

Convolution Surfaces

Jules Bloomenthal
Ken Shoemake

Xerox Palo Alto Research Center
Palo Alto, California 94304

Abstract

Smoothly blended articulated models are often difficult to construct using current techniques. Our solution in this paper is to extend the surfaces introduced by Blinn [Blinn 1982] by using three-dimensional convolution with skeletons composed of polygons or curves. The resulting convolution surfaces permit fluid topology changes, seamless part joins, and efficient implementation.

CR Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - curve, surface, solid, and object representations.

Additional Keywords and Phrases: Implicit Surface, Parametric Surface, Convolution, Solid Modeling, Blends.

INTRODUCTION

Animators seek models that flex and transform, but which are easy to position and mold. Designers often create these lively shapes by skillfully combining primitives, such as parametric surfaces (including polygons), or implicit surfaces and solids.

A parametric surface is given by a spatial position function: $\mathbf{p}(u, v) = [x(u, v), y(u, v), z(u, v)]$. In practice, the functions are splines defined by pieces of polynomials, or ratios of polynomials [Farin 1990], and are shaped by a sparse set of control points with an intuitive geometric relation to the surface. A single B-spline surface is naturally smooth, despite its piecewise construction. It is difficult, however, to create a smooth union of surfaces automatically.

An implicit surface is the zero-set of an implicit function $f(\mathbf{p}) = f(x, y, z)$. Including points for which $f(\mathbf{p})$ is positive gives a solid. In practice the most common functions used are polynomials, especially quadratics. The resulting algebraic surfaces can represent any rational polynomial parametric surface, as shown by classical algebraic geometry theory [Sederberg 1983]. The reverse is not true, however, suggesting that algebraic surfaces are more powerful than parametric surfaces. Unfortunately, quadrics are limited in shape, higher degree surface methods are in their infancy [Sederberg 1985] [Bajaj 1990], and blending surface construction [Warren 1989] seems difficult to automate.

Although algebraic surfaces show promise, in this paper we explore the advantages of implicit surfaces based on *skeletons*. Like control points for a spline surface, a simple lower dimensional object, the skeleton, resembles and controls the shape of a more complicated object. Vision research suggests that stick figure skeletons are natural abstractions for shapes

[Nevatia 1982].

In particular, we extend the approach of Blinn [Blinn 1982], Wyvill et al. [Wyvill 1986], and Nishimura et al. [Nishimura 1985], who used implicit functions defined by the summation of point potentials. The points generate spherical iso-surfaces which blend smoothly into each other when brought together; hence a point may be considered a skeleton which is fleshed out to form a body. Points, however, are not entirely satisfactory skeletons; for example, points that approximate a flat surface must be closely packed to avoid bumps.

After briefly considering an alternative generalization, we propose the use of *convolution surfaces*, and show they are a natural, powerful re-interpretation and generalization of potential surfaces. Colburn has used implicit surfaces based on convolution to round a solid model [Colburn 1990]; we use convolution with piecewise planar skeletons to generate models. Convolution surfaces incorporate the smooth blending power and easy manipulability of potential surfaces while expanding the skeletons from points to lines, polygons, planar curves and regions, and in principle, any geometric primitive. We exploit properties of convolution in general, and Gaussian convolutions in particular, to compute our surfaces efficiently.

POTENTIAL SURFACES

Blinn stepped beyond algebraic surfaces for molecular modeling by generating an exponentially decreasing field from the center of each atom and rendering the iso-potential surfaces [Blinn 1982]. That is, from a set S of atom centers an implicit function is defined at any point \mathbf{p} in space as

$$f(S, \mathbf{p}) = \sum_{\mathbf{s} \in S} \exp\left(\frac{-\|\mathbf{s} - \mathbf{p}\|^2}{2}\right).$$

The surface is given by those points \mathbf{p} satisfying $f(S, \mathbf{p}) - c = 0$, where c is the iso-potential value.

Others have preferred pieces of polynomials for the field functions [Wyvill 1986], [Nishimura 1985]. Essential features of any such function are that it decrease monotonically, and drop to a negligible value beyond a moderate radius. (Although Blinn observed that the decay need not be spherically symmetric, this possibility seems to have been largely neglected.) Thus a single point generates a spherical shell, and well-separated points generate separate spheres. As two points are brought together, their shells reach out and merge smoothly. When the points are coincident, a single larger sphere results.

Because the non-negative regions of these implicit functions define solid volumes, CSG set operations are also possible. Simple arithmetic operations on the function values suffice [Ricci 1973]; for example, $\max(f(S_1, \mathbf{p}), f(S_2, \mathbf{p}))$ gives the union of the two volumes generated by S_1 and S_2 . Negating the implicit function is also of interest, allowing us to subtract volumes.

One advantage of potential surfaces is that they blend smoothly. Another is that they are simple to edit; to alter the

surface one merely moves, adds, or deletes points. Unfortunately, flat surfaces can only be approximated.

DISTANCE SURFACES

Point skeletons can be generalized to polygonal skeletons in at least two ways: by computing the potential from only the nearest point of the polygon, or by summing the potentials from all the points. The second possibility gives convolution surfaces; the first gives distance surfaces—or offset solids in the sense of Requicha [Requicha 1983]. Distance surfaces are iso-surfaces of $\mu(S, \mathbf{p})$, the function value at a point \mathbf{p} defined by

$$\mu(S, \mathbf{p}) = \min_{\mathbf{s} \in S} \|\mathbf{s} - \mathbf{p}\|.$$

When S is a spline curve or planar polygon, μ can be computed without explicitly calculating the distance to each point \mathbf{s} of the curve or polygon [Bloomenthal 1989]. For a polygon, projecting \mathbf{p} onto the plane of S reduces the problem to one in two dimensions. If the projection lies inside S , use the distance to the plane; otherwise, use the distance to the nearest point on an edge.

As defined, μ is not suitable for blending; however we can use it to replace the distance calculation in Blinn's exponential, giving one generalization of potential surfaces, $f(S, \mathbf{p}) = \exp(\mu^2(S, \mathbf{p})/2)$. Because this is a composition of monotonic functions, one of which is decreasing, it can be written as

$$f(S, \mathbf{p}) = \max_{\mathbf{s} \in S} \exp\left(\frac{-\|\mathbf{s} - \mathbf{p}\|^2}{2}\right).$$

This function gives the union of the volumes generated by all the individual points of the collective skeleton, S . When the skeletons are not convex, the resulting distance surfaces can show creases, or curvature discontinuity, as seen in Figure 1; these are often undesirable.

Figure 1: Distance surfaces—skeletons, sum, union

The blending of primitives within a solid modeling system has received considerable study, as shown by the survey of methods in [Woodwark 1986], and the more recent [Rockwood 1989], [Sederberg 1987], and [Warren 1989]. As Warren [Warren 1989] has shown, for algebraic surfaces blends have a well-defined form involving a weighted sum of products of the defining polynomials. The simplest approach for blending distance surfaces is to sum the values from each of the skeletons. This eliminates creases but also creates bulges. For polygonal skeletons, especially, it is awkward to achieve blends without bulges. One bulge prevention method for algebraic surfaces is proposed in [Middleditch 1985]; it is expensive and complex, however, especially for more than two primitives. Furthermore, our surfaces are not algebraic.

CONVOLUTION SURFACES

We propose to have the best of both worlds: the spline and polygon generators of distance surfaces plus the well-behaved blends of potential surfaces. Although potential functions based on μ reduce to Blinn potentials when applied to a skeleton consisting of a single point, they behave differently for extended skeletons like polygons. One particular difference is instructive:

the surface from a skeleton broken into pieces is not the same as that of the unbroken skeleton. For example, two halves of a line segment produce a surface which bulges at the joint. This is because each skeleton generates a surface which is a union, using max, while the blending uses summation. If the skeleton is broken down into infinitesimal pieces, i.e., individual points, the union becomes irrelevant, and the result is a pure summation,

$$f(S, \mathbf{p}) = \sum_{\mathbf{s} \in S} \exp\left(\frac{-\|\mathbf{s} - \mathbf{p}\|^2}{2}\right).$$

or more properly, an integration,

$$f(S, \mathbf{p}) = \int_S \exp\left(\frac{-\|\mathbf{s} - \mathbf{p}\|^2}{2}\right) ds.$$

This new f is, in fact, the convolution of a spatially extended skeleton, not just a point, with a three-dimensional Gaussian filter kernel [Dudgeon 1984]. Formally, let $S(\mathbf{p})$ be the characteristic function for the skeleton (meaning $S(\mathbf{p}) = 1$ if \mathbf{p} is a point of the skeleton, otherwise 0), and let

$$h(\mathbf{p}) = \exp\left(\frac{-\|\mathbf{p}\|^2}{2}\right).$$

(For the sake of brevity, we omit a more rigorous development involving Dirac delta functions.) Then, using \star to represent convolution, we have

$$f(\mathbf{p}) = (h \star S)(\mathbf{p}) = \int_S \exp\left(\frac{-\|\mathbf{s} - \mathbf{p}\|^2}{2}\right) ds.$$

Convolution is usually considered part of the signal processing theory used to discuss and deal with aliasing in rendering [Foley 1990]; it is not commonly thought of for modeling shapes. Yet uniform B-spline curves and surfaces can be defined as the convolution of the B-spline basis functions with the control points [Farin 1990, p. 147], and robot path planning is simplified by convolving the room obstacle geometry with the robot's shape (the robot can then be treated as a point [Lengyel 1990]). Colburn has used an implicit surface based on convolution of a solid model with a Gaussian kernel as a way to round the corners of the solid [Colburn 1988, 1990].

Since we base our kernel on the potential function, when the skeletons are points convolution exactly reproduces the potential surface. For isolated convex skeletons such as triangles, rectangles, or line segments, convolution surfaces have almost the same shape as distance surfaces. Now, however, concave skeletons will also be smooth, and adjacent surfaces will blend seamlessly. Indeed, the superposition property of convolution guarantees that two abutting polygons will yield the same surface as a single more complex polygon which is their union:

$$h \star (S_1 + S_2) = (h \star S_1) + (h \star S_2).$$

This is shown diagrammatically in Figure 2.

dimensional Gaussians. We can, for example, separate the z component:

$$\begin{aligned} h(\mathbf{p}) &= \exp\left(\frac{-\|\mathbf{p}\|^2}{2}\right) = \exp\left(\frac{-(x^2+y^2+z^2)}{2}\right) \\ &= \exp\left(\frac{-(x^2+y^2)}{2}\right) \exp\left(\frac{-z^2}{2}\right). \end{aligned}$$

Thus to convolve with a polygon lying in the x - y plane, we can first perform a planar convolution, then convolve in z . Because polygons have infinitesimal depth, the z convolution is trivial. The planar convolution requires more work, but is again separable into x and y . We have reduced the spatial convolution of a polygon to

$$\begin{aligned} f(S, \mathbf{p}) &= \exp\left(\frac{-\|z_S - z_{\mathbf{p}}\|^2}{2}\right) \\ &\int_{S|_y} \exp\left(\frac{-\|y - y_{\mathbf{p}}\|^2}{2}\right) \int_{S|_x} \exp\left(\frac{-\|x - x_{\mathbf{p}}\|^2}{2}\right) dx dy. \end{aligned}$$

For a skeleton such as a line segment, the y integral collapses like z ; a point requires no integration. A Gaussian is also spherically symmetric—it looks the same in all directions—so this same kind of three axis separation can be used no matter how the skeleton is oriented in space.

This suggests a convenient approach for planar skeletons. Scan convert each polygon into its own digital image, filter the image in two directions by a Gaussian, then multiply a Gaussian function of the distance from \mathbf{p} to the plane of the image by the intensity at the point onto which it projects. Figure 4 illustrates the process. The images cache planar convolution results, and can be computed efficiently if convolution is performed during scan conversion.

Figure 2: Superposition

To construct a smooth, complex surface from simple skeletons, we need only sum their convolutions. Figure 3 illustrates results for the animation of two adjacent rectangular skeletons, with the upper one rotating; the primitives merge smoothly into a single shape. In contrast, the sum of algebraic surfaces is completely unsatisfactory, since the complexity of the surface is limited by the degree, which summing does not increase. For example, the sum of any number of quadric surfaces, say spheres, is a single quadric surface!

Figure 3: Model articulations.

IMPLEMENTATION

One motivation for using distance surfaces is that they are reasonable to compute; it is not immediately clear that the same is true for convolution surfaces. In some sense, however, convolution is less complicated than minimizing distance, and is more efficient to compute.

Because of the superposition property, we are free to partition a skeleton. Still, it is impossible to evaluate the convolution, even for a polygon, by explicitly summing the influence of each point. A Gaussian filter, however, has the special property of being separable; it can be factored into a product of lower-

Figure 4: Computing value of the three-dimensional convolution at a point in space.

In practice, we approximate a Gaussian with a cubic spline, to simplify computation and to limit kernel width. Artifacts of the scan conversion can be avoided by choice of a suitable resolution. A Gaussian kernel approximates an ideal low-pass filter, removing high frequency details of the skeletons. Hence the effective bandwidth of the Gaussian can be used to determine the sampling frequency needed to preserve accuracy in the sampled images, and can guide the choice of spatial sampling frequency for polygonization [Bloomenthal 1988], [Hall 1990], [Dobkin 1990]. A standard Gaussian passes less than 1% of the energy in frequencies higher than half a unit, so

in this case four samples per unit should suffice. Colburn discusses analogous resolution requirements for his octrees.

Although Colburn quotes compute times of days, we polygonize a surface in minutes. Colburn, however, is solving a different problem: he wants to make minimal changes to an existing solid. The solid is diced into tiny cubes before convolving; and while he uses separability, he cannot cache planar convolutions as we do. His method requires significant operator input to define patches through which to trace rays.

Our planar approach can be especially fast for animation; when a skeletal piece is used in many frames without change in shape, the planar images can be reused. Convolution surfaces are cheap in other situations as well. For a potential surface, evaluating the implicit function at a point requires calculating the distance to each nearby point, mapping each distance through a Gaussian, and summing. For a convolution surface, a swarm of co-planar points can be replaced by a single planar image, which requires only one distance calculation, one Gaussian evaluation, and interrogation of the image. The sum over points has been replaced by a planar convolution that is factored out of the inner evaluation loop and need only be calculated within a kernel width of the polygon perimeter.

Any point, line, or planar skeleton can be handled, and more general skeletons can be diced into polygons or polylines using standard techniques; the bandwidth of the Gaussian filters provides a least upper bound on the size of the pieces. As an example of the versatility of our method, Figure 5 is a convolution surface whose skeleton is a five-sided S-patch [Loop 1989].

Figure 5: Convolution surface from S-patch skeleton.

VARIATIONS

The shape of a convolution surface can be varied in (at least) five ways: by changing the iso-value, changing the shape of the skeleton, changing the skeleton "weight," changing the convolution kernel, and by spatial deformation. These can be illustrated with the two-dimensional potential function depicted as a height field in Figure 6. The usual CSG operations are still possible, so components of a model need not blend together. As noted previously, unions and intersections can be obtained by applying max and min to the component functions.

Figure 6: Convolution variations.

The iso-value defining the surface (or curves, in this two-dimensional example) is represented by a horizontal plane that intersects the mounds in a contour. Raising and lowering the plane, which is equivalent to adding a constant to the potential field, causes the plane to intersect different contours. Contours could also be determined as the intersection of some curved surface with the mounds; but again, the same effect can be had by changing the potential field.

Changes in the shape of the skeleton correspond to moving the mounds. This is the most basic design variation.

It is not necessary for $S(\mathbf{p})$ to be restricted to 0 or 1. When S is a set of points, each point can be given its own weight, and its influence will be scaled accordingly. In the illustration, this corresponds to the differences in height of the mounds. Negative weights correspond to pits rather than mounds, and offer a way to avoid unwanted blending, such as between the fingers of a hand. Decreasing the skeleton weight along a line, for example, gives a tapered shape, like a carrot. Incorporating a weight function, $w(\mathbf{s})$, in our defining function yields

$$f(S, \mathbf{p}) = \int_S w(\mathbf{s}) \exp\left(\frac{-\|\mathbf{s}-\mathbf{p}\|^2}{2}\right) d\mathbf{s}.$$

If the kernel is to remain a Gaussian, the only aspect of its shape that can change is its width. It is not necessary to convolve all parts of a skeleton with the same width Gaussian; narrow widths can be used where more detail is desired, while still blending well. This difference is illustrated by the low mounds in the figure. We speculate that Gaussians with broader widths can be used to provide models with less detail for small or distant objects.

Deformations [Barr 1984] [Sederberg 1986] can be applied to any form of surface, but convolution surfaces allow new possibilities. For example, the skeleton offers a convenient reference frame for a spatially variant function, such as a stretch perpendicular to the skeleton, breaking the symmetry of the Gaussian. The general quadric kernels Blinn used for his "Bobby Man" can also be considered deformations [Blinn 1982]. Note that deforming the skeleton produces a different effect than deforming the surface. Figure 7 illustrates a surface in which the convolution is twisted; the arm muscles in Figure 8 are stretched.

Figure 7: A twisted convolution.

SUMMARY

The use of convolution surfaces based on skeletons suggests numerous applications in design and animation. We have used them to embed organic forms, such as muscles, within other organic forms, such as an arm. A procedurally generated example is shown in Figure 8. The arm was procedurally generated according to joint angles and muscle size; a corresponding parametric shape would be difficult to generate procedurally. Figure 9 depicts a mosaic of convolved planar images.

Convolution surfaces offer a number of advantages, such as:

- The shape of the skeleton suggests the shape of the surface.
- The surfaces are smooth even if the skeletons are not.
- The topology of the surface can vary fluidly with changes in the skeleton.
- The kernel width limits the influence of skeletons, providing local control and allowing surface bounds estimation.
- The blends are well behaved.
- The implementation is simple and efficient.

One disadvantage is the loss of an analytic representation for the resulting surface.

Figure 8: Arm.

ACKNOWLEDGEMENTS

We are grateful to Paul Heckbert, Pat Hanrahan, and Michael Plass for technical advice; we especially thank Paul for a critical reading. We also thank Mark Bloomenthal, Kim Brook, Liz Chase, Richard Manuck, George Robertson, and Brian Tramontana for their help.

Figure 10: Convolution mosaic.

REFERENCES

- Bajaj, C., and Ihm, I. *Algebraic Surface Design with Hermite Interpolation*. Technical Report CSD-TR-939, Computer Sciences Dept., Purdue University, January 1990.
- Barr, A. "Global and Local Deformations of Solid Primitives." Proceedings of SIGGRAPH'84 (Minneapolis, Minnesota, July 23–27, 1984). In *Computer Graphics* **18** (3), (July 1984), 21–30.
- Blinn, J.F. "A Generalization of Algebraic Surface Drawing." *ACM Transactions on Graphics* **1** (3) (July 1982), 235–256.
- Bloomenthal, J. "Polygonization of Implicit Surfaces." *Computer Aided Geometric Design* **5** (1988), 341–355.
- Bloomenthal, J. "Techniques for Implicit Modeling." Xerox PARC Technical Report P89-00106. 1989.
- Colburn, S. "Method for Global Blending of Computer Modeled Solid Objects using a Convolution Integral." United States Patent No. 4,791,583, December 1988.
- Colburn, S. "Solid Modeling with Global Blending for Machining Dies and Patterns." SAE Technical Paper Series #900878, Society of Automotive Engineers, Inc., 1990.
- DeRose, T.D. *Geometric Continuity: a Parametrization-Independent Measure of Continuity for Computer-Aided Geometric Design*. Ph.D. dissertation, Computer Science division, University of California, Berkeley, California, August 1985.
- Dobkin, D., Levy, S., Thurston, W., and Wilks, A. "Contour Tracing by Piecewise Linear Approximations." *ACM Transactions on Graphics*, **9** (4), (October 1990), 389–423.
- Dudgeon, D. and Mersereau, R. *Multidimensional Digital Signal Processing*. Prentice Hall, 1984.

- Duff, T. "Polygon Scan Conversion by Exact Convolution." *Proceedings of the International Conference on Raster Imaging and Digital Typography* (Lausanne, Switzerland, October, 1989), 154–168.
- Farin, G. *Curves and Surfaces for Computer Aided Geometric Design, 2nd Edition*. Academic Press, 1990.
- Foley, J., van Dam, A., Feiner, S., and Hughes, J. *Computer Graphics: Principles and Practice, 2nd Edition*. Addison-Wesley, 1990.
- Hall, M., and Warren, J. "Adaptive Polygonalization of Implicitly Defined Surfaces." *IEEE Computer Graphics and Applications* **10** (6), (November 1990), 33–42.
- Hoffman, C. and Hopcroft, J. *The Potential Method for Blending Surfaces and Corners*. Technical Report TR 85-674 Computer Science Dept., Cornell University, 1985.
- Lengyel, J., Reichert, M., Donald, B.R., and Greenberg, D.P. "Real-Time Robot Motion Planning Using Rasterizing Computer Graphics Hardware." *Proceedings of SIGGRAPH'90* (Dallas, Texas, August 6–10, 1990). In *Computer Graphics* **24** (4), (August 1990), 327–335.
- Loop, C., and DeRose, T. "A Multisided Generalization of Bezier Surfaces." *ACM Transactions on Graphics* **8** (3), (July 1989), 204–234.
- Middleditch, A.E. and Sears, K.H. "Blend Surfaces for Set Theoretic Volume Modeling Systems." *Proceedings of SIGGRAPH'85* (San Francisco, California, July 22–26, 1985). In *Computer Graphics* **19** (3), (July 1985), 161–170.
- Nevatia, R. *Machine Perception*. Prentice-Hall, 1982.
- Nishimura, H., Hirai, A., Kawai, T., Kawata, T., Shirakawa, I., and Omura, K. "Object modeling by distribution function and a method of image generation." *Journal of papers given at the Electronics Communications Conference 1985*, J68-D(4), 1985 (In Japanese).
- Requicha, A.A.G. "Toward a Theory of Geometric Tolerancing." *International Journal of Robotics Research* **2** (4) (Winter 1983), 45–49.
- Ricci, A. "A Constructive Geometry for Computer Graphics." *The Computer Journal* **16** (2), (May 1973), 157–160.
- Rockwood, A.P. "The Displacement Method for Implicit Blending Surfaces in Solid Models." *ACM Transactions on Graphics* **8** (4), (October 1989), 279–297.
- Rossignac, J.R. and Requicha, A.A.G. "Constant-Radius Blending in Solid Modeling." *Computers in Mechanical Engineering* (July 1984), 65–73.
- Sederberg, T. *Implicit and Parametric Curves and Surfaces for Computer Aided Geometric Design*. Ph.D. dissertation, Mechanical Engineering, Purdue University, 1983.
- Sederberg, T. "Piecewise Algebraic Surface Patches." *Computer Aided Geometric Design*, **2**, (1985), 53–59.
- Sederberg, T., and Parry, S. "Free-Form Deformations of Solid Geometric Models." *Proceedings of SIGGRAPH'86* (Dallas, Texas, August 18–22, 1986). In *Computer Graphics* **20** (4) (August 1986), 151–160.
- Sederberg, T. "Algebraic Geometry for Surface and Solid Modeling." *Geometric Modeling: Algorithms and Trends*, G. Farin, ed., SIAM Press, 1987.
- Warren, J. "Blending Algebraic Surfaces." *ACM Transactions on Graphics* **8** (4) (October 1989), 263–278.
- Woodward, J.R. "Blends in Geometric Modelling." *The Mathematics of Surfaces II*, ed. R. Martin, 255–297.
- Wyvill, G., McPheeters, C., and Wyvill, B. "Data Structure for Soft Objects." *Visual Computer* **2** (4), (August 1986), 227–234.