attributes, such as heterogeneous materials, various physics attributes, textures, and fields. Therefore, HE objects with diverse attributes can be modeled. Here, we provide a representation method for HE objects as follows with definition as

$$\begin{aligned} &p = (p_g, p_m) \\ &p_g = (x, y) \in \Omega \subset \mathrm{R}^2 \\ &p_m = (a_1, a_2, \dots a_k) \in \mathrm{R}^k \end{aligned} \tag{17}$$

where p is a HE object, p_g is an arbitrary point on a DBSC, and p_m are the attributes on the point. Obviously, p_g is a DBSC. According to 12, $p_g = p(t, u)$. To keep the geometry and attributes consistent, we still use the B-spline form for the attributes p_m, i.e., the attributes on the DBSC are determined by these attributes on control disks blended by the B-spline basis. We regard the DBSC as a single parameter blending of control disks. For each disk, we use a polar coordinate system, i.e., (ρ, θ). For each point $p_g = p(t, u)$,

$$\begin{aligned} &\rho = \|P(t,u) - c(t)\| \\ &\theta = \arccos\left(\frac{P(t,u)-c(t)}{\|P(t,u)-c(t)\|}, \frac{c'(t)}{\|c'(t)\|}\right) \end{aligned}$$

here, $c(t)$ is the corresponding point on the skeleton. Therefore, we have the attributes on the point as

$$p_m(t,u) = (a_1(t,u), a_2(t,u), \dots a_k(t,u)) = \sum_{i=0}^{n} N_{i,p}(t) A_i(\rho, \theta) \tag{18}$$

where $A_i(\rho, \theta), i = 0, 1, \dots, n$ are the attributes on control disks.

Interactive HE object design

Designing a HE object is a challenging problem. Here, we provide interactive HE object modeling methods.

Efficient tessellation: Based on the computation above, we can obtain a group of points inside a DBSC at any resolution. By linking these corresponding points, the 2D region is tessellated, as shown in Figure 9. With tessellation, rendering and attribute mapping can be implemented.

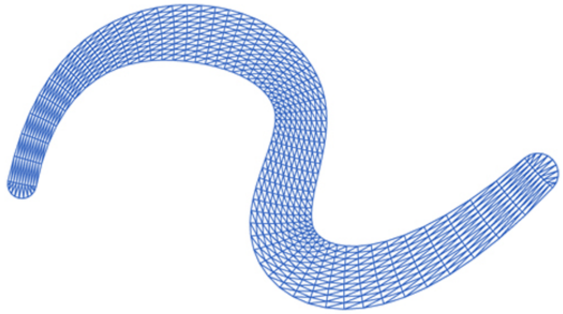
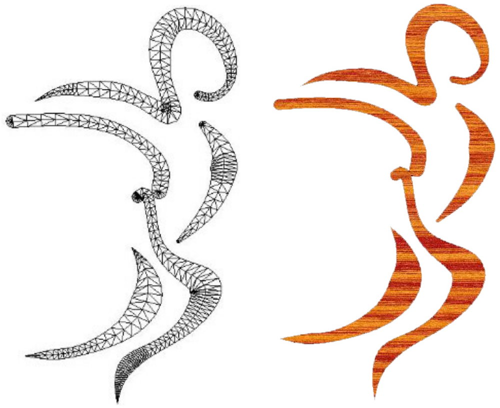


Figure 9: Tessellation of the region represented by a DBSC

Attribute mapping: After tessellation, an attribute mapping technique can be directly applied. For the purpose of visualization, we use different colors to indicate different attributes. Various patterns and styles can be generated by attribute mapping, as shown in Figures 10 and 11 (here we display various attributes by colors).



(a) Triangulation

(b) Attribute mapping

Figure 10: Triangulating and attribute mapping DBSC

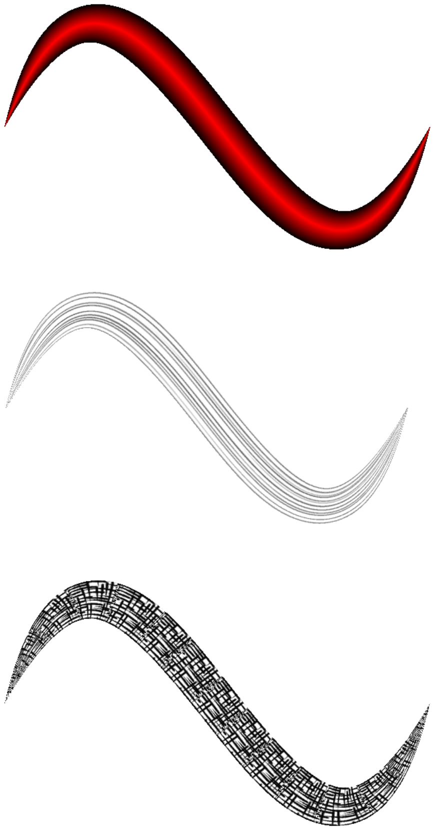


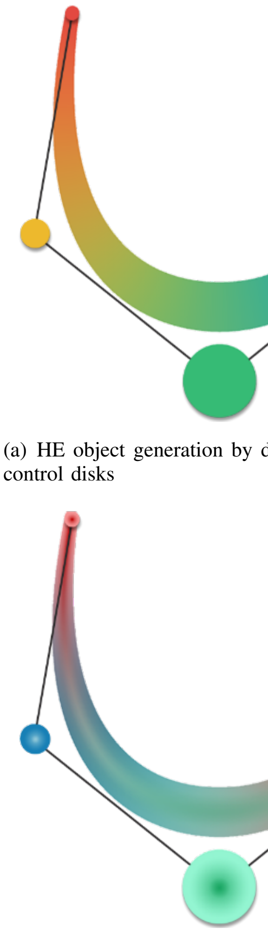
Figure 11: Various styles of brushstrokes

Interactive HE object design: To apply attributes on each control disk of a DBSC, users can design various attributes (display by colors), i.e., HE objects based on a DBSC. According to equation 18, the attributes on a DBSC can be controlled through the attributes on control disks, as shown in Figure 12.

One advantage of these HE objects is that users can design different HE objects by adjusting the attributes on the control disks interactively. Another advantage is that the attributes on a DBSC remain consistent when users change a shape represented by the DBSC by adjusting its control points and radii, as shown in Figure 13 (display by colors).

MODELING RESULTS AND CONCLUSIONS

The DBSC presented in this paper is of a general nature and is a skeleton-based parametric representation of a 2D region; i.e., the DBSC describes the center curve (skeleton) of the region directly. It can be used not only in bounding error calculations but also as a geometric model for representing and modeling 2D freeform regions. The examples in Figure 14 and 15 demonstrate the powerful modeling capabilities of a DBSC.



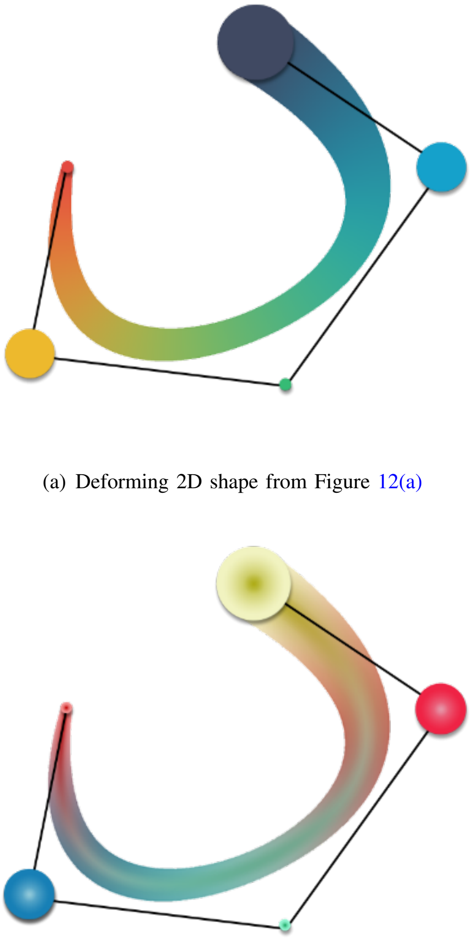
(a) HE object generation by defining attributes of control disks

(b) HE object generation by changing attributes of control disks

Figure 12: Interactive HE object design by defining control disks' attributes (display by colors)

Compared with other approaches, such as mesh or contour approaches, a DBSC has a number of advantages, including the following:

- Solid mathematical fundamentals
- Precise evaluation
- Flexibility in manipulations and deformations
- Representation of both a 2D region and its skeleton
- More compact dataset than discrete or linear representations when defining a freeform 2D region.
- Representation of an HE object on a DBSC.



(a) Deforming 2D shape from Figure 12(a)

(b) Deforming 2D shape from Figure 12(b)

Figure 13: The attributes remain consistent when the 2D shape changes with control points and radii

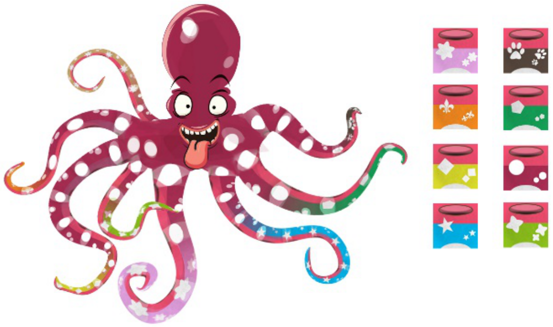
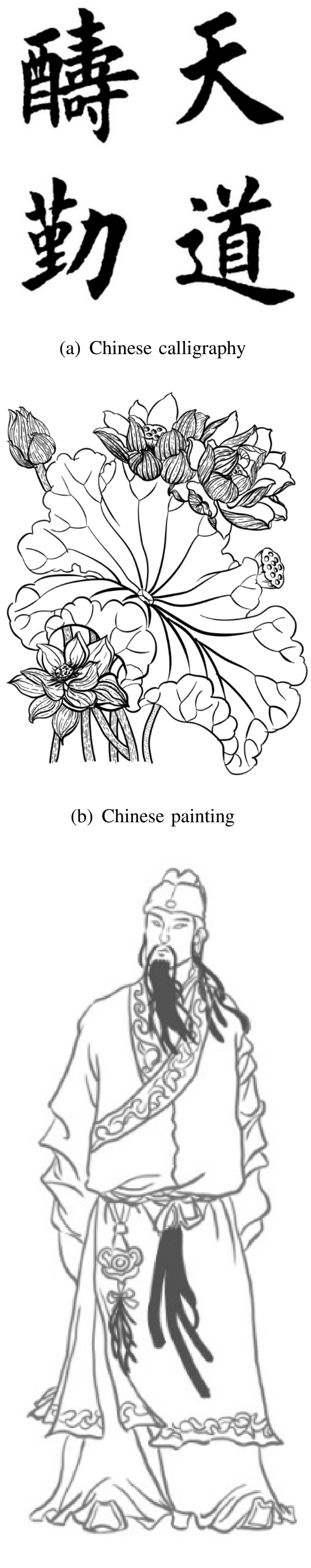


Figure 14: HE object model



(a) Chinese calligraphy

(b) Chinese painting

(c) Portrait

Figure 15: Models of disk B-spline curves

In this paper, we have discussed the fundamental properties and algorithms of DBSC and derived explicit evaluations and derivative formulas on arbitrary points of a DBSC. We have also presented fundamental modeling methods for DBSCs, such as interpolation, approximation and deformation. Moreover, we have provided a novel representation of HE objects and implemented an interactive heterogeneous object design method. These results will lay a solid foundation for further research on DBSC .

By defining specific attributes on the region, such as a scalar field and vector field, DBSC representation will have wide and deep applications in various areas such as nonphotorealistic rendering, 2D modeling, computer-assisted cel animation, and medical imaging. Further research is needed in the future on the heterogeneous object simulation, additional modeling methods, HE object design and simulation methods based on disk B-splines, and bioprinting.

Acknowledgement

This work is partially supported by National Natural Science Foundation of China (No.61972041, No.62072045), the National Key Cooperation between the BRICS of China (No.2017YFE0100500), National Key R&D Program of China (No. 2017YFB1002604, No.2017YFB1402105).

■ REFERENCES

1. T. W. Sederberg, R. T. Farouki. ”Approximation by interval Bézier curves,” IEEE Computer Graphics & Applications, 12(5):P.87-95, 1992.
2. Q. Lin, J. G. Rokne. ”Disk Bézier curves.” Computer Aided Geometric Design 15(7): 721-737, 1998.
3. Z. Wu, H. S. Seah, F. Tian, X. Xiao, X. Xie, ”Simulating Artistic Brushstrokes Using Disk B-Spline Curves,” MAAP 2004, Singapore
4. X. Xiao, H. S. Seah, Z. Wu, F. Tian, X. Xie, ”Interactive Free-hand Drawing and In-between Generation with Disk B-spline Curves,” MAAP 2004, Singapore
5. T. Whitted, ”Anti-aliased Line Drawing using Brush Extrusion,” SIGGRAPH ’83, 151 – 156, 1983.
6. T. Pudet, ”Real Time Fitting of Hand-Sketched Pressure Brushstrokes,” Computer & Graphics Forum. 13(3):205–220, 1994.
7. H. S. Seah, Z. Wu, F. Tian, X. Xiao, B. Xie, ”Artistic brushstroke representation and animation with disk b-

spline curve,” Proceedings of the ACM SIGCHI International Conference on Advances in computer entertainment technology, p.88-93, June 15-17, 2005, Valencia, Spain

8. S. L. Su, Y. Q. Xu, H. Y. Shum, F. Chen, ”Simulating Artistic Brushstrokes Using Interval Splines,” The 5th International Conference on Computer Graphics and Imaging, pp. 85-90 Kauai, Hawaii, August 2002.
9. Q. Fu, Z. Wu, X. Ying, M. Wang, X. Zheng, M. Zhou, ”Generating Chinese Calligraphy on Freeform Shapes.” In Transactions on Computational Science XXVIII (pp. 69-87, 2016). Springer, Berlin, Heidelberg.
10. Q. Chen, Z. Wu, F. Tian, H. S. Seah, J. Qiu, K. Melikhov, ”Vectorization of Raster Line-drawings in Cartoons,” In Proceedings of International Conference on Computer Animation and Social Agents, CASA, pages 151-158, 2005.
11. Q. Chen, F. Tian, H. S. Seah, Z. Wu, J. Qiu, and M. Konstantin, ”DBSC-based animation enhanced with feature and motion,” Comput. Animat. Virtual Worlds, vol. 17, no. 3-4, 189—198, 2006.
12. D. Liu, Q. Chen, J. Yu, H. Gu, D. Tao, H. S. Seah, ”Stroke correspondence construction using manifold learning,” Comput. Graph. Forum, pp. 2194–2207, 2011.
13. X. Ning, H. Laga, S. Saito, M. Nakajima, ”Extended papers from NPAR 2010: Contour-driven Sumi-e rendering of real photos,” Computers and Graphics, v.35 n.1, p.122-134, 2011
14. T. Zhang, X. Wang, Q. Jiang, Z. Wu, M. Zhou, H. S. Seah, ”G2-continuity extension algorithm for disk B-spline curve,” In 2013 International Conference on Computer-Aided Design and Computer Graphics (pp. 413-414, 2013). IEEE.
15. C. Min, & Guojin, W, ”Shape blending of artistic brushstroke represented by disk B-spline curves,” Progress in Natural Science, 17(12), 1501-1507, 2007.
16. H. C. Liu, J. S. Deng, ”Fitting Scattered Data with Disk/Ball Bézier and B-Spline Curves/Surfaces,” Journal of University of Science and Technology of China, Vol.38, No.2, 113–120, 2008.2, 2008.
17. X. Ao, Q. Fu, Z. Wu, X. Wang, M. Zhou, Q. Chen, H. S. Seah, ”An intersection algorithm for disk B-spline curves, ” Computers & Graphics, 70, 99-107, 2018.
18. Y. Zhang, Z. Wu, & X. Wang, ”Dynamic disk B-spline curves,” Computer Animation and Virtual Worlds, e1955, 2020.
19. Kou, X. Y., & Tan, S. T. (2007). Heterogeneous object modeling: A review. Computer-Aided Design, 39(4), 284-301.
20. Tereshin, A., Pasko, A., Fryazinov, O.,& Adzhiev, V. (2021). Hybrid Function Representation for Heterogeneous Objects. Graphical Models, 101098.
21. Li, B., Fu, J., Feng, J., Shang, C., & Lin, Z. (2020). Review of heterogeneous material objects modeling in additive manufacturing. Visual Computing for Industry, Biomedicine, and Art, 3(1), 1-18.
22. L. Piegl and W. Tiller, ”The NURBS Book, Springer-Verlag,” New York, 1995.
23. H. Pottmann and M. Peternell, ”Envelopes-computational theory and applications,” In B. Falcidieno (eds, Spring Conference on Computer Graphics, Comenius University, Bratislava, pp 3-23, 2000.

Zhongke Wu is Full Professor in School of artificial intelligence, Beijing Normal University (BNU), China. He was the member of the first Steering Committee for Professional Teaching of Animation, Digital Media in Colleges and Universities of Ministry of Education, China. Prior to joining BNU, he worked in Nanyang Technological University (NTU), Singapore, Institute National de Recherche en Informatique et en Automatique (INRIA) in France, Institute of High-Performance Computing (IHPC), Singapore, and Institute of Software, Chinese Academy of Sciences, in China from 1995 to 2006. Prof. WU has been working in the field of computer graphics since 1988. He led and took part in various research and development projects in computer graphics and related areas. Prof. WU's current research interests include computer graphics, animation virtual reality, geometric modeling, shape analysis and medical imaging. Contact him at zwu@bnu.edu.cn.

Xingce Wang is a Professor in School of artificial intelligence, Beijing Normal University, PR China. She completed major in the 3D modeling and 3D visualization. Her current research interests include computer graphics, medical imaging, artificial intelligence and machine learning. Contact her at wangxingce@bnu.edu.cn.

Shaolong Liu obtained his bachelors and masters degrees in Digital Media Arts from Beijing Normal University, China. He is currently a Ph.D. candidate of the School of Artificial Intelligence, Beijing Normal University, China. His current research interests include computer graphics, computer-assisted animation and VR. Contact him at 201531210002@mail.bnu.edu.cn.

Dr. Chen Quan received his BSc in Computer Science and Technology from Fudan

University, China, and PhD in computer engineering from Nanyang Technological University (NTU), Singapore. His research interests include computer-assisted animation, computer graphics and game related technologies. He is one of the inventors of Computer-Assisted Cel Animation (CACANi) technology and holds several patents. He is a Principal Research Fellow in NTU and is currently Associate Director of Singtel-NTU Corporate Lab. Contact him at QChen@ntu.edu.sg.

Hock Soon Seah is a professor at the School of Computer Science and Engineering at Nanyang Technological University (NTU), Singapore, and the Director of the Centre for Augmented and Virtual Reality. His research interests are in geometric data modeling, image sequence analysis, nonphotorealistic rendering, computer animation and game, and virtual and augmented reality. He is the principal investigator of the Computer Assisted Cel Animation (CACANi) software system and Fellow of the Singapore Academy of Engineering. Before joining NTU, he was a programmer analyst at IBM Singapore Pte Ltd. Contact him at ashsseah@ntu.edu.sg.

Feng Tian is a Professor in the Faculty of Science & Technology, Bournemouth University, UK. Before joining Bournemouth University, he was a faculty in the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include computer graphics, computer games, virtual reality, augmented reality, machine learning and image processing. Contact him at ftian@bournemouth.ac.uk.