Vol. 6, No. 3: 00–00

Convolution Surfaces for Line Skeletons with Polynomial Weight Distributions

Xiaogang Jin
State Key Lab of CAD & CG, Zhejiang University

Chiew-Lan Tai
Hong Kong University of Science and Technology

Jieqing Feng and Qunsheng Peng
State Key Lab of CAD & CG, Zhejiang University

Abstract. Convolution surfaces generalize point-based implicit surfaces to incorporate higher-dimensional skeletal elements; line segments can be considered the most fundamental skeletal elements since they can approximate curve skeletons. Existing analytical models for line-segment skeletons assume uniform weight distributions, and thus they can produce only constant-radius convolution surfaces. This paper presents an analytical solution for convolving line-segment skeletons with a variable kernel modulated by a polynomial function, allowing generalized cylindrical convolution surfaces to be modeled conveniently. Its computational requirement is competitive with that of uniform weight distribution. The source code of the field computation is available online.

1. Introduction

An implicit surface is a good representation for modeling and animating a smooth deformable object of complex topology that may change over time, such as liquid, snow, cloud, and organic shapes [Bloomenthal et al. 97],

1086-7651/01 $0.50 per page

[Dobashi et al. 00], [Cani, Desbrun 97], [Nishita et al. 97], [Jin et al. 00]. A skeleton-based implicit surface, S, is most commonly defined as an iso-surface satisfying the equation,

$$S = \{(x, y, z) | \sum F_i(x, y, z) - T = 0\}, \tag{1}$$

where F_i is the field function of the i^{th} contribution source and T is the threshold field value. For example, the implicit functions in metaballs (blobs or soft objects) [Blinn 82], [Nishimura et al. 85], [Wyvill et al. 86], [Wyvill, Wyvill 89] are defined as summations of point fields. Metaballs are widely implemented in commercial software packages (e.g., Softimage, 3D Studio Max), but they are inadequate for representing flat surfaces and generalized cylinders. A distance surface allows the use of higher-dimensional skeletons [Bloomenthal, Wyvill 90], [Bloomenthal 95]; however, creases or curvature discontinuities may arise when dealing with multiple non-convex skeletal primitives.

Bloomenthal and Shoemake [Bloomenthal, Shoemake 91] present convolution surfaces as natural and powerful extensions to point-based field surfaces. By convolving skeletons with a three-dimensional low-pass Gaussian filter kernel, the resulting iso-surfaces overcome the above-mentioned weakness of distance surfaces [Bloomenthal 97]. The skeletal elements in convolution surfaces can be points, line segments, curves, polygons, and other geometric primitives. Since skeletons are natural abstractions of shapes, convolution surfaces offer a convenient means of controlling the shape of the underlying modeling object.

The modeling potential of convolution surfaces is very attractive, but their mathematical formulation still poses some open problems, stemming from the fact that convolution integrals seldom yield closed-form solutions that can be directly evaluated. The derivation of closed-form solutions depends on both the kernel function and the skeletal element. Bloomenthal and Shoemake [Bloomenthal, Shoemake 91] calculate the field function numerically using a point-sampling method, which, potentially could under-sample artifacts. By employing a new kernel function—the Cauchy function—McCormack and Sherstyuk [McCormack, Sherstyuk 98], [Sherstyuk 99] deduce analytical solutions for point, line segment, polygon, arc, and plane elements. Their method thus provides an accurate and robust solution for more generally shaped convolution surfaces.

The analytical line-segment model derived by McCormack and Sherstyuk assumes uniform weight distribution and produces only constant-radius convolution surfaces around the line-segment skeleton. Thus, multiple line segments must be specified to model surfaces of varying radius. To overcome this problem, this paper presents a closed-form model for line-segment skeletons with weight distribution modulated by a polynomial. Since many objects can be abstracted with curve skeletons, which can in turn be approximated

by line segments, our closed-form solution facilitates the modeling of a large variety of generalized cylindrical shapes[1].

2. Convolution Surface

A convolution surface is an iso-surface in a scalar field implicitly defined by a skeleton consisting of three-dimensional points and by a potential function representing the contribution of each skeletal point to the scalar field. In this paper, we adopt the following convolution surface definition, given by McCormack and Sherstyuk [McCormack, Sherstyuk 98]: Let $\mathbf{P}(x, y, z)$ be a point in R^3, and let $g : R^3 \rightarrow R$ be the *geometry function* that represents a modeling skeleton V:

$$g(\mathbf{P}) = \begin{cases} 1, & \mathbf{P} \in \text{ skeleton } V; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let $f : R^3 \rightarrow R$ be a potential function generated by a single point in the skeleton V, and let $\mathbf{Q}$ be a point in the skeleton. The total field, F, contributed by the skeleton at a point, $\mathbf{P}$, is the convolution of two functions, f and g,

$$F(\mathbf{P}) = \int_V g(\mathbf{Q}) f(\mathbf{P} - \mathbf{Q}) dV = (f \otimes g)(\mathbf{P}), \tag{3}$$

Thus, f is also called the *convolution kernel*. For convenience, we rewrite the field function of the convolution surface as the volume integral along the skeleton,

$$F(\mathbf{P}) = \int_V f(\mathbf{P} - \mathbf{Q}) dV. \tag{4}$$

In this paper, we adopt the Cauchy kernel function proposed by McCormack and Sherstyuk [McCormack, Sherstyuk 98],

$$f(\mathbf{P} - \mathbf{Q}) = \frac{1}{(1 + s^2 r^2)^2}, \tag{5}$$

where $r = ||\mathbf{P} - \mathbf{Q}||$, and s is a parameter for controlling the width of the kernel. The field function $F(\mathbf{P})$ now becomes

$$F(\mathbf{P}) = \int_V \frac{dV}{(1 + s^2 r^2)^2}. \tag{6}$$

[1]Grimm [Grimm 99] has also investigated implicit generalized cylinders; her method involves specifying an axis and one or more profile curves. In contrast, our method produces exact convolution surfaces and extends the versatility of modeling with convolution surfaces.

Superposition is one of the most important properties of convolution surfaces; that is, summing the convolution surfaces generated by two separate skeletons yields the same surface as that generated by their combined skeleton. This property allows convolution surfaces to overcome the problem of bulges and creases encountered in distance surfaces. The independent evaluation feature guarantees that the user need only be concerned with the shape of the skeleton, not the number of segments used, when designing a convolution surface. The superposition property ensures that the skeletons can be subdivided arbitrarily and that the field function of the sub-skeletons can be summed simply to evaluate the final convolution surface.

3. Line-Segment Skeleton with Polynomial Weight Distributions

We multiply the field function at a point, $\mathbf{Q}$, by $q(\mathbf{Q}) : R^3 \rightarrow R$ to define a weighted convolution surface model with non-uniform weight distributions:

$$F(\mathbf{P}) = \int_V q(\mathbf{Q}) f(\mathbf{P} - \mathbf{Q}) dV = \int_V \frac{q(\mathbf{Q}) dV}{(1 + s^2 r^2)^2}. \tag{7}$$

We now derive the analytical formulae for the line-segment skeleton with polynomial weight distributions. A line segment of length l, with starting point, $\mathbf{b}$, and unit direction, $\mathbf{a}$, can be represented parametrically as

$$\mathbf{L}(t) = \mathbf{b} + t\mathbf{a}, \qquad 0 \leq t \leq l. \tag{8}$$

Letting $\mathbf{d} = \mathbf{P} - \mathbf{b}$, the squared distance from a point, $\mathbf{P}$, to a point on the line $\mathbf{L}(t)$ is then

$$r^2(t) = d^2 + t^2 - 2t\mathbf{d} \cdot \mathbf{a}, \tag{9}$$

where $d = ||\mathbf{d}||$. Let $F_{\text{line}}^{t^i}(\mathbf{P})$ denote the field function of the line segment $\mathbf{L}(t)$ with weight distribution t^i. For $i = 0, 1, 2, 3$, we obtain

$$\begin{aligned} F_{\text{line}}^{1}(\mathbf{P}) &= \int_0^l \frac{dt}{(1 + s^2 r^2(t))^2}, \quad F_{\text{line}}^{t}(\mathbf{P}) = \int_0^l \frac{t dt}{(1 + s^2 r^2(t))^2}, \\ F_{\text{line}}^{t^2}(\mathbf{P}) &= \int_0^l \frac{t^2 dt}{(1 + s^2 r^2(t))^2}, \quad F_{\text{line}}^{t^3}(\mathbf{P}) = \int_0^l \frac{t^3 dt}{(1 + s^2 r^2(t))^2}. \end{aligned} \tag{10}$$

Substituting $r^2(t)$ into $F^1_{\text{line}}(\mathbf{P})$, letting $h = \mathbf{d}\cdot\mathbf{a}$, and integrating, we obtain

$$\begin{aligned} F^1_{\text{line}}(\mathbf{P}) &= \int_0^l \frac{dt}{(1+(s^2(t-h)^2+s^2d^2-s^2h^2))^2} \\ &+ \frac{1}{s^4}\int_{-h}^{l-h} \frac{dt}{(t^2+(p/s)^2)^2} \\ &= \frac{1}{2p}\left[\frac{h}{s^2h^2+p^2}+\frac{l-h}{s^2(l-h)^2+p^2}\right] \\ &\quad +\frac{1}{2sp^3}\left(\arctan\left[\frac{sh}{p}\right]+\arctan\left[\frac{s(l-h)}{p}\right]\right), \end{aligned} \tag{11}$$

where p is a distance term: $p^2 = 1+s^2(d^2-h^2)$.

Similarly, we can derive the analytical formulae for $F^t_{\text{line}}(\mathbf{P})$, $F^{t^2}_{\text{line}}(\mathbf{P})$, $F^{t^3}_{\text{line}}(\mathbf{P})$ as follows:

$$\begin{aligned} F^t_{\text{line}}(\mathbf{P}) &= \int_0^l \frac{tdt}{(s^2(t-h)^2+p^2)^2} \\ &= \frac{1}{2s^2}\left[\frac{1}{s^2h^2+p^2}-\frac{1}{s^2(l-h)^2+p^2}\right]+hF^1_{\text{line}}(\mathbf{P}); \end{aligned} \tag{12}$$

$$\begin{aligned} F^{t^2}_{\text{line}}(\mathbf{P}) &= \int_0^l \frac{t^2dt}{(s^2(t-h)^2+p^2)^2} \\ &= -\frac{1}{2s^2}\left[\frac{h}{s^2h^2+p^2}+\frac{l-h}{s^2(l-h)^2+p^2}\right] \\ &\quad +\frac{1}{2ps^3}\left(\arctan\left[\frac{sh}{p}\right]+\arctan\left[\frac{s(l-h)}{p}\right]\right) \\ &\quad -h^2F^1_{\text{line}}(\mathbf{P})+2hF^t_{\text{line}}(P) \\ &= \frac{1}{s^3p}(\arctan\left[\frac{sh}{p}\right]+\arctan\left[\frac{s(l-h)}{p}\right]) \\ &\quad -(h^2+(p/s)^2)F^1_{\text{line}}(\mathbf{P})+2hF^t_{\text{line}}(\mathbf{P}); \end{aligned} \tag{13}$$

$$\begin{aligned} F^{t^3}_{\text{line}}(\mathbf{P}) &= \int_0^l \frac{(t-h)^3dt}{(s^2(t-h)^2+p^2)^2}+h^3F^1_{\text{line}}(\mathbf{P})-3h^2F^t_{\text{line}}(\mathbf{P})+3hF^{t^2}_{\text{line}}(\mathbf{P}) \\ &= -\frac{1}{2s^4}\left[\ln\frac{s^2(l-h)^2+p^2}{s^2h^2+p^2}\right]-\frac{p^2}{2s^4}\left[\frac{l}{s^2(h)^2+p^2}-\frac{1}{s^2(1-h)^2+p^2}\right] \\ &\quad +h^3F^1_{\text{line}}(\mathbf{P})-3h^2F^t_{\text{line}}(\mathbf{P})+3hF^{t^2}_{\text{line}}(\mathbf{P}). \end{aligned} \tag{14}$$

The analytical formulae for $F^{t^i}_{\text{line}}(\mathbf{P})$ with $i \geq 4$ can be derived analogously; however, since cubic polynomials are usually sufficient for user requirements, we omit these formulae here. Note that $F^{t^i}_{\text{line}}(\mathbf{P})$ can be represented in terms of $F^{t^{i-1}}_{\text{line}}(\mathbf{P}), F^{t^{i-2}}_{\text{line}}(\mathbf{P}) \ldots . F^{t^1}_{\text{line}}(\mathbf{P})$; this enables the use of more efficient incremental calculations. Based on the closed-form field functions of a line segment with weight distributions $1, t, t^2, t^3$, we can now use a cubic polynomial function to define a profile distribution function along the skeletal line segment. To provide an intuitive interface for controlling the cubic curve, we represent it in the Bézier form with control points $(j/3, q_j)$ where $j = 0, \ldots, 3$. This form of Bézier curve can be rewritten simply as $q(u) = \sum q_j B_j(u)$, where the $B_j(u)$ are cubic Bernstein basis functions [Farin 97]. The designed profile Bézier curve is then converted to power basis form.

For linear weight distribution (most frequently used in practice), the formula is very simple. Let the weights at the start and end points of the line segment $\mathbf{L}(t)$ be $\tilde{q_0}$ and $\tilde{q_1}$ respectively; then the weight at parameter t is

$$q(t) = \tilde{q_0} + \frac{\tilde{q_1} - \tilde{q_0}}{l}\, t, \tag{15}$$

and the field function of the entire line segment is

$$F_{\text{line}}(\mathbf{P}) = \tilde{q_0} F^1_{\text{line}}(\mathbf{P}) + \frac{\tilde{q_1} - \tilde{q_0}}{l} F^t_{\text{line}}(\mathbf{P}). \tag{16}$$

We now derive the field function of a skeletal polyline, $\mathbf{P}_0\mathbf{P}_1 \ldots \mathbf{P}_n$, with linear weight distributions, given that the weights at $\mathbf{P}_0$ and $\mathbf{P}_n$ are $\tilde{q_0}$ and $\tilde{q_1}$, respectively. Let the length of the i^{th} segment, $\mathbf{P}_i\mathbf{P}_{i+1}$, be $l_i = ||\mathbf{P}_{i+1} - \mathbf{P}_i||$, $i = 0, 1, \ldots, n-1$, and denote the field functions along the line segment $\mathbf{P}_i\mathbf{P}_{i+1}$ with weight distributions l, t as ${}^iF^1_{\text{line}}(\mathbf{P})$, ${}^iF^t_{\text{line}}(\mathbf{P})$, respectively. It is easy to derive the weight at point $\mathbf{P}_i$, denoted by q_i, as follows:

$$q_i = \tilde{q_0} + \left(\sum_{j=0}^{i-1} l_j / \sum_{j=0}^{n-1} l_j \right) (\tilde{q_1} - \tilde{q_0}). \tag{17}$$

Thus, the field function for the entire polyline is

$$F_{\text{polyline}}(\mathbf{P}) = \sum_{i=0}^{n-1} \left\{ q_i \; {}^iF^1_{\text{line}}(\mathbf{P}) + \frac{(q_{i+1} - q_i)}{l_i} \, {}^iF^t_{\text{line}}(\, \mathbf{P}) \right\}. \tag{18}$$

4. Examples and Performance

To demonstrate the capabilities of our method, we have modeled a variety of surfaces. For uniform processing, all the convolution models are first polygonized into polygon meshes, which are then ray-traced with solid or projective

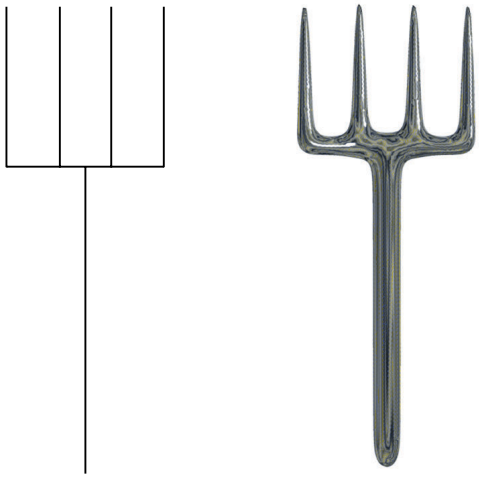


Figure 1. Hayfork. (a) Skeleton. (b) Convolution surface.

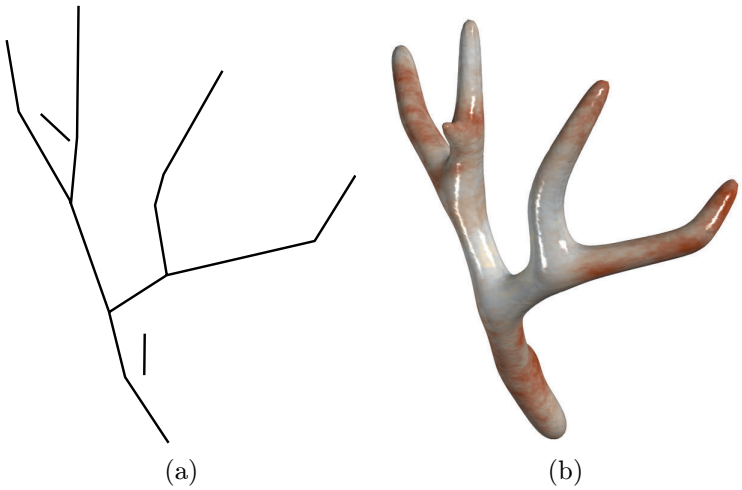
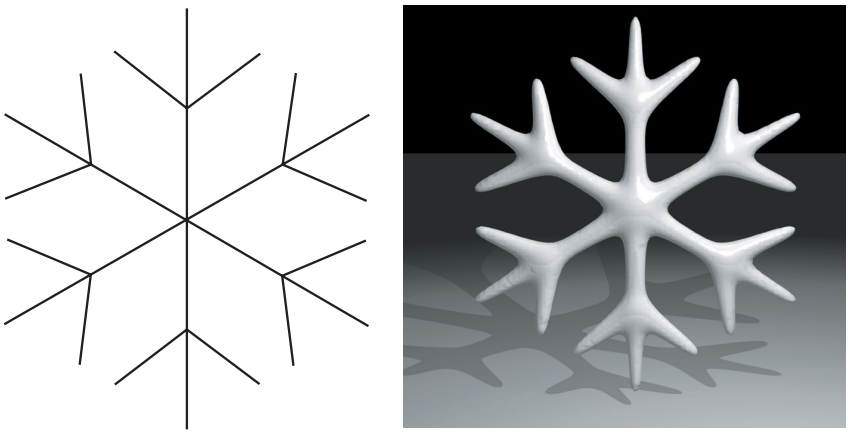


Figure 2. Deer horn. (a) Skeleton. (b) Convolution surface.

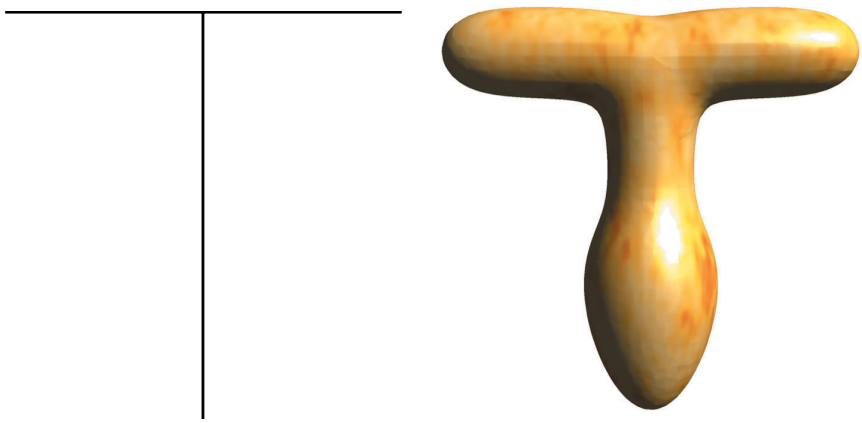
texture mapping. Figures 1–3 illustrate convolution surfaces that adopt linear weight distributions. The underlying skeletal representation is shown on the left of each illustration. The hayfork, deer horn, and snowflake, respectively, are modeled using 6, 15, and 18 line segments. Figure 4 shows a model that uses a cubic polynomial distribution for the vertical line segment.



(a) (b)

Figure 3. Snowflake. (a) Skeleton. (b) Convolution surface.

A more complex example—a maple tree—is illustrated in Figure 5; the stem and main branches are modeled using polyline skeletons with linear weight functions (the twigs and leaves are modeled using traditional modeling methods). The branches of the potted plant in Figure 6 are modeled in a similar way (the leaves are modeled separately before being added to the branches). These examples demonstrate the convenience of using our analytical methods to model convolution surfaces of generalized cylindrical shapes.



(a) (b)

Figure 4. T-shaped object. (a) Skeleton. (b) Convolution surface.

Figure 5. Maple tree.

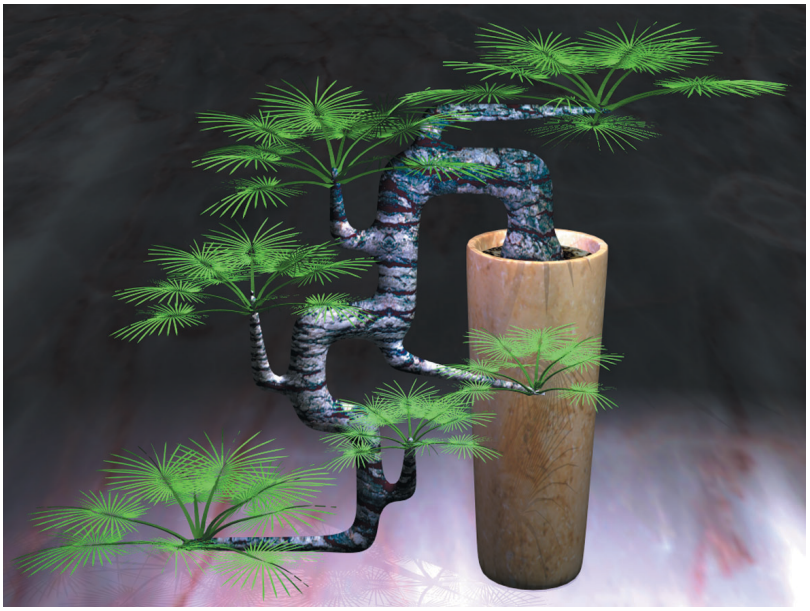


Figure 6. Potted plant.

Weight distribution	Special functions and floating-point operations						
	arctan	sqrt	ln	*	/	+	-
1	2	1	0	17	5	10	5

Table 1. Computational costs for a line segment convolved with constant distribution.

Tables 1 and 2 show the computational efficiency of our algorithm. Table 1 gives the number of special function calls and floating-point operations required in calculating the field function for a line segment with constant distribution; optimizations have been performed to reduce the number of operations. Table 2 gives the incremental operations required for a line segment with weight distributions t, t^2, t^3. From these tables, we can conclude that the incremental cost from constant distribution to linear distribution is nominal–only five additional multiplications/divisions and two additions/subtractions are needed. Even for cubic polynomial distribution, the additional computational cost is only one ln operation, 24 multiplications/divisions, and 10 additions/subtractions, which is less than twice the cost of constant distribution.

Weight distribution	Incremental special functions and floating-point operations						
	arctan	sqrt	ln	*	/	+	-
t	0	0	0	2	3	1	1
t^2	0	0	0	4	2	2	1
t^3	0	0	1	8	5	2	3

Table 2. Incremental computational costs for a line segment convolved with t, t^2, t^3 distribution.

5. Discussion

Skeleton-based convolution surface modeling can create and animate a wide variety of complex objects, which may be difficult with parametric geometric modeling methods. Since curve skeletons are good abstractions for a wide variety of natural forms and they can be approximated by polylines, our method is rather general in its applicability. Combined with other skeletal elements, our method facilitates the modeling of trees, sea-forms, and other organic shapes. By defining the tapering factors in grammar rules, our method also can be incorporated easily into string rewriting systems (L-systems) to describe the geometric shapes of plant trunks [Prusinkiewicz, Lindenmayer 90].

Since convolution surfaces are iso-surfaces that are determined by skeletons, profile curves, and threshold field values, our method has the following limitations: (1) Unlike traditional sweeping operations, the profile curve only approximates that of the generated generalized cylinder, because the profile curve in general does not lie on the iso-surface. Thus precise specification of the radius of the generalized cylinder is difficult; (2) If the threshold exceeds the field of some part of a skeleton, no convolution surface is produced at that part; this requires special attention and is inconvenient in some applications.

Acknowledgements. Part of this research work was conducted while the first author was a visiting researcher at the Hong Kong University of Science and Technology. This work received support from Hong Kong Research Grant Council (HKUST6215/99E), National Natural Science Foundation of China (Grant No. 69973040), Zhejiang Provincial Natural Science Foundation (Grant No. 698022) and Innovative Research Groups (Grant No. 60021201). The authors are grateful to Andrei Sherstyuk for his help and discussions on convolution surfaces.

References

[Blinn 82] J. Blinn. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics* 1(3): 235–256 (1982).

[Bloomenthal 95] J. Bloomenthal. *Skeletal Design of Natural Forms*. Ph.D thesis, University of Calgary, Dept. of Computer Science, 1995.

[Bloomenthal 97] J. Bloomenthal. Bulge Elimination in Convolution Surfaces. *Computer Graphics Forum* 16(1): 31–41 (1997).

[Bloomenthal et al. 97] J. Bloomenthal, C. Bajaj, J. Blinn, M. Cani, A. Rockwood, B. Wyvill and G. Wyvill. *An Introduction to Implicit Surfaces*. Los Altos, CA: Morgan Kaufmann Publishers, 1997.

[Bloomenthal, Shoemake 91] J. Bloomenthal and K. Shoemake. Convolution Surface. *Computer Graphics (Proc. SIGGRAPH 91)* 25(4): 251–256 (1991).

[Bloomenthal, Wyvill 90] J. Bloomenthal and B. Wyvill. Interactive Techniques for Implicit Modeling. *Computer Graphics* 24(2): 109–116 (1990).

[Cani, Desbrun 97] M. Cani and M. Desbrun. Animation of Deformable Models Using Implicit Surfaces. IEEE Transactions on Visualization and Computer Graphics 3(1): 39–50 (1997).

[Dobashi et al. 00] Y. Dobashi, K Kaneda, H.Yamashita, T. Okita and T. Nishita. A Simple, Efficient Method for Realistic Animation of Clouds. In *Proc. of SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series*, edited by Kurt Akeley, pp.19–28, Reading, MA: Addison Wesley, 2000.

[Farin 97] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Fourth edition. San Diego, CA: Academic Press, 1997.

[Grimm 99] C. Grimm. Implicit Generalized Cylinders using Profile Curves. In *Proc. of Implicit Surfaces 99*, pp.33–41. City : Publisher, 1999.

[Jin et al. 00] X. Jin, Y. Li and Q. Peng. General Constrained Deformations Based on Generalized Metaballs. *Computers & Graphics* 24(2): 219–231 (2000).

[McCormack, Sherstyuk 98] J. McCormack and A. Sherstyuk. Creating and Rendering Convolution Surfaces. *Computer Graphics Forum* 17(2): 113–120 (1998).

[Nishimura et al. 85] H. Nishimura, M. Hirai and T. Kawai. Object Modeling by Distribution Function and a Method of Image Generation. Transactions on IECE 68-D(4): 718725 (1985).

[Nishita et al. 97] T. Nishita, H. Iwasaki, Y. Dobashi and E. Nakamae. A Modeling and Rendering Method for Snow by Using Metaballs. *Computer Graphics Forum* 16(3): 357–364 (1997).

[Prusinkiewicz, Lindenmayer 90] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. New York: Springer-Verlag, 1990.

[Sherstyuk 99] A. Sherstyuk. Kernel Functions in Convolution Surfaces: A Comparative Analysis. *The Visual Computer* 15(4): 171–182 (1999).

[Wyvill et al. 86] G. Wyvill, C. McPheeters and B. Wyvill. Data Structure for Soft Objects. *The Visual Computer* 2(4): 227–234 (1986).

[Wyvill, Wyvill 89] B. Wyvill and G. Wyvill. Field Functions for Implicit Surfaces. *The Visual Computer* 5(1/2): 75–82 (1989).

Web Information:
The C source code of the field computation and images are available online at http://www.acm.org/jgt/papers/Jinetal01

Xiaogang Jin, State Key Laboratory of CAD &CG, Zhejiang University, Hangzhou, 310027, P.R. China. (jin@cad.zju.edu.cn)

Chiew-Lan Tai, Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. http://www.cs.ust.hk/ taicl/ (taicl@cs.ust.hk)

Jieqing Feng, State Key Laboratory of CAD &CG, Zhejiang University, Hangzhou, 310027, P.R. China. (jqfeng@cad.zju.edu.cn)

Qunsheng Peng, State Key Laboratory of CAD &CG, Zhejiang University, Hangzhou, 310027, P.R. China. (peng@cad.zju.edu.cn)

Received March 13, 2001; accepted in revised form November 7, 2001.