Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Controllable Blending of Line and Polygon Skeleton-based Convolution Surfaces with Finite Support Kernels

Xiaoqiang Zhu[a], Qi Chen[a], Sihu Liu[a], Chenjie Fan[a], Chenze Song[a], Junjie Zhang[a,*], Dan Zeng[a], Xiaogang Jin[b]

[a] *School of Communication and Information Engineering, Shanghai University, Shanghai, 200444, China*
[b] *State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China*

ARTICLE INFO

Article history:
Received May 28, 2022

Keywords: Convolution Surfaces, Implicit Surface Blending, Semi-analytical Solutions, Support Radius

ABSTRACT

We present a novel approach to control the blending of line and polygon skeleton-based convolution surfaces using locally varying Ratio of Support radius and Thickness (RST). With our method, solutions for local convolution surface approximation with prescribed surface thickness and support radii can be derived analytically. In addition, iso-surface shrinkage can be avoided by offsetting the endpoints of line skeletons and the edges of polygon skeletons. Our RST-based blending for convolution surfaces is local and can generate desired blending effects while approximating shapes with a specified thickness. Moreover, our method is intuitive and users can control the blending by adjusting the skeletal radius or the support radius of the finite support kernel independently. As our blending utilizes convolution integration only without requiring any extra composition operators, it allows for successive convolution blending operations to create complex shapes.

1. Introduction

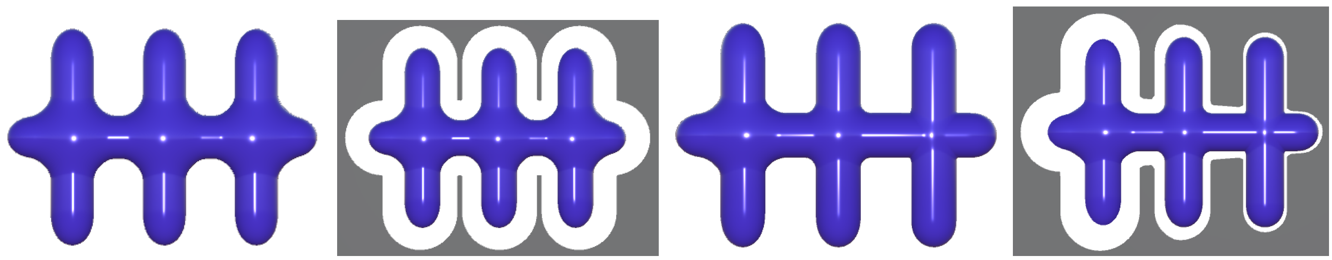
Due to their powerful modeling ability to represent deformable objects with variable topologies, implicit surfaces are one of the most important 3D modelling techniques. To efficiently design complex shapes in an intuitive way, skeleton-based implicit surfaces [1] are preferred for editing-intensive applications. Convolution surface is a typical one, as it possesses some important properties such as superposition and smoothness (Fig. 1(a)(b)).

However, the generated convolution surfaces with the same thickness [2] usually lead to similar blending effects at branches (e.g. Fig. 1(a)) due to their uniform Ratios of Support radius and Thickness (RSTs) (Fig. 1(b)), and the underlying non-intuitive blending control hinders its applications when both a predefined thickness and a desired blending effect are required. As a useful implicit modelling operator, blending has been investigated in many different applications for creating complex models from simple ones. Although some blending techniques [3] for general implicit surfaces could generate arbitrary blending, the important superposition property of convolution surfaces will no longer be preserved. To this end, we propose a controllable blending method for line (e.g. Fig. 1(c)) and polygon skeleton-based convolution surfaces with finite support kernels, which allows for various blends in the resulting convolution surfaces with prescribed thickness by tuning their RSTs (Fig. 1(d)). Our method utilizes a summation operator without any extra blending operators, which is compatible with the superposition property. Therefore, our method allows for successive convolution surface composition and it can be easily extended to n-ary blending.

In summary, our paper makes the following contributions:

- A novel blending control method for line and polygon skeleton-based convolution surfaces using locally varying RSTs, and it can approximate shapes with a specified

*Corresponding author
e-mail: `junjie_zhang@shu.edu.cn` (Junjie Zhang)



(a) Same blends at different intersections (b) Uniform RST distributions along all skeletons (c) Different blends at different intersections (d) Varying RST distribution along the skeleton from left to right

Fig. 1. Our method is able to generate controllable blending at different intersections, which facilitates users to create convolution surfaces with more control. The blue surfaces in (a)-(d) are the convolution surfaces with the same thickness, and the white regions in (b) and (d) represent the support radii for the underlying skeletons. Prior convolution surfaces will create the same blends with the same thickness (a) because they adopt the same support radius which indirectly depends on RST (b). In contrast, by tuning their RSTs, our method is able to achieve different blends (c) when the convolution surface with the uniform thickness generated by the horizontal line segment skeleton blends with three convolution surfaces generated by three vertical line segments with the same thickness (d).

thickness with local control;

- Analytical solutions for line and polygon skeleton-based local convolution surface approximation with prescribed surface radii and support radii are derived. In addition, numerical solutions for skeletal offsets are proposed for approximating the endpoints of line skeletons and the edges of polygon skeletons.

The remainder of the paper is organized as follows. After introducing the related work in Section 2, convolution surfaces with finite support are described in Section 3. Then we give a detailed description of convolution surface blending by controlling the support radii in Section 4, followed by implementation details and some modelling results with our approach in Section 5, and our paper ends with the conclusion section.

2. Related Work

Although quite a lot of works are related to our approach, the most related ones include convolution surfaces and the blending of implicit surface, which will be detailed in this section.

Convolution surfaces are defined by convolving geometric skeletons with a kernel function. It was firstly introduced in [4] to create smooth surfaces using Gaussian kernel. In contrast to globally supported kernels such as Gaussian function and Cauchy function, finite supported kernels [5, 6] are developed due to their locality property for efficient computation. Although the intensive computation of convolution integration hinders its applications, it is still feasible to employ convolution surfaces in interactive modeling systems [7, 2, 8] due to the closed-form solutions [9]. Theoretically, any geometric primitives can serve as skeletons for convolution surfaces, where line segments [10, 11] are the mostly preferred ones for producing branching structures for their simplicity and efficiency. When shapes with a planar surface are to be modeled, polygon skeletons will be a better option if their analytical solutions can be derived [7, 12, 13]. To further enrich the diversity of created convolution surfaces, approaches of convolving line skeletons with smoothly varying thickness are proposed [14, 15, 16], and extended convolution surfaces [17] with anisotropy for 1D skeletons further increase modeling freedom. Inspired by convolution surfaces, integral surfaces have been proposed to design scale-invariant skeleton-based implicit surfaces [18], which approximate prescribed radii well. Furthermore, convolution surface has been recently employed for producing heterogeneous objects [19] and smoothing distance fields [20] with high-quality continuity. In [21], general barycentric coordinates are introduced to interpolate the convolution surface thickness, and it successfully extends the modeling ability of convolution surfaces. However, to the best of our knowledge, no approach is available to explicitly control the blending of convolution surface without losing its intrinsic superposition property.

Implicit surface blending is an important technique for modeling complex shapes with implicit surfaces, which are commonly used in fluid simulation [22], sketch-based modeling [23, 7, 11, 24], cloth [25], and character skin animation [26]. In [27], an approach combining various implicit surfaces with soft blending capacities in a CSG tree is proposed, and it allows for integration of plane surfaces, skeletons, and many other types of implicit surfaces. In order to control the blending, bounded regions between components are defined to restrict a locally blended shape [28, 3] avoiding unwanted topology changes. In [29], a new family of binary composition operators are introduced based on both the field values and the angles between the gradients of the input fields, which solve the problems of locality, bulge, absorption, and topology. To model inner shapes with intersections or differences, new constraints on field functions are introduced to create continuous fields for both inner and outer boundaries [30]. However, most of these approaches are designed for general implicit surfaces instead of convolution surfaces. As a result, the important superposition property of convolution surfaces usually cannot be preserved when extra blending operators are introduced. Specifically, for convolution surfaces [10] and SCALIS surfaces [18], the unwanted blending can be avoided by neglecting topologically too far away skeletons or scale the convolved fields. However, it is nontrivial to satisfy the requirement of various blends within a large skeleton, especially for polygon skeletons. Based on the summation operator for blending SCALIS surfaces, Zanni *et. al.* introduced an approach for blending n-ary skeleton-based implicit surfaces [31], which can control the topology using both field values and norms of gradients. Different from them, we propose a novel approach for control the blending of line and polygon skeleton-based convolution surfaces by simply controlling support radii, and it is capable of adjusting blending regions without losing its advantageous superposition property.

3. Convolution Surfaces with Finite Support

3.1. Convolution Surfaces

A convolution surface [4] is a special implicit surface whose potential field is generated by convolving a kernel function with geometric skeletons. Theoretically, any geometric primitive can serve as a skeleton V:

$$g(\mathbf{p}) = \begin{cases} 1, & \mathbf{p} \in V \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Given a kernel function $f : R^3 \rightarrow R$, the convolution field at p is the integral of the functions g and f:

$$h(\mathbf{p}) = \int_V g(\mathbf{q}) f(\mathbf{p}\text{-}\mathbf{q}) dV = (f \otimes g)(\mathbf{p}). \tag{2}$$

When series of skeletons are taken into consideration, the convolution surface S can be defined as:

$$S = \left\{ \mathbf{p} | \sum_{i=1}^{n} \lambda_i h_i(\mathbf{p}) - T = 0 \right\}, \tag{3}$$

where h_i and λ_i are the convolution field and weight of the i^{th} skeleton, respectively, and T is the predefined threshold of the iso-surface.

3.2. Analytical Solutions for Convolution Surfaces

As we all know, field generation is the most computation-intensive part in convolution surface modeling, which should be implemented as efficiently as possible. However, convolution field generation relies on both the kernel function and the embedded skeleton, and both of them should be taken into consideration.

Many low-pass filtering kernels (such as Gaussian, Cauchy, Inverse, Squared, and Polynomial [9] kernels) and geometric skeletons (such as point, line, plane, arc, and triangle) can be utilized to define a convolution scalar field, but elaborate strategies are needed if an analytical solution is desired. So far, closed-form convolution integrals exist only for limited choices of kernels and skeletons, as derivation of arbitrary convolution integrals is not always possible. Fortunately, more and more analytical solutions [9, 15, 13] for pairs of kernels and skeletons have been derived, and they have been adopted to create a variety of models in video games, movies, and scientific applications.

3.3. Finite Support Kernel

Among all the kernels, local support kernels (e.g. polynomials) are preferred compared to global support ones (e.g. Gaussian and Cauchy kernels), as a local support kernel cuts off skeleton segments that are far away from the current position in question.

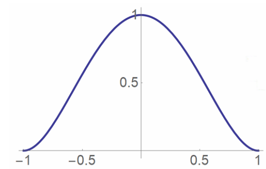


Fig. 2. Quartic kernel

As far as we know, finite support polynomial kernels could be paired with almost all common geometric primitives (point, line, plane, arc, and triangles) to achieve analytical solutions [9]. Moreover, the quartic polynomial has been further investigated for general polygon skeletons [7], local and boundary approximation [11, 2], and thickness interpolation using barycentric coordinates [21]. Another advantage of quartic polynomial kernel over others is its intuitive support range, which is suitable for tuning blending regions. Therefore, the frequently used finitely supported quartic polynomial kernel [32, 33] in Eq. 4

$$f(r) = \begin{cases} (1 - \frac{r^2}{R^2})^2, & r \leq R \\ 0, & r > R \end{cases} \tag{4}$$

is adopted in our approach (Fig. 2), where effective radius R is employed for clipping skeletons that are far away from the position in question.

4. Solutions for Finite Supported Convolution Surface Approximation with Varying Blends

We choose the quartic polynomial kernel for its efficiency. We discuss here the derivation of the blending for the most frequently adopted geometric primitives, including lines and polygons.

4.1. Line Skeleton

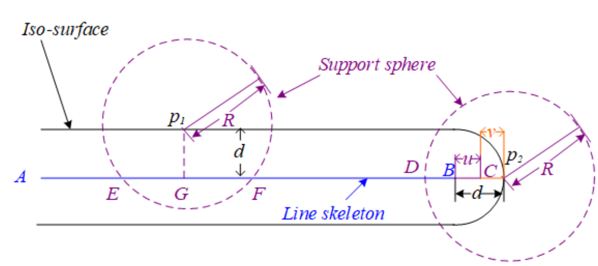


Fig. 3. Convolution approximation for a line skeleton

4.1.1. Thickness Approximation with Varying RSTs

Background. For a line skeleton $skel_i := AB$, as shown in Fig. 3, if the distance from the iso-surface passing point $\mathbf{p}_1$ to AB is d_i, the contribution weight λ_i of $skel_i$ can be calculated using a local convolution surface approximation [11]:

$$\begin{aligned} F_i(\mathbf{p}_1) &= \lambda_i \int_{V=line_{EF}} g(\mathbf{q}) f(r) dV \\ &= 2\lambda_i \int_0^{\sqrt{R_i^2 - d_i^2}} \left(1 - \frac{d_i^2 + x^2}{R_i^2}\right)^2 dx = T, \end{aligned} \tag{5}$$

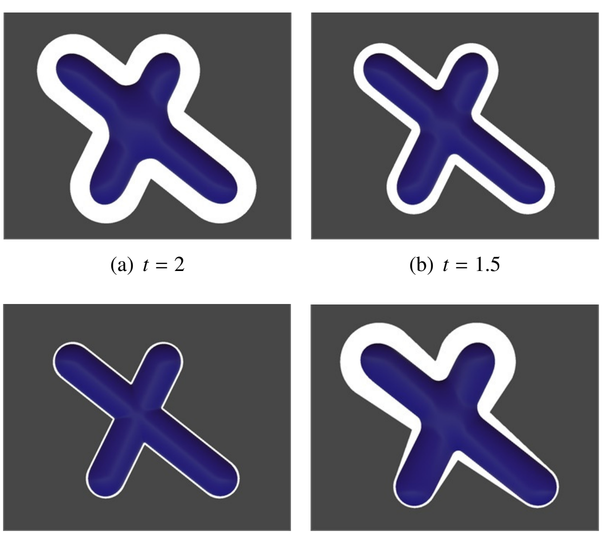
where EF is the skeletal segment clipped by the support sphere. Although our local approximation is derived for a line with infinite length, the method also applies if the skeleton segment has two intersections with the support sphere, which is easy to meet for a finite support kernel.

Analytical Solutions for Contribution Weights. As we know, the region of influence associated with each skeleton directly depends on the support radius. Therefore, to conveniently adjust the blending region, we introduce the RST $t = \frac{R_i}{d_i}$ to control the resulting iso-surface. From Eq. 5, the convolution field weight λ_i can be resolved as:

$$\lambda_i = \frac{15T}{16R_i\left(1-\frac{1}{t^2}\right)^{\frac{5}{2}}} = \frac{15T}{16d_i\, t\left(1-\frac{1}{t^2}\right)^{\frac{5}{2}}}. \tag{6}$$

Our method offers two different choices by either specifying the predefined support radius or the convolution surface thickness (see Eq. 6), which can be chosen according to users' preferences to fine-tune the convolution surface blending.

As illustrated in Fig. 4, all convolution surfaces in blue based on two line skeletons are created with the same thickness. While the support radii in white vary in these sub-figures, which produces various blending effects.



(a) $t = 2$ (b) $t = 1.5$ (c) $t = 1.1$ (d) $2 \geq t \geq 1.1$

Fig. 4. Convolution surfaces with the same thickness but different support radii. When the RST decreases from (a) to (c), the blending regions are constrained within smaller and smaller ranges. Varying RSTs can also be designed to achieve different blending for convolution surfaces with uniform thickness (d)

4.1.2. *End Offset Approximation with Varying RSTs*

Background. Due to the shrinkage of the convolution surfaces at the end of a line segment skeleton AB in Fig. 3, the extended skeleton segment BC enables the iso-surface to pass the desired end position $\mathbf{p}_2$. As described in [2], if we extend the length of the segment as:

$$u = |\mathbf{BC}| = d_i \mp v, \tag{7}$$

where v is the distance from $\mathbf{p}_2$ to $\mathbf{C}$, $v = \left|\mathbf{p}_2\mathbf{C}\right|$, the convolution field at $\mathbf{p}_2$ can be calculated as:

$$\begin{aligned} F_1(\mathbf{p}_2) &= \int_{V=line_{CD}} g(\mathbf{q})\, f(r)\, dV = \int_{d_i-u}^{R_i}\left(1-\frac{x^2}{R_i^2}\right)^2 dx \\ &= \int_{v}^{R_i}\left(1-\frac{x^2}{R_i^2}\right)^2 dx = -v - \frac{v^5}{5R_i^4} + \frac{2v^3}{3R_i^2} + \frac{8R_i}{15}, \end{aligned} \tag{8}$$

and

$$\begin{aligned} F_2(\mathbf{p}_2) &= \int_{V=line_{CD}} g(\mathbf{q})\, f(r)\, dV \\ &= \int_{0}^{R_i}\left(1-\frac{x^2}{R_i^2}\right)^2 dx + \int_{0}^{v}\left(1-\frac{x^2}{R_i^2}\right)^2 dx \\ &= v + \frac{v^5}{5R_i^4} - \frac{2v^3}{3R_i^2} + \frac{8R_i}{15}, \end{aligned} \tag{9}$$

for the "+" case and the "-" case, respectively (see Eq. 7).

Numerical Solutions for Offset Approximation. From Eq. 6-Eq. 9, v can be solved from

$$\mp\frac{v^5}{5R_i^4} \pm \frac{2v^3}{3R_i^2} \mp v + \frac{8R_i}{15} = \frac{16\left(1-\frac{1}{t^2}\right)^{\frac{5}{2}} R_i}{15}. \tag{10}$$

Once t is determined for approximating the thickness of a line segment, the extended segment can be further deduced through

$$\begin{cases} \mp\frac{x^5}{5} \pm \frac{2x^3}{3} \mp x + \frac{8-16\left(1-\frac{1}{t^2}\right)^{\frac{5}{2}}}{15} = 0, \\ x = \frac{v}{R_i}, t > 1, \end{cases} \tag{11}$$

where $x = \frac{v}{R_i}$.

Actually, Eq. 11 can be divided into two sub-equations,

$$\pm f(x) + g(t) = 0, \tag{12}$$

where

$$f(x) = -\frac{x^5}{5} + \frac{2x^3}{3} - x, \tag{13}$$

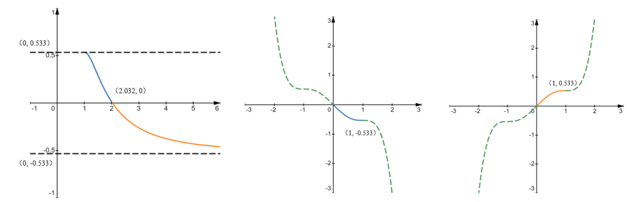
and

$$g(t) = \frac{8-16\left(1-\frac{1}{t^2}\right)^{\frac{5}{2}}}{15}. \tag{14}$$

From Eq. 7, $v = 0$ is the critical point, and t can be solved numerically by $t = 2.032$. Therefore, the intervals $t \in (1, 2.032]$ and $t \in (2.032, +\infty)$ correspond to the "+" case and the "-" case, respectively (see Eq. 12), and the curves of $g(t)$ and $f(x)$ are shown in Fig. 5.

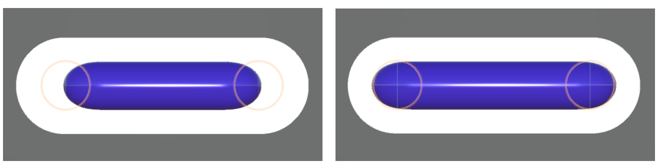
As t falls within $t \in (1, +\infty)$, it is easy to determine the range of $g(t) \in (-0.533, 0.533)$ (see Fig. 5(a)). Although no closed-form solution could be derived for $f(x)$ given a specified t, it can be seen in Fig. 5(b) and (c) that $f(x)$ has the monotonicity property, and therefore the Newton's method can be employed to find the unique numerical solution quickly.

As illustrated in Fig. 6, an obvious shrinkage occurs in (a), which can be avoided by offsetting the line segment in (b).



(a) $g(t)$ (b) $f(x)$ (c) $-f(x)$

Fig. 5. Curve of Eq. 11



(a) End shrinkage (b) Offset-based end approximatiion

Fig. 6. The approximation of the endpoints of line skeletons by offsetting. The potential field generated by the embedded line skeleton (orange) usually results in iso-surface shrinkage at ends (a), which can be compensated (b) by offsetting the skeleton.

4.2. Polygon Skeleton

4.2.1. Thickness Approximation with Varying RSTs

Background. Although convolution surface with planar surfaces can be approximated by convolving lots of line skeletons, the gaps between them has to be elaborately designed and too much integral computation will be involved. A preferred option is to adopt polygon skeletons instead of line segments.

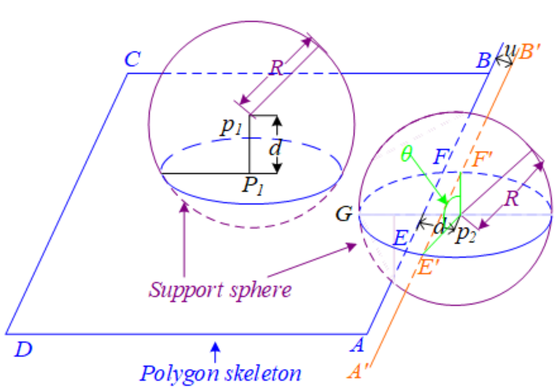


Fig. 7. Convolution approximation for polygon skeleton

In [2], an infinite polygon skeleton is taken into consideration to derive an analytical solution for a local approximation of the surface thickness. As illustrated in Fig. 7, the solutions are valid only if the clipping circle $\mathbf{P}_1$ of the support sphere centered at $\mathbf{p}_1$ in question is inside of the polygon skeleton $ABCD$. If we want to create a convolution surface with a distance d_i to the skeleton, the convolution field weight of the polygon skeleton can be calculated as:

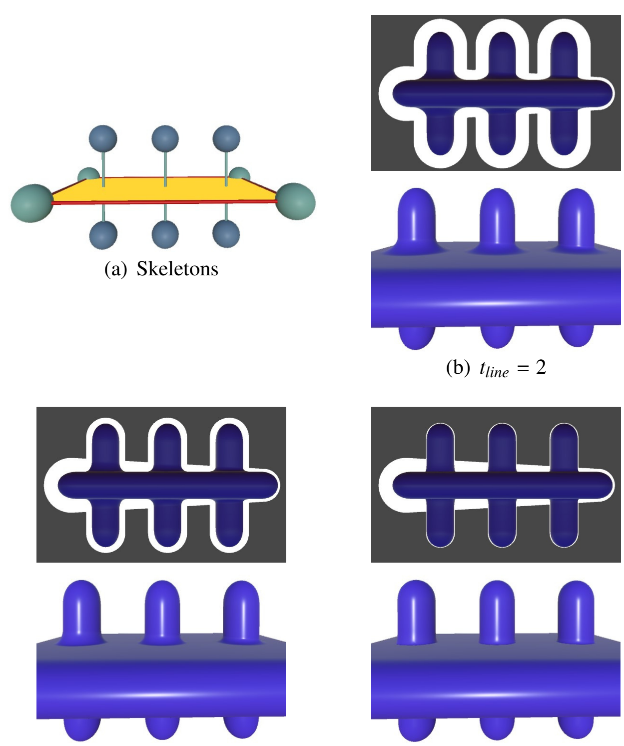
$$\begin{aligned} F(\mathbf{p}_1) &= \lambda_i \int_{V=circle_{P_1}} g(\mathbf{q})\, f(r)\, dV \\ &= \lambda_i \int_0^{2\pi} \int_0^{\sqrt{R_i^2-d_i^2}} \left(1 - \frac{d_i^2 + r^2}{R_i^2}\right) r\, dr\, d\theta = T. \end{aligned} \tag{15}$$

Analytical Solutions for Contribution Weights. Similar to the line skeleton case, the RST $t = \frac{R_i}{d_i}$ can also be adopted to adjust the blending region for a polygon skeleton-based convolution surface. From Eq. 15, the field contribution weight of the polygon can be computed by:

$$\lambda_i = \frac{6R_i^4 T}{2\pi\left(R_i^2 - d_i^2\right)^3} = \frac{3T}{\pi R_i^2\left(1-\frac{1}{t^2}\right)^3} = \frac{3T}{\pi d_i^2\, t^2\left(1-\frac{1}{t^2}\right)^3}, \tag{16}$$

where the iso-surface thickness d_i and the support radius R_i can both be predefined according to the modeling requirements, based on which t can be used to control the blending.

As shown in Fig. 8, the three line skeletons in each sub-figure are identical and they have the same thickness and support radius. In contrast, the polygon skeleton has a uniform thickness but it is supported with smoothly varying radii. Therefore, different blending shapes are produced with different RSTs.



(a) Skeletons (b) $t_{line} = 2$ (c) $t_{line} = 1.5$ (d) $t_{line} = 1.1$

Fig. 8. Convolution surfaces based on a polygon skeleton with smoothly varying thickness ($2 \geq t_{polygon} \geq 1.2$) and three line skeletons with the same thickness. With the RST of the polygon skeleton (a) decreasing from left to right, its influence regions are constrained within smaller and smaller ranges in each sub-image (upper parts in (b)-(d)). When the polygon skeleton-based convolution surfaces are blended with three identical line skeleton-based convolution surfaces, more compact blends are created for smaller RSTs (lower parts in (b)-(d)).

4.2.2. Approximation of Edges with Varying RSTs

Background. Similar to the endpoints of line skeletons, each edge (e. g., AB in Fig. 7) of the polygon should be extended to an outer offset position $A'B'$ to achieve a more precise convolution surface approximation on the edges of polygons. If EGF and $E'GF'$ are the arcs of the support sphere intersected by polygon $ABCD$ and its offset polygon respectively, the edge offset distance u can be calculated as:

$$u = d_i - R_i \cos\left(\frac{\theta}{2}\right), \tag{17}$$

where θ represents the central angle formed by the arc $E'GF'$, $\theta = \angle E'P_2F'$.

Then, the convolution field at $\mathbf{p}_2$ can be calculated based on the approach proposed in [12]:

$$\begin{aligned} F(\mathbf{p}_2) &= \int_{V=arch_{E'GF'}} g(\mathbf{q})\, f(r)\, dV \\ &= \frac{\theta \times r_0^6}{6R_i^4} \pm \frac{\left(22 - 9\times\cos\theta + 2\times\cos^2\theta\right)\times r_0^6 \sin\theta}{90R_i^4}, \end{aligned} \tag{18}$$

Numerical Solutions for Offset Approximation. Combining Eq. 16 with Eq. 18, an equation for solving θ can be derived:

$$\begin{aligned} &\frac{\theta \times r_0^6}{6R_i^4} \pm \frac{\left(22 - 9\times\cos\theta + 2\times\cos^2\theta\right)\times r_0^6 \sin\theta}{90R_i^4} \\ &\qquad = \frac{\pi\left(1-\frac{1}{t^2}\right)^3}{3} R_i^2, \end{aligned} \tag{19}$$

where the projection radius $r_0 = R_i$. Therefore, the solutions can be derived from:

$$\begin{cases} \frac{\theta}{6} \pm \frac{\left(22-9\times\cos\theta+2\times\cos^2\theta\right)\times\sin\theta}{90} = \frac{\pi\left(1-\frac{1}{t^2}\right)^3}{3}, \\ t > 1. \end{cases} \tag{20}$$

Similar to the line skeleton situation, we firstly divide Eq. 20 into:

$$f(\theta) = g(t), \tag{21}$$

$$f(\theta) = \frac{\theta}{6} \pm \frac{\left(22 - 9\times\cos\theta + 2\times\cos^2\theta\right)\times\sin\theta}{90}, \tag{22}$$

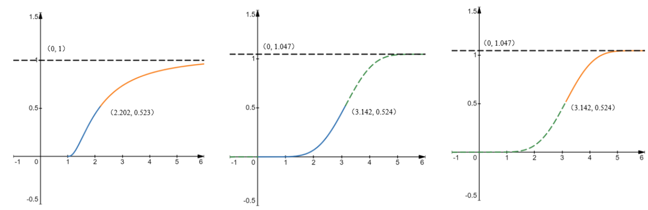
and

$$g(t) = \frac{\pi\left(1-\frac{1}{t^2}\right)^3}{3}, \tag{23}$$

where the "+" case and "-" case for $f(\theta)$ correspond to $\theta > \pi$ and $\theta < \pi$, respectively (see Eq. 17), which indicates an outward offset $u > d_i$ and $u < d_i$. Moreover, we can find the critical point for $t = 2.202$ at $\theta = \pi$. We illustrate the curves of $g(t)$ and $f(\theta)$ in Fig. 9.

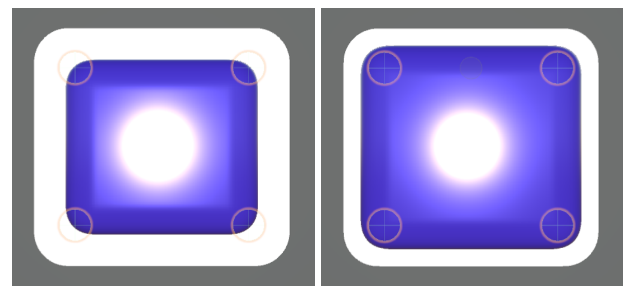
As $t \in (1, +\infty)$, $g(t)$ should fall into $g(t) \in (0, \frac{\pi}{3})$ (see Fig. 9(a)) according to Eq. 23. Once the RST t is determined, a definite numerical solution for θ can be obtained within a monotonous interval (Fig. 9(b)).

As illustrated in Fig. 10 (a), there is obvious convolution surface shrinkage on edges, which can be avoided (Fig. 10 (b)) by offsetting the embedded polygon skeleton.



(a) $g(t)$ (b) $f(\theta)$ ("-") (c) $f(\theta)$ ("+")

Fig. 9. Curves of Eq. 20



(a) Convolution surface shrinkage on edges (b) Approximation of edges of convolution surfaces based on offset polygons

Fig. 10. Approximation of the edges of polygon skeleton-based convolution surfaces. Boundary shrinkage (a) of a polygon skeleton (orange)-based convolution surface (blue) can be solved (b) using an outward offset.

5. Experiments and Results

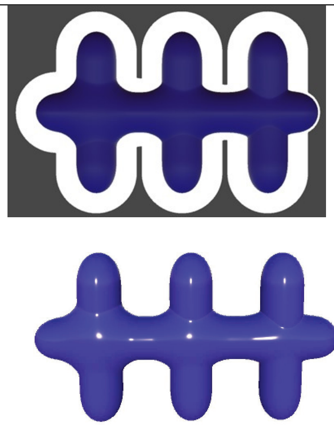
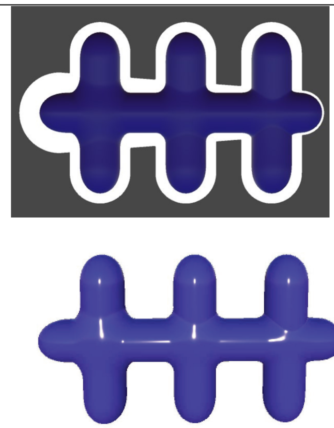
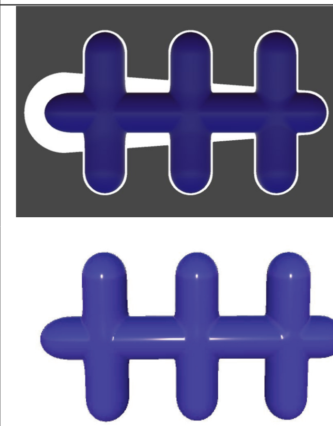
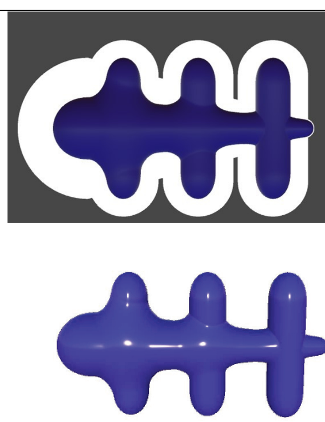
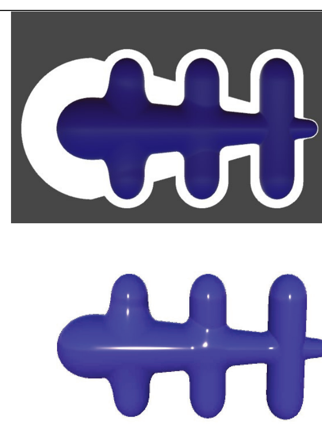
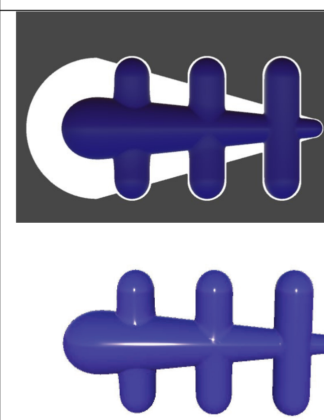
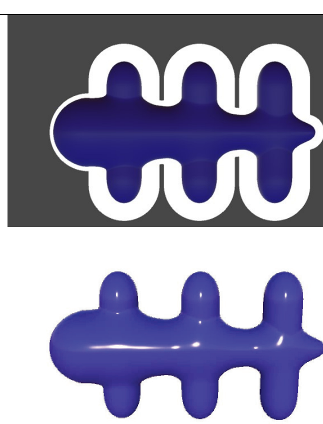
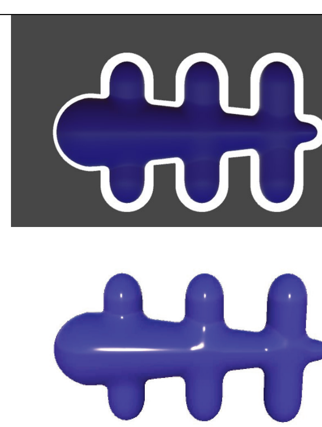
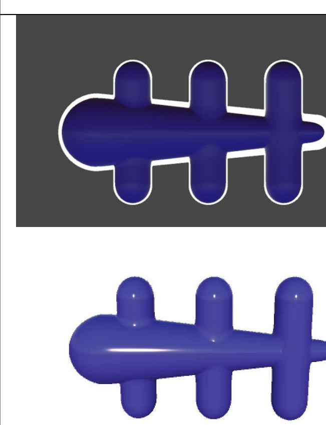
To validate our proposed approach, more experiments are performed using controllable convolution surface blending. All the results in the paper are tested on a PC with a 4.0 GHz Intel Core i7-6700K CPU with 16 GB memory. The convolution potential field calculation and the iso-surface extraction are both performed based on the Marching Cubes [34] algorithm with a resolution of $200 \times 200 \times 200$. The whole algorithm is implemented in the Unity3D engine and the core calculation steps are implemented with C# language on CPU.

5.1. Varying Blending for Line Skeletons

In order to illustrate the relationship between the blending and RST, a series of line skeleton-based convolution surfaces are created with different blending shapes in Table 1. The horizontal skeletons in the same row are all identical to each other including their RSTs at the same positions, which produce the same convolution surfaces. Three vertical line skeletons in each sub-figure are identical, and we adopt the uniform convolution surface thickness and RSTs. However, different RSTs (2, 1.5 and 1.1) in different columns are employed for blending control.

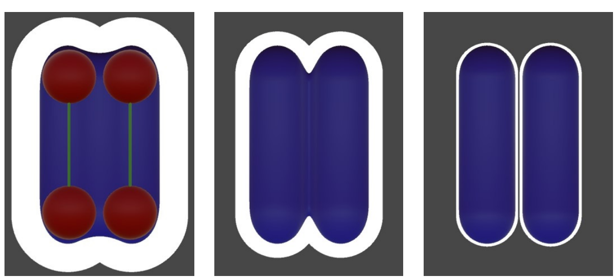
a) In the $1st$ row, linearly decreasing RSTs $2 \geq t_h \geq 1.1$ is designed, and convolution surfaces with uniform thickness are created. However, when they are blended with three identical line skeleton-based convolution surfaces, larger blending regions can be easily achieved at larger RSTs. b) In the second row, although the same ratios $2 \geq t_h \geq 1.1$ as the $1st$ row

Table 1. Convolution surfaces based on a line skeleton with smoothly varying RST (horizontal) and other three line skeletons with uniform RST (vertical)

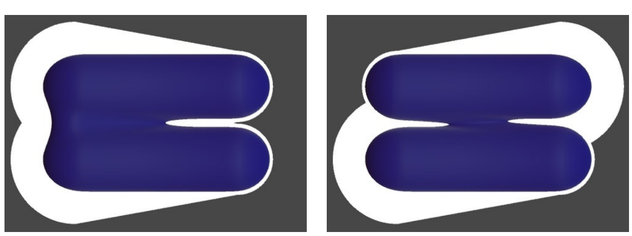
t_v / t_h	2	1.5	1.1
$2 \geq t_h \geq 1.1$			
$2 \geq t_h \geq 1.1$			
$1.1 \leq t_h \leq 2$			

are designed for the horizontal line skeletons, smoothly varying thickness along the skeletons is generated for the created convolution surfaces. It is more obvious when there are larger blending regions at larger RSTs which are placed at skeletons with larger support radii. c) The convolution surfaces in the third row are produced with a decreasing thickness similar to the second row while increasing the ratios $1.1 \leq t_h \leq 2$ which are reverse to the previous two rows. It means that for each horizontal line skeleton, the blending regions grow from left to right. Therefore it can be seen that the blending regions at the left vertical skeletons are smaller than those in the second row, while the blending regions at the right vertical skeletons are larger than those in the second row.

It is easy to see that plenty of continuous blending shapes can be achieved for the same pairs of convolution surfaces with identical shapes outside blending regions, which is beneficial to producing complex convolution surfaces.



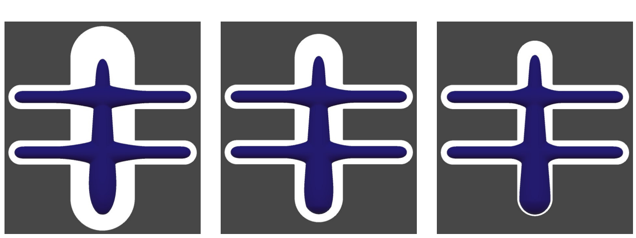
(a) $t_l = t_r = 2$ (b) $t_l = t_r = 1.5$ (c) $t_l = t_r = 1.1$



(d) $2 \geq t_{up}, t_{low} = 2 \geq 1.1$ (e) $1.1 \leq t_{up} \leq 2, 2 \geq t_{low} \geq 1.1$

Fig. 11. Line skeleton-based convolution surfaces with uniform thickness and various RST distributions.

In addition, we design a pair of parallel line skeletons in Fig. 11. All these convolution surfaces have the same thickness along the skeletons, while their blending can be positioned arbitrarily by adjusting RSTs. In (a)-(c), a uniform RST is offered for each skeleton, and the two completely blended convolution surfaces (a) depart from each other (b) and eventually are splitted into separated surfaces (c) when the ratio decreases from 2 to 1.5 and 1.1. Moreover, a bounded blend can also be achieved using varying RSTs for the skeletons (c)-(d).



(a) $8 \geq t_{vertical} \geq 2$ (b) $6 \geq t_{vertical} \geq 1.5$ (c) $4.4 \geq t_{vertical} \geq 1.1$

Fig. 12. Line skeleton-based convolution surfaces with a smoothly varying thickness and a uniform support radius (vertical) along the skeleton are blended with two identical convolution surfaces with uniform RSTs ($t_{horizontal} = 2$)

In each sub-figure of Fig. 12, a vertical convolution surface with a smoothly varying thickness and a uniform support radii along the skeleton is designed. Although the thickness of the vertical convolution surfaces increases from top to bottom, similar blending shapes can be generated when they are blended with two identical horizontal convolution surfaces, only if decreasing RSTs are applied to the vertical skeletons.

5.2. Varying Blending for Polygon Skeletons

Similar to line skeletons, several experimental cases for polygon skeletons are illustrated in Table 2, where three vertical line skeletons with uniform thickness and RSTs are blended with a horizontal polygon skeleton with a smoothly varying thickness and RSTs. To approximate the desired compound skeletons with predefined thickness in the left column, several groups of RSTs can produce similar convolution surfaces while achieving distinct blends between them. Therefore, the same convolution surfaces can be produced outside the blending regions with various blends inbetween by adjusting their RSTs.

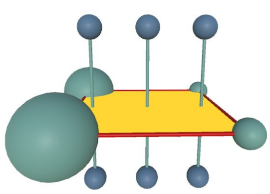
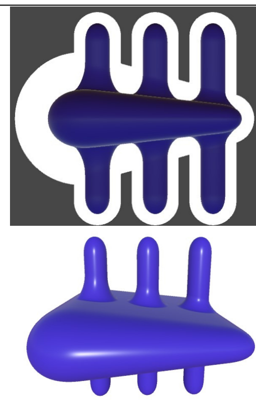
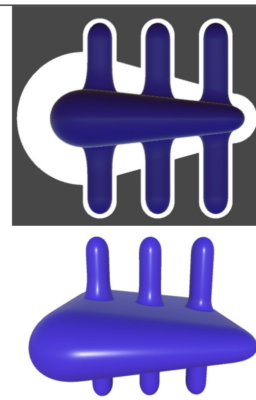
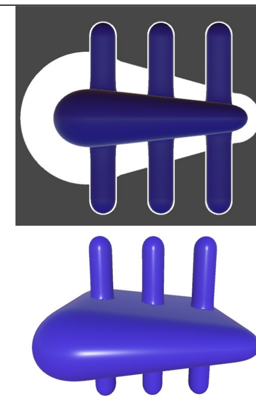
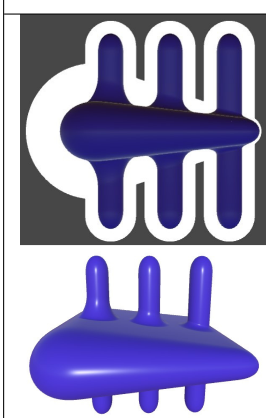
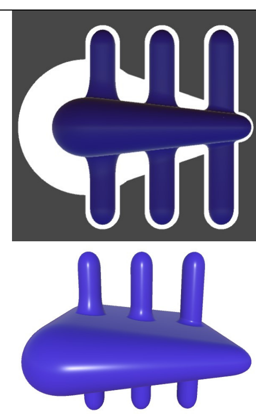
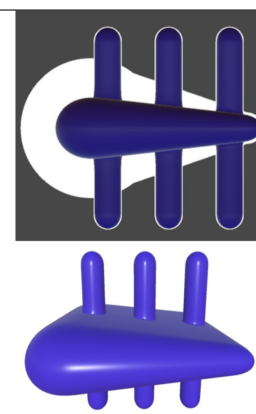
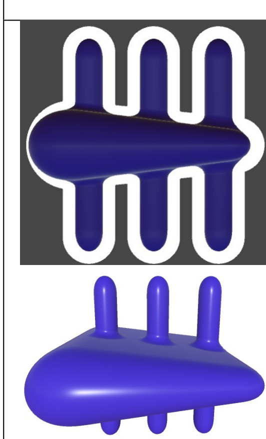
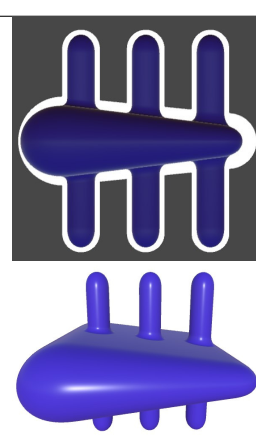
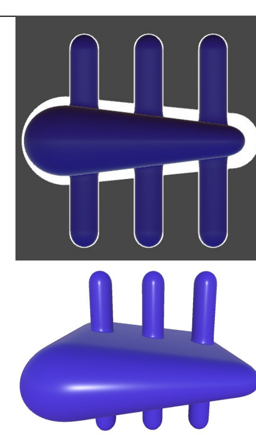
In blending regions, skeletons with larger RSTs affect other nearby skeletons more seriously, which leads to obvious blending. As shown in Table 2: a) In the 1*st* row, a horizontal convolution surface with a decreasing thickness from left to right and uniform ratios $t_{polygon} = 2$ results in larger blending regions on the left sides. b) The horizontal iso-surface with a decreasing thickness and ratio $2 \geq t_{polygon} \geq 1.1$ in the second row leads to much larger blending regions at the leftmost vertical skeletons. c) In the last row, although the horizontal convolution surfaces with a similar decreasing thickness to the previous two rows are designed, a reversely increasing ratio $1.1 \leq t_{polygon} \leq 2$ to the second row is adopted to achieve smaller blending regions for even thicker iso-surfaces on the left sides.

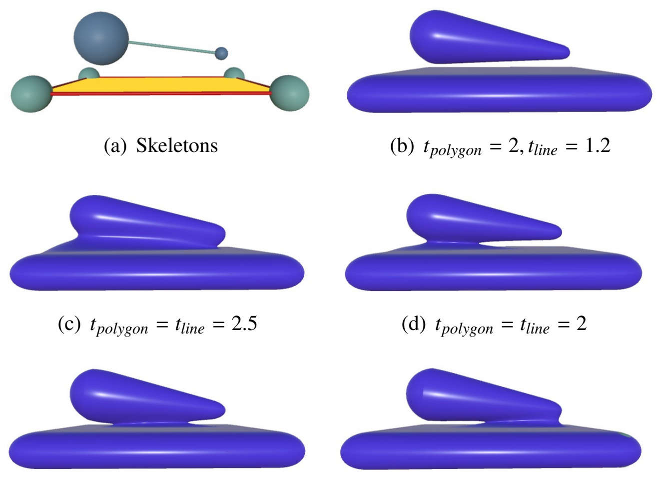
Another case for placing blending between a line skeleton and a polygon skeleton is presented in Fig. 13(a) to illustrate varying blending positions between them. A small RST separates the components cleanly (Fig.13(b)), while a larger RST merges them together (Fig.13(c)). Then by decreasing the RST at the right end, a local blending will only bridge the convolution surfaces (13(d)) on the leftmost side. After that, if the RST at the left end decreases and at the same time the RST on the right side increases, the blending position will move to the middle (13(e)) and the right side (13(f)) of the iso-surface.

5.3. Skeletal Offset for Approximation

In order to analyze the relationship between the required offsets for local convolution surface approximation of the endpoint of line skeletons, varying thickness (Fig. 14(a)-(b)) and varying RSTs (Fig. 14(c)-(d)) of convolution surface are designed separately. It can be seen that the offset is proportional to the convolution surface thickness (a-b) and RST (c-d). Especially

Table 2. Convolution surfaces based on a polygon skeleton with smoothly varying support radii and other three line skeletons with uniform support radii

Skeletons	2	1.5	1.1	t_{line} \ $t_{polygon}$
				$t_{polygon} = 2$
				$2 \geq t_{polygon} \geq 1.1$
				$1.1 \leq t_{polygon} \leq 2$



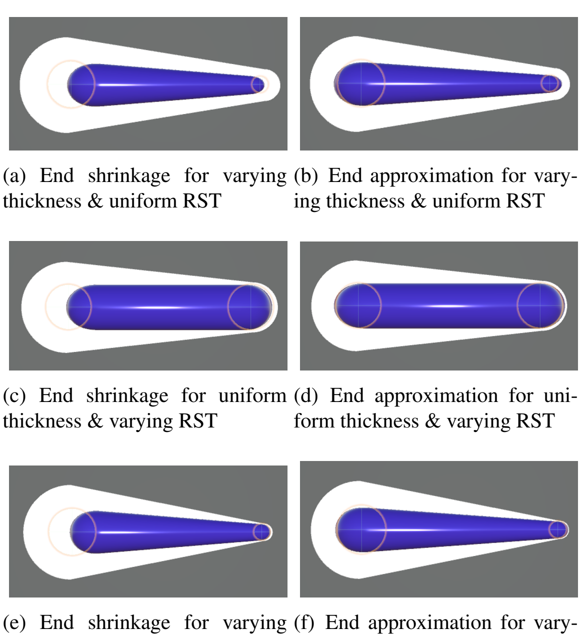
(a) Skeletons (b) $t_{polygon} = 2, t_{line} = 1.2$

(c) $t_{polygon} = t_{line} = 2.5$ (d) $t_{polygon} = t_{line} = 2$

(e) $t_{polygon} = 2, 1.2 \leq t_{line} \leq 2.5$ (f) $1.2 \leq t_{polygon} \leq 2.5, 1.3 \leq t_{line} \leq 3$

Fig. 13. Blending control with smoothly varying support radii for polygon skeletons

much larger offset is essential when a thicker convolution surface with a larger RST are applied to a part of the surface at the same time (e-f). The similar conclusions can be drawn for polygon skeleton-based convolution surfaces from Fig. 15.



(a) End shrinkage for varying thickness & uniform RST (b) End approximation for varying thickness & uniform RST

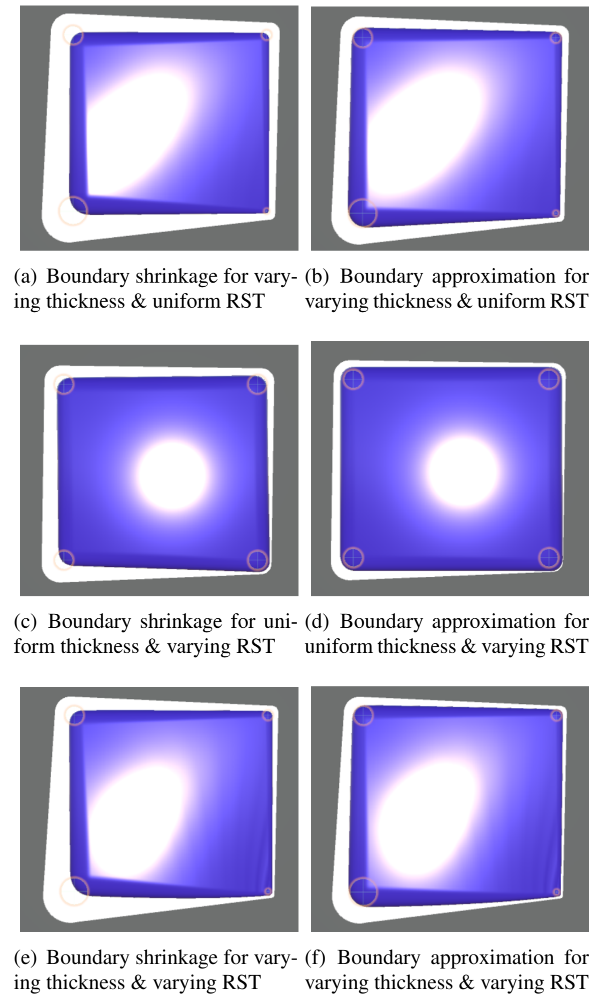
(c) End shrinkage for uniform thickness & varying RST (d) End approximation for uniform thickness & varying RST

(e) End shrinkage for varying thickness & varying RST (f) End approximation for varying thickness & varying RST

Fig. 14. The approximation of line skeletons by offsetting.

5.4. Comparisons

Comparison to No Blending Control. Prior convolution surfaces are prone to formation of unwanted merging when two disjoint skeletons are too close to each other, and it is nontrivial to avoid such artifacts without introducing extra implicit composition operators. Although such artifacts can be solved to some extent with a subdivision policy [10] and a projection point inquiry to discard skeletons far away, this method depends on a global topology of the embedded skeletons.

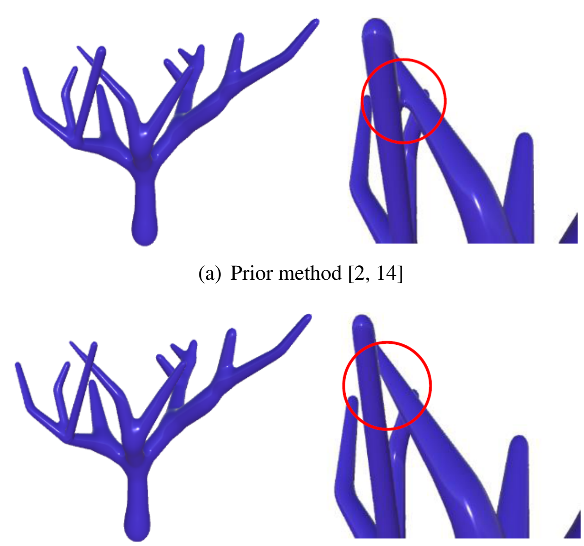


(a) Boundary shrinkage for varying thickness & uniform RST (b) Boundary approximation for varying thickness & uniform RST

(c) Boundary shrinkage for uniform thickness & varying RST (d) Boundary approximation for uniform thickness & varying RST

(e) Boundary shrinkage for varying thickness & varying RST (f) Boundary approximation for varying thickness & varying RST

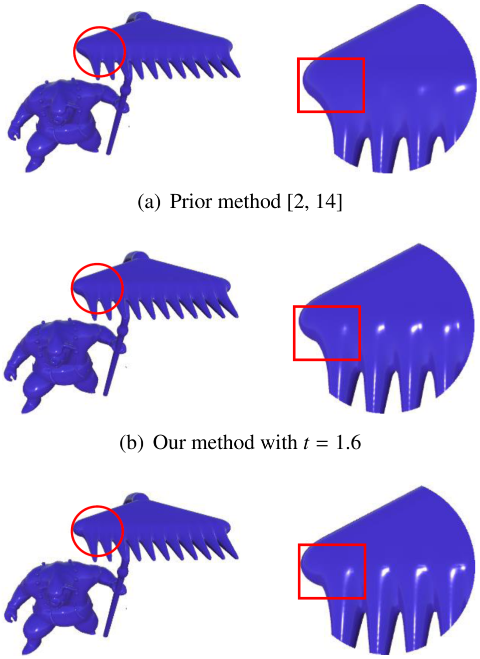
Fig. 15. Polygon skeleton boundary approximation by offset.



(a) Prior method [2, 14]

(b) Our method

Fig. 16. Unwanted blending at close branches.



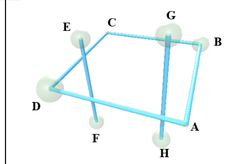
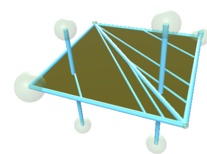
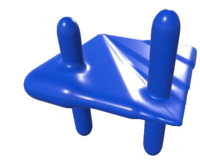
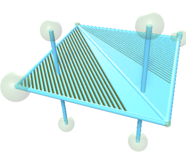
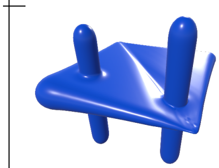
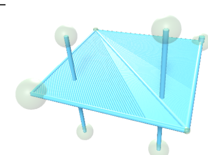
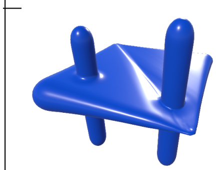
(a) Prior method [2, 14]

(b) Our method with $t = 1.6$

(c) Our method with $t = 1.3$

Fig. 17. More blending choices using our approach.

Table 3. Comparison between Zanni's method [18] and ours.

	sub-skels	Skeleton	Iso-surface	Time (s)
Ours	1			39.258
Zanni's	8			7.642
	104			29.255
	204			56.191

Here we present some comparative illustrations between our blending to prior ones with fixed RST [2, 14]. As illustrated in Fig. 16(a), unwanted merging usually arises between close skeletons when they are not topologically connected, which can be easily avoided using our controllable convolution surface blending if a small RST is used (Fig. 16(b)).

Actually, prior blending for convolution surfaces can be considered as a special case of our approach. As shown in Fig. 17(a), prior methods usually create convolution surfaces with a fixed RST, which is implemented by setting $t = 2$ for practical applications. However, various blending effects can be easily achieved by adjusting RST (Fig. 17(b)-(c)).

Comparison to Varying Blending Control. One advantage of our controllable convolution surface blending over previous approaches is the varying blending across a large skeleton, since our derived solutions are based on local convolution surface approximations developed in [2], which have been extended for thickness interpolation with barycentric coordinates [21].

In Table 3, convolution surfaces based on a polygon skeleton (horizontal) with smoothly varying support radii ($r_A : r_B : r_C : r_D = 1 : 4 : 2 : 6$) and other two line skeletons (vertical) with the same support radii ($r_C = r_D = r_E = r_F$) are presented for comparison between Zanni's [18] method and ours. As our derived solutions and the barycentric convolution surfaces [21] are both based on the same local convolution surface approximation [2], our blending possesses a similar advantage of barycentric interpolation over the semi-numerical integral policy [18]. Here varying ratios ($t_A\ t_B : t_C : t_D = 2 : 1.2 : 2 : 1.2$) of polygon $ABCD$ and an identical ratio of line skeleltons EF and GH are applied for interpolating blending on $ABCD$. It is obvious that our solutions are suitable for smooth controllable convolution surface blending without intensive computation, as only

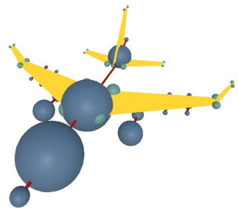
Table 4. Timings for experiment results.

Fig.1	(a)	$(a)_c$	$(c)_c$			
Time(s)	0.859	1.158	1.007			
Fig.8	(b)	$(b)_c$	$(c)_c$	$(d)_c$		
Time(s)	3.636	42.204	42.417	41.571		
Fig.13	(d)	$(d)_c$	$(b)_c$	$(c)_c$	$(e)_c$	$(f)_c$
Time(s)	3.559	42.252	42.563	43.388	41.651	42.068
Fig.18	(b)	$(b)_c$	$(c)_c$			
Time(s)	16.398	173.302	169.124			
Fig.19	(b)	$(b)_c$				
Time(s)	3.184	20.052				

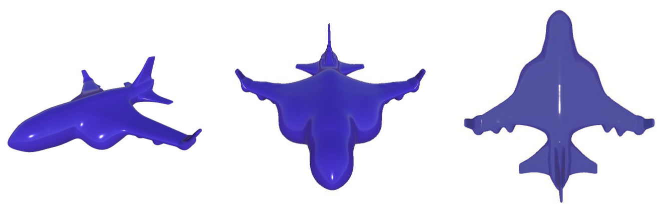
one control parameter RST has to be calculated. However, for semi-numerical integrals [18], the polygon skeleton $ABCD$ has to be subdivided into smaller sub-polygons [18] to achieve an interpolated blending, which usually costs more time for an acceptable smooth blending interpolation.

5.5. Applications.

Using one group of compound skeletons of an airplane in Fig. 18(a), a series of convolution surfaces with varying shapes are produced while preserving the intrinsic superposition property. A large RST leads to a fat cartoon plane (Fig. 18(b)) while a thinner plane with a compact silhouette can be created based on a smaller ratio (Fig. 18(c)). Moreover, a thin plane with locally-inflated blending wings can be generated using large RSTs at wings and smaller ones for other places as illustrated in Fig. 18(d-e). From Fig. 19, it is easy to see that blending regions can be easily placed arbitrarily along a line skeleton of the back of a chair. The blending between the pairs of vertical line skeletons at the chair back and vertical line skeletons at chair handles can be achieved by increasing RSTs at desired blending regions while keeping the thickness of their original convolution surface.

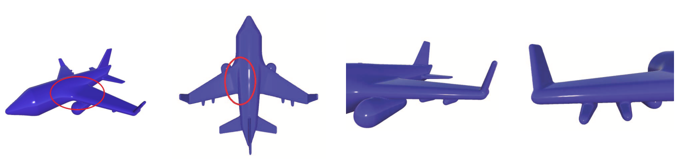


(a) Skeletons

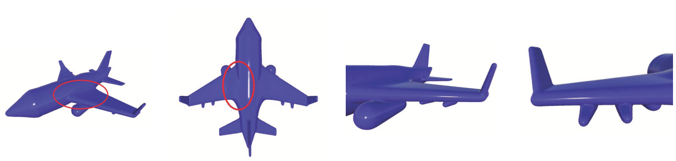


(b) $t = 2$

(c) $t = 1.5$

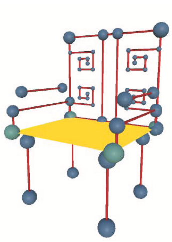


(d) Locality 1



(e) Locality 2

Fig. 18. A series of airplanes



(a) Skeletons

(b) Convolution surfaces

Fig. 19. A chair example

5.6. *Timings for experiment results*

As all our experiment results are created in the Unity3D engine with C# language, we list some timings for the convolution field computation and iso-surface extraction using Marching Cubes in Table. 4. In the table, we present two versions

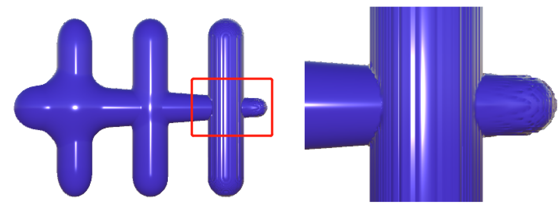
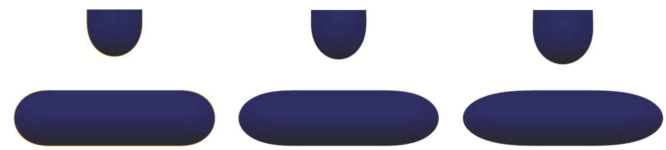


Fig. 20. Artifacts due to too small RST for low iso-surface polygonization resolution



(a) $t_{horizontal} = 1.2$ (b) $t_{horizontal} = 4$ (c) $t_{horizontal} = 10$

Fig. 21. Surface shrinkage for too large RST ($t_{vertical} = 1.2$)

of convolution surface blending with uniform RSTs in the first column and the second column, which represent surfaces without and with controllable blending (a special case of our varying RSTs) respectively. Due to the interpolation of the field contribution weight of each sampling position in the Marching Cubes, the computations with controllable blending (the second column) require more time which is 1.3∼11.9 times of the ones without controllable blending (the first column). Similarly, the timings in the columns behind the second column are almost the same as the second column due to the same interpolations.

6. Conclusion and Future Work

In this paper, a controllable convolution surface blending approach is proposed based on RST. Our method allows for varying RSTs along a line skeleton or within a polygon skeleton, and various blending effects can be produced as desired. Moreover, as no extra blending technique is involved, our method preserves the superposition property of convolution surface, which enable successive convolution field compositions.

Our method has limitations. As shown in Fig. 20, artifacts may arise if a voxel-based iso-surface extraction with a too low resolution is employed when a very small RST is adopted in our blending. Another limitation is that a too large support radius ratio will affect both its own end shapes and other nearby convolution surface components (as illustrated in Fig. 21), therefore too large RST is not recommended in practical applications. Finally, our method focuses on line and polygon skeleton-based convolution surfaces only, and it is worthwhile to extend our approach to curve, surface and volume skeleton-based convolution surfaces.

References

[1] Zanni, C. Skeleton-based implicit modeling and applications. Ph.D. thesis; Grenoble; 2013.

[2] Zhu, X, Song, L, You, L, Zhu, M, Wang, X, Jin, X. Brush2model: Convolution surface-based brushes for 3d modelling in head-mounted display-based virtual environments. Computer Animation and Virtual Worlds 2017;28(3-4).
[3] Bernhardt, A, Barthe, L, Cani, M, Wyvill, B. Implicit blending revisited. Comput Graph Forum 2010;29(2):367–375.
[4] Bloomenthal, J, Shoemake, K. Convolution surfaces. In: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991. 1991, p. 251–256.
[5] Jin, X, Tai, CL. Convolution surfaces for arcs and quadratic curves with a varying kernel. The Visual Computer 2002;18(8):530–546.
[6] Hubert, E, Cani, M. Convolution surfaces based on polygonal curve skeletons. Journal of Symbolic Computing 2012;47(6):680–699.
[7] Zhu, X, Jin, X, Liu, S, Zhao, H. Analytical solutions for sketch-based convolution surface modeling on the GPU. The Visual Computer 2012;28(11):1115–1125.
[8] Bernhardt, A, Pihuit, A, Cani, MP, Barthe, L. Matisse: painting 2d regions for modeling free-form shapes. In: EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling, SBIM'08. 2008, p. 57–64.
[9] Sherstyuk, A. Kernel functions in convolution surfaces: a comparative analysis. The Visual Computer 1999;15(4):171–182.
[10] Hornus, S, Angelidis, A, Cani, M. Implicit modeling using subdivision curves. The Visual Computer 2003;19(2-3):94–104.
[11] Zhu, X, Jin, X, You, L. High-quality tree structures modelling using local convolution surface approximation. The Visual Computer 2015;31(1):69–82.
[12] Jin, X, Tai, CL, Zhang, H. Implicit modeling from polygon soup using convolution. The Visual Computer 2009;25(3):279–288.
[13] Hubert, E. Convolution surfaces based on polygons for infinite and compact support kernels. Graphical Models 2012;74(1):1–13.
[14] Su¢rez, AJF, Hubert, E. Convolution surfaces with varying radius: Formulae for skeletons made of arcs of circles and line segments. Research in Shape Analysis 2018;:37–60.
[15] Jin, X, Tai, CL. Analytical methods for polynomial weighted convolution surfaces with various kernels. Computers & Graphics 2002;26(3):437–447.
[16] Jin, X, Tai, C, Feng, J, Peng, Q. Convolution surfaces for line skeletons with polynomial weight distributions. Journal of Graphics Tools 2001;6(3):17–28.
[17] Suárez, AJF, Hubert, E, Zanni, C. Anisotropic convolution surfaces. Computers & Graphics 2019;82:106 – 116.
[18] Zanni, C, Bernhardt, A, Quiblier, M, Cani, M. Scale-invariant integral surfaces. Computer Graphics Forum 2013;32(8):219–232.
[19] Gupta, V, Tandon, P. Heterogeneous object modeling with material convolution surfaces. Computer-Aided Design 2015;62:236–247.
[20] Schmeißer, A, Wegener, R, Hietel, D, Hagen, H. Smooth convolution-based distance functions. Graphical Models 2015;82:67–76.
[21] Zhu, X, Song, C, Zhu, M, Wang, X, You, L, Jin, X. Barycentric convolution surfaces based on general planar polygon skeletons. Graph Model 2020;109:101069.
[22] Müller, M, Charypar, D, Gross, MH. Particle-based fluid simulation for interactive applications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, July 26-27, 2003. 2003, p. 154–159.
[23] Tai, C, Zhang, H, Fong, JC. Prototype modeling from sketched silhouettes based on convolution surfaces. Computer Graphics Forum 2004;23(1):71–84.
[24] Angles, B, Tarini, M, Wyvill, B, Barthe, L, Tagliasacchi, A. Sketch-based implicit blending. ACM Trans Graph 2017;36(6):181:1–181:13.
[25] Rohmer, D, Popa, T, Cani, M, Hahmann, S, Sheffer, A. Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles. ACM Trans Graph 2010;29(6):157.
[26] Vaillant, R, Barthe, L, Guennebaud, G, Cani, M, Rohmer, D, Wyvill, B, et al. Implicit skinning: real-time skin deformation with contact modeling. ACM Trans Graph 2013;32(4):125:1–125:12.
[27] Barthe, L, Gaildrat, V, Caubet, R. Combining implicit surfaces with soft blending in a csg tree. In: CSG Conference Series, Ammerdown, U.K. 1998, p. 17–31.
[28] Pasko, G, Pasko, AA, Kunii, TL. Bounded blending for function-based shape modeling. IEEE Computer Graphics and Applications 2005;25(2):36–45.
[29] Gourmel, O, Barthe, L, Cani, M, Wyvill, B, Bernhardt, A, Paulin, M, et al. A gradient-based implicit blend. ACM Trans Graph 2013;32(2):12:1–12:12.
[30] Canezin, F, Guennebaud, G, Barthe, L. Adequate inner bound for geometric modeling with compact field functions. Computers & Graphics 2013;37(6):565–573.
[31] Zanni, C, Gleicher, M, Cani, M. N-ary implicit blends with topology control. Computers & Graphics 2015;46:1–13.
[32] Wyvill, B, Wyvill, G. Field functions for implicit surfaces. Vis Comput 1989;5(1&2):75–82. URL: `https://doi.org/10.1007/BF01901483`. doi:`10.1007/BF01901483`.
[33] Wyvill, B, Guy, A, Galin, E. Extending the CSG tree - warping, blending and boolean operations in an implicit surface modeling system. Computer Graphics Forum 1999;18(2):149–158.
[34] Lorensen, WE, Cline, HE. Marching cubes: A high resolution 3d surface construction algorithm. In: Stone, MC, editor. Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987. ACM; 1987, p. 163–169.