Box-spline based CSG blends

Jörg Peters Michael Wittman

Purdue University

Abstract

The zero set of a trivariate spline is used to blend basic CSG surfaces of algebraic degree up to four. The resulting volume-bounded blend surface is generically curvature continuous, and piecewise of algebraic degree four itself, independent of the number of surfaces joined. The algorithm consists of two parts: representing each of the n basic surfaces within the blend volume as a trivariate (box-)spline, and combining the information from the n 3D arrays of spline coefficients into one that represents a new spline. The zero set of this new spline defines the blend surface. It is traced using approximation by subdivision.

1 Introduction

The blending of surfaces that smoothly join primary surfaces has motivated extensive research both on parametrically and implicitly defined surfaces (see e.g. the surveys [27], [26]). This paper suggests a new implicit approach based on the zero set of the average of box splines. The approach may be used to smoothly join an arbitrary number of low-degree primary surfaces and results in a blend that is itself of a fixed low algebraic degree and consists of the zero set of a fixed number of polynomial pieces. Specifically, the blend surface has the following properties:

- The blend surface is curvature continuous.
- The blend surface joins C^2 with input surfaces of algebraic degree three (and some of degree four) that are separate outside the blend volume or join C^2 outside the blend volume.
- The approximation order to general smooth surfaces is $O(h^4)$.
- The algebraic degree of the blend surface is four, independent of the number of input surfaces.
- The blend surfaces can be rendered stably and moderately fast.
- The representation is compatible with set-theoretic representations; e.g. point classification (set membership determination) is supported.
- The surface representation has volume elements associated with it.
- The blend is volume bounded.

Related literature. Starting with [8], box splines have been developed during the last decade. With the notable exception of [24], [2], [4], and [10], most results on non-tensor-product box-splines have been published in journals on approximation theory and seem to not have entered the standard repertoire of CAD. This is unfortunate since the representation combines smoothness, efficient evaluation and high approximation order [9] as summarized in the appendix.

Besides the well-known implicit blending constructions of [12], [13], [15], [14], there are currently two main approaches to defining the individual pieces of a function whose zero set represents a surface. The first is to generate an approximate triangulation of the surface and then erect a shell-like structure of trivariate polynomial pieces over this parametrization [25], [6], [11], [5], [1], [16]. The second is to define a function on a regular, global lattice, for example, a piecewise triquadratic, C^1 tensor-product spline [17]. The regular lattice has the advantage that non-rectilinear features of the surface do not require special treatment and that no parametrization other than the lattice structure is imposed. The algorithm defined below is of the second kind. It can alternatively be viewed as a systematic way of creating a field in the spirit of 'blobby objects' used in animation [28] or as approximate version of constructive geometry in the sense of Ricci [23].

The paper is structured as follows. Section 2 explains the basic idea, discrete blending in coefficient space, the

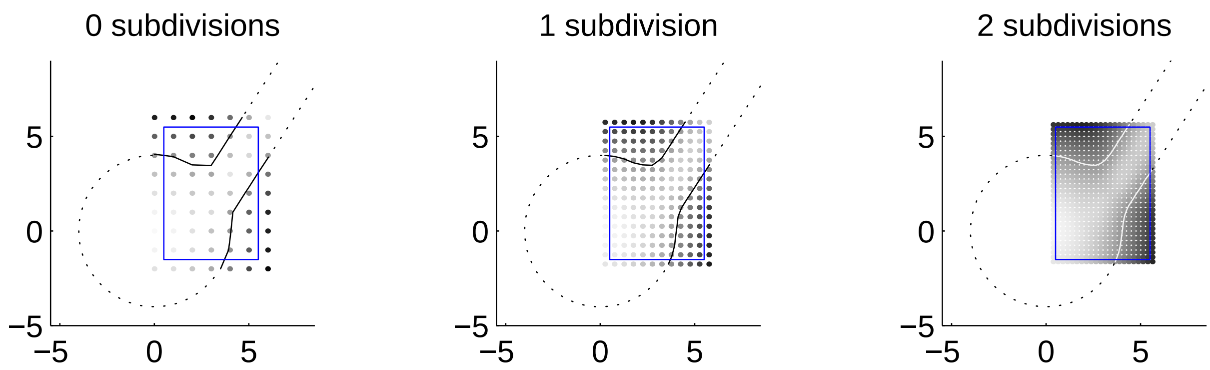

Figure 1: The grid of dots represents the array of coefficients that weigh shifts of the positive, symmetric function depicted in Figure 12(*bottom*). Darker dots correspond to negative array entries. The array is initialized as the maximum of two arrays containing the coefficients of a spline whose zero set is a circular disk, and another array that represents the intersection of two linearly bounded half spaces: the result is a C^1 quadratic blend of $-x^2-y^2+16 >= 0$ union $(-x+.65y+3.5 >= 0$ intersect $x-.65y-.75 >= 0)$. (cf. [19]).

third section gives a formal statement of the algorithm, Section 4 gives the details of the 3D construction and Section 5 discusses the properties of the representation and gives examples. A review of box spline properties used in this paper is appended.

2 Discrete blending in coefficient space

We start with an intentionally simple example in two rather than three dimensions. Consider two arrays with ± 1 entries:

$$M_1 := \begin{matrix} +1 & +1 \\ -1 & -1 \end{matrix} \qquad \text{and} \qquad M_2 := \begin{matrix} +1 & -1 \\ +1 & -1 \end{matrix}$$

and a third array generated as the componentwise maximum

$$M_3 := \max\{M_1, M_2\} := \begin{matrix} +1 & +1 \\ +1 & -1 \end{matrix}$$

We interpret each entry $M_k(i,j), k \in \{1,2,3\}, i,j \in \{1,2\}$ as the coefficient of a positive, symmetric function of compact support in two variables centered at (i,j), say the bell-shaped function displayed in Figure 12,*bottom* so that the array M_k represents the superposition of the four functions. We can visualize the zero set associated with M_1 as a horizontal line segment, the zero set of M_2 as a vertical line segment and the zero set of M_3 as a right angle, that has to be smoothed out since the four function shifts are each smooth and regular and hence so is their zero set. Thus, if we interpret the positive side of each line segment as the interior of some two dimensional object, then M_3 represents a smoothed-out union of the two objects. Correspondingly, we call this approach *discrete blending in coefficient space*. A number of detailed examples of this blending in two dimensions have been worked out in [19]. For example, in Figure 1,*left*, the grid of dots represents the array of coefficients weighing shifts of the positive, symmetric function depicted in Figure 12,*bottom*. The shading is proportional to the value of the array-entry, the darker the more negative, with black denoting "outside". The array is initialized as the maximum of two arrays containing the coefficients of a spline whose zero set is a circular disk, and another array that represents the intersection of two linearly bounded half spaces. The details of the initialization of this two-dimensional example are given in [19], Section 3. The general algorithm for the initialization is explained in the next two sections.

A useful property of surfaces defined as shifts of a box-spline is that the surface is approximated well by the piecewise linear interpolant to the coefficients in the array, the better the less the local variation between the coefficients (see also Appendix 7.3). This is the basis for fast algorithms for graphic display and rendering because a simple averaging rule, called *subdivision*, increases the number of coefficients and decreases the variation between them quickly, so that the function values are well approximated by the coefficients after only a few steps of subdivision. Correspondingly, the zero set can be well approximated by the sign change in the coefficient array. This is illustrated by the sequence in Figure 1 where, from left to right, the zero level set is approximated at three consecutive subdivision steps and the white curve inside the rectangle of the second subdivision is a piecewise linear approximation to the quadratic blend curve. Note also that the ever denser set of coefficients shrinks towards a two dimensional ver-

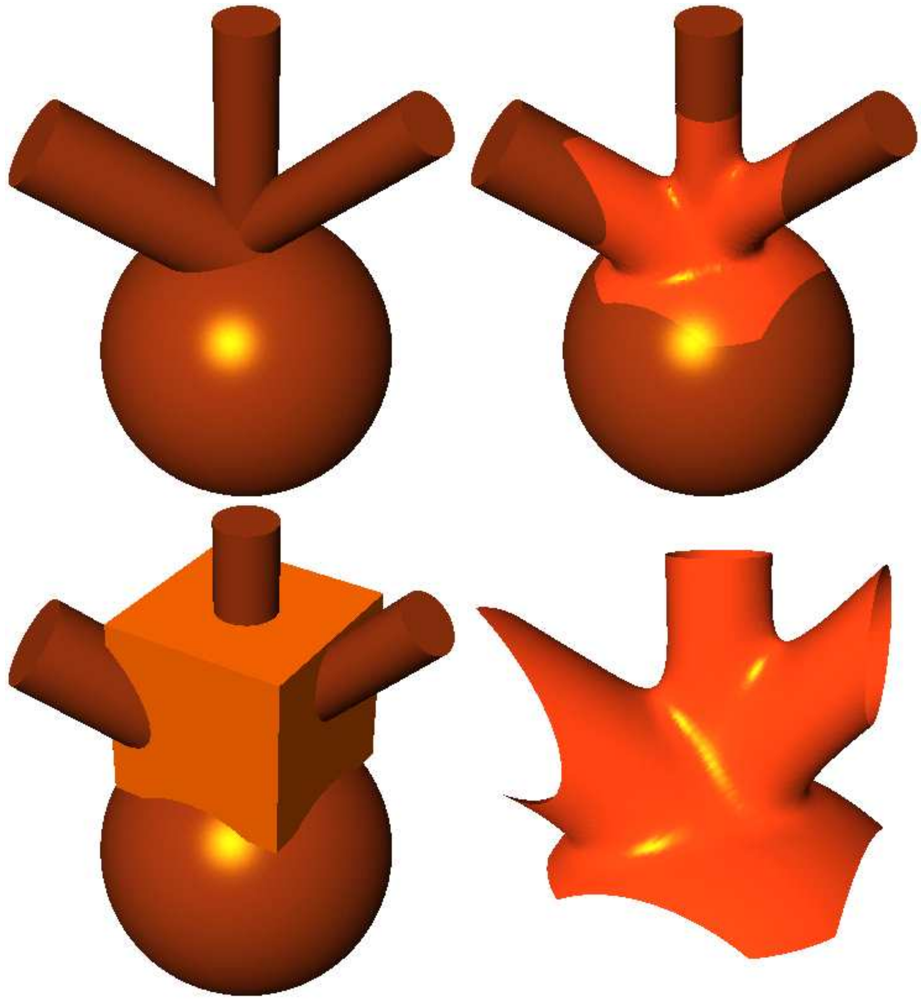

Figure 2: (*Upper left*) Primary surfaces, (*lower left*) blend volume in place, (*lower right*) zero set of the box-spline, (*upper right*) blended ensemble.

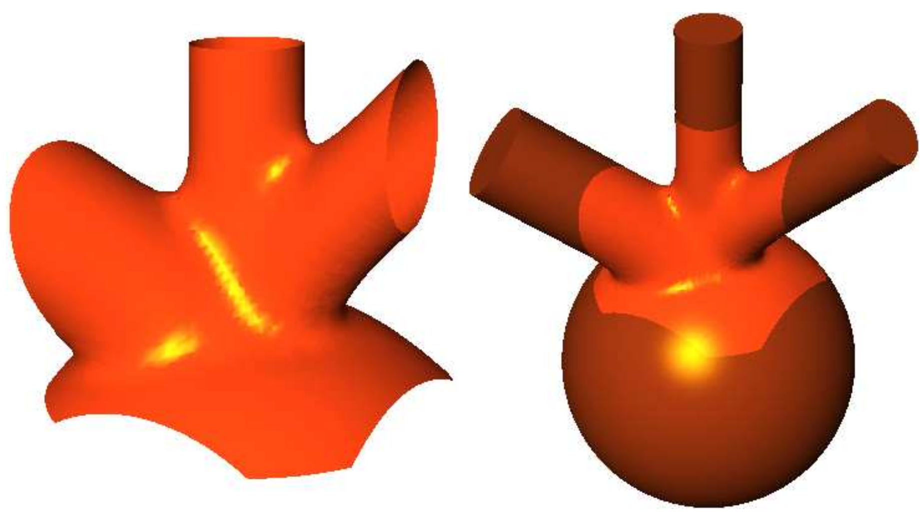

Figure 3: Termination of the blend with the left cylinder altered by subtracting a half space.

sion of the blend volume.

We are now ready for a complete example in three dimensions illustrated by Figure 2. On the left, we see the primary surfaces, and below, a blend volume, here a simple brick shape, that covers the volume to be blended. The blend volume serves as spline domain. The zero set of this spline is shown on the right, both in place as blending surface and enlarged, by itself. The blend surface is rendered just like the white blend curve in the two-dimensional example. The next section is devoted to explaining the details of the algorithm.

3 The algorithm

The *input* to the algorithm are

a. the n defining polynomials $p_i(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^3$, of the primary surfaces.

b. a blend volume and its partition (default 5 by 5 by 5),

c. a map that combines n array entries to one, and scaling factors (default 1), and

d. a number of averaging steps (default 0).

For example, the defining polynomial of the unit sphere $p_1 \geq 0$ is $p_1(\mathbf{x}) = p_1(x,y,z) = -x^2 - y^2 - z^2 + 1$; that is, the interior corresponds to positive values of the defining polynomial. The *blend volume* is a typically block-shaped region in 3-space that encloses the blend surface to be constructed. The blend volume need not be aligned with the global xyz coordinate axes and can be the image of a cube under a more general map. An example of an operation is to choose the maximum of a set of coefficients. This will correspond to an approximate union or blend, while a minimum operation approximates an intersection.

The three *steps of the algorithm* are as follows. First, each of the n basic primitives within the blend volume is automatically expanded as a trivariate spline. That is, the defining polynomial of each basic primitive, is written as a linear combination of shifts of the spline basis function M. For example, the first defining polynomial is $p_1(\mathbf{x}) = \sum a^1(\alpha)M(\mathbf{x}-\alpha)$, where $\alpha = (\alpha_x, \alpha_y, \alpha_z) \in \mathbb{R}^3$ is a point on a lattice that sub-divides the blend volume. The explicit formulas for the constants $a^1(\alpha) \in \mathbb{R}$ in terms of the coefficients of the defining polynomial are derived in Section 4. The zero set of the spline then defines the surface. Second, the operation specified generates from the n 3D arrays of spline coefficients one entry in each position and hence a new 3D array and associated spline, $p(\mathbf{x}) = \sum a(\alpha)M(\mathbf{x}-\alpha)$. Typically, when blending, $a(\alpha)$ is defined as $a(\alpha) := \max_\ell\{a^\ell(\alpha)\}$ except for α in the boundary set. The *boundary set* of the blend consists of α within a certain range of a sign change on a face of one of the arrays where all other arrays have negative entries. A face of the array is the 2 dimensional subarray where one of the three indices is either maximal or minimal. Thus where a single input surface meets the blend volume, its coefficients are preferred. This preference also holds during the optional third and final step, local averaging that allows to smoothen features, and thus guarantees a smooth transition between primary surface and blend surface. When surfaces are added to the blend ensemble, blends on blends are avoided by regenerating the blend based on all primitives.

4 Blend Surfaces

According to the first step of the algorithm, we select a suitable box-spline in three variables. A poor choice of box-spline can result in artifacts as shown in Figure 4. We choose for M the centered 7-direction quartic box spline which is a serendipitous element among the trivariate box-splines in that it combines a high degree of symmetry, with high reproduction of polynomials, almost maximal smoothness and low degree. The specific properties, degree four, curvature continuity and reproduction of all cubics and some quartics, are derived in the appendix.

The first step requires representing the defining polynomial p_ℓ of each primary surface in terms of shifts of M:

$$p_\ell(\mathbf{x}) = \sum a^\ell(\alpha)M(\mathbf{x}-\alpha).$$

Since each p_ℓ decomposes into monomials $x^i y^j z^k$, namely $p_\ell = \sum c^\ell_{ijk} x^i y^j z^k$, it suffices to determine scalar coefficients $a_{ijk}(\alpha)$ such that the following (Marsden) identity holds

$$x^i y^j z^k = \sum a_{ijk}(\alpha)M(\mathbf{x}-\alpha).$$

Then

$$a^\ell(\alpha) := \sum_{i+j+k} c^\ell_{ijk} a_{ijk}(\alpha).$$

For the centered 7-direction box-spline, the scalar coefficients for $i+j+k < 3$ are particularly simple:

$$a_{ijk}(\alpha) = \alpha_x^i \alpha_y^j \alpha_z^k - \begin{cases} 5/12 \text{ if } i=2 \text{ or } j=2 \text{ or } k=2 \\ 0 \text{ else} \end{cases}.$$

For example, to represent the cylinder

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

the array entry $(\alpha_x, \alpha_y, \alpha_z) \in \mathbb{N}^3$ for a unit partitioned blend volume at the origin is

$$(\alpha_x^2 - \frac{5}{12}) - 2\alpha_x\alpha_y + (\alpha_y^2 - \frac{5}{12}) + (\alpha_z^2 - \frac{5}{12}) - 1.$$

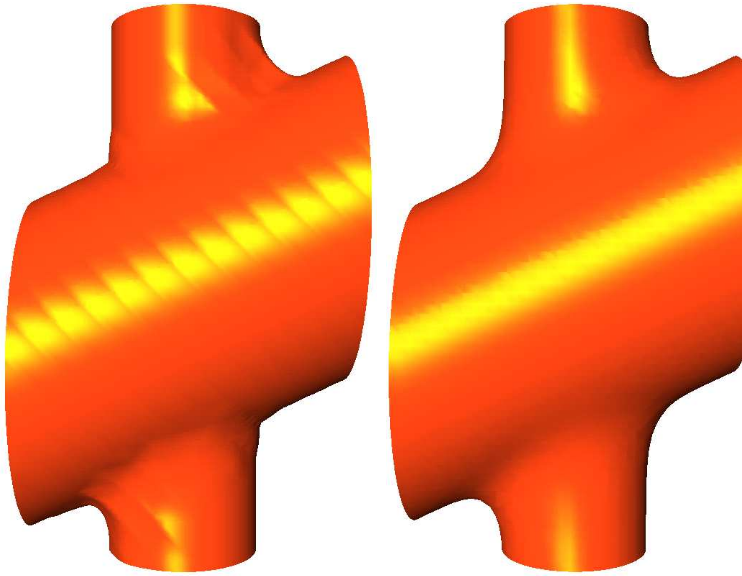

Figure 4: Artifacts on a blend surface generated as the zero set of the 5-direction box spline (*left*) with the unsymmetric direction matrix $\Xi := \begin{bmatrix} 1&0&0&-1&-1 \\ 0&1&0&1&-1 \\ 0&0&1&-1&1 \end{bmatrix}$ The zero set of this box-spline and hence the blend consists of quadrics. The zero set of the 7-direction box spline is shown (*right*).

To map the integer indices of the array onto the partition of the blend volume in the general case let A be the transformation that maps a box of size $n_1 \times n_2 \times n_3$ at the origin to the blend volume. Then the 3D array V is initialized as

$$V(\beta) := a(\alpha) = a(A(S_M \beta)),$$
$$\beta \in [0..n_1] \times [0..n_2] \times [0..n_3].$$

Here S_M is a shift by 1.5 followed by scaling by $(n_i + 1)/n_i$ in the ith component so that evaluation by subdivision converges exactly to the blend volume.

For additional smoothing, an optional averaging step may be added. None of the figures displayed in this paper required this extra smoothing. With the boundary set kept fixed and positive and negative coefficients are averaged separately by a Gaussian. The averaging is not expensive since it involves at most the $n_1 \times n_2 \times n_3$ interior coefficients of the array V. The boundary set of the 7-direction box-spline blend consists of the indices within an index range of 2 from sign changes on the array boundary.

5 Discussion of the blend surfaces

5.1 Smoothness and reproduction

Within the blend volume the blend surface is the zero set of a C^2 function. If this function is regular within the blend volume then, by the implicit function theorem, its zero set is also C^2. Using singular defining polynomials, it is also possible to represent singularities such as the apex of a cone. A C^2 join with the input surface across the boundary of the blend volume is guaranteed if the blend surface matches the input surface up to second order across the boundary. This is in particular the case if the boundary sets do not overlap and the defining polynomial is of degree less than four or equal to four with no fourth order mixed monomials. *Cyclides*, whose defining polynomial is $p(x,y,z) = (x^2+y^2+z^2-m^2+b^2)^2 - 4(ax-cm)^2 - 4b^2y^2$ (see e.g.[22]), have a combination of mixed terms $x^2y^2 + x^2z^2 + y^2z^2$ that is not reproduced exactly. Yet, the box-spline approximation is visually indistinguishable from the correct zero set (c.f. Figure 5) traced out using the explicit parametrization. This may be explained by the $O(h^4)$ approximation order of the box spline. If we choose an (n,n,n)-partition of the blend volume, then the error in the mixed terms is of the order n^{-4} with respect to the size of the bounding box. Moreover, for the smooth blend, global reproduction is not necessary. Local reproduction of the first three Taylor terms across a plane (or other blend volume boundary) suffices. For example, if we choose the blend volume boundary to coincide with a line of curvature of the cyclide then only a quadratic has to be reproduced for continuity and a cone for tangent continuity.

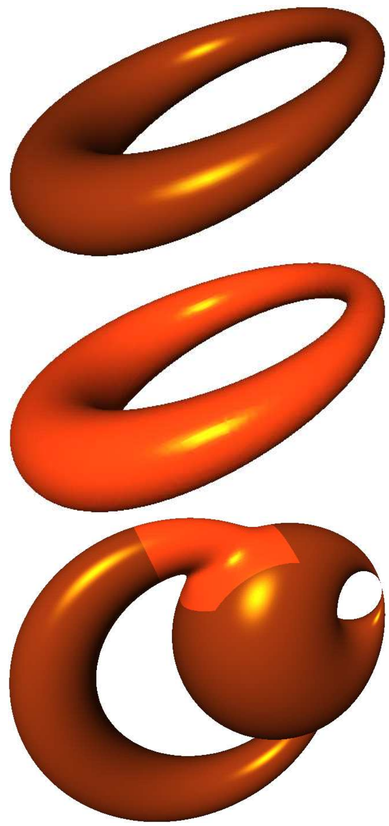

Figure 5: Exact cyclide, approximate cyclide as zero set of box-splines and two blended cyclides.

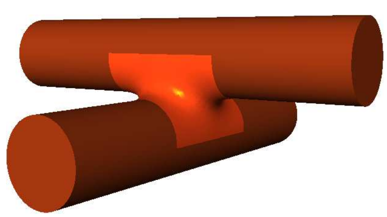

Figure 6: Dominance of the interior can join two non-intersecting primary surfaces inside the blend volume if they are sufficiently close. To keep the surfaces separate, a finer partition of the blend volume suffices, or, of course, removal of the blend volume.

5.2 Zero sheets, rendering, evaluation and point-classification

In the approximate union or blend operation represented by the the maximum operation in Step 2 of the algorithm, positive entries dominate. That is, if any of the entries in a one of the 3D arrays is positive, the resulting entry will be positive. Conversely, a negative entry will only appear in the resulting array, if all contributing entries are negative. Thus interior volume dominates and hence zero sheets can disappear but no more sheets can appear than were present in the original n primary surfaces. At worst, as Figure 6 illustrates, separate zero sheets can join if they are sufficiently close within the blend volume. The analogous argument holds for smoothed intersections – here the intersection is smoothed by removing volume in the interior of some but not all primary surfaces.

To extract a continuous piecewise linear approximation of the zero sheet of the spline, we traverse the coefficient array to detect sign changes. In standard fashion, here additionally motivated by the tetrahedral support of the polynomial pieces, each cube with a sign change is split into tetrahedra according to Figure 10, associating the average of the values at the vertices with the center of each cube and cube face. For each edge whose endpoints have an opposite sign, we mark the midpoint. Each tetrahedron has either zero, three or four marked edge-midpoints. Correspondingly, we add no, one or two (coplanar) triangles connecting the midpoints to a list of triangles. The union of the triangles in the list then form the surface approximation.

The surface approximation is refined by averaging the array entries according to the subdivision rules of the 7-direction box spline (cf. the Appendix). That is, each value is replicated over a cube of half the edge length and then the values on this refined lattice are averaged consecutively in each of the four diagonal directions of the box spline. With each step of the subdivision, the approximation gains two additional digits of accuracy.

To test for intersection, we subdivide depth-first to obtain a nested sequence of bounding boxes. If a single point is to be classified, the check is first against the blend volume then against the subcubes and tetrahedra already generated. If this check fails because the point is very close to the boundary, the spline is exactly evaluated. A stable evaluation algorithm and code for the box-spline basis functions are given in [7].

5.3 Shape and detail control

Since the spline is the limit of the coefficients generated by the subdivision process, and subdivision reduces variation through averaging, its rough shape and hence

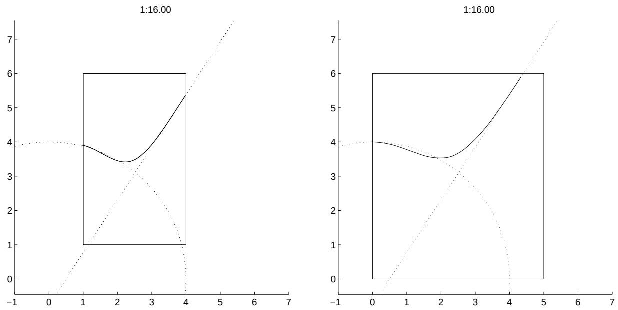

Figure 7: Blends under variation of blend volume size.

the shape of its zero set can be inferred from the 3D array of coefficients (cf. the preceding discussion of zero sheets). The blend can be modified by scaling the array-entries other than the boundary sets, e.g. by local averaging, by changing the scale λ, or by changing the blend volume(cf. Figure 7).

5.4 Compatibility with existing shape description techniques.

Combining the blend volume-approach with a set-theoretic modeling environment is straightforward: subtract the volume covered by the blend volume and add the blend volume. The 3D examples consist of clipped parametric or implicit primitives outside the blend volume while the data inside are specified by a Boolean expression over the coefficient arrays.

6 Conclusion

Smooth surfaces as the zero set of a piecewise polynomial function on a fixed-grid are capable of modeling free-form objects (see Figure 8 and also [17], [18]). However, this representation is clearly less efficient for many tasks than parametric spline-based surfacing. Our implementation is therefore specific to local blending.

The general approach offers a number of degrees of freedom that may still be explored: apart from the boundary set, the 3D array defining the blend surface can be initialized independently of the primary surfaces. In particular, non-constant scaling may be applied.

Acknowledgement: The work was supported by NSF National Young Investigator grant 9457806-CCR and a Research Experience for Undergraduates supplement.

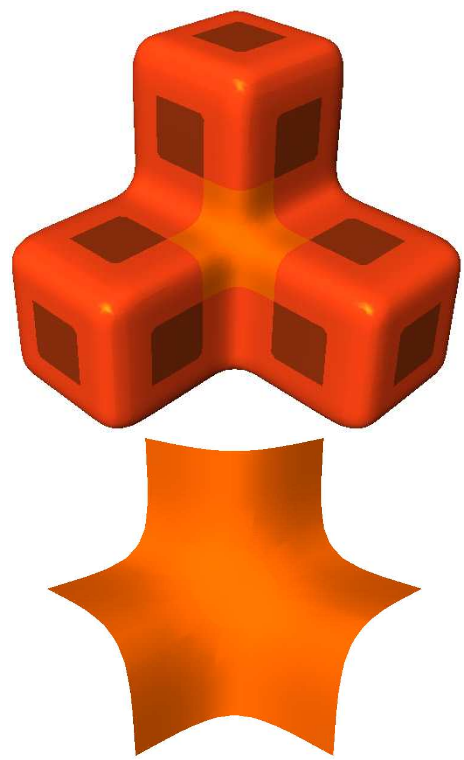

Figure 8: A curvature continuous higher-order saddle point (monkey saddle) modeled with surfaces of algebraic degree 4.

References

[1] Bajaj, C., Chen, J., and Xu, G. Free-form surface design with A-patches. In *Graphics Interface*

'94 (1994), Canadian Man-Computer Communications Society.

[2] Boehm, W. Subdividing multivariate splines. *Computer Aided Design 15* (1983), 345–352.

[3] Boor, C. D., Höllig, K., and Riemenschneider, S. *Box splines.* Springer Verlag, 1994.

[4] Dahmen, and Micchelli, C. Subdivision algorithms for the generation of box-spline surfaces. *Computer Aided Geometric Design 1* (1994), 115–129.

[5] Dahmen, and Thamm-Schaar, T.-M. Cubicoids: modeling and visualization. *Computer Aided Geometric Design 10*, 93 (1993), 93–108.

[6] Dahmen, W. Smooth piecewise quadratic surfaces. In *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche and L. Schumaker, Eds. Academic Press, Boston, 1989, pp. 181–193.

[7] de Boor, C. On the evaluation of box splines. *Numer.Algorithms 5* (1993), 5–23.

[8] de Boor, C., and DeVore, R. Approximation by smooth multivariate splines. *Trans Amer Math Soc 276* (1983), 775–788.

[9] de Boor, C., and Höllig, K. B-splines from parallelepipeds. *J. Analyse Math. 42* (1982), 99–115.

[10] Goodman, T., and Ong, B. Calculating areas of box spline surfaces. *Computer Aided Design 27*, 6 (1995), 479–486.

[11] Guo, B. *Modeling arbitrary smooth objects with algebraic surfaces.* PhD thesis, Computer Science, Cornell University, 1991.

[12] Hoffmann, C., and Hopcroft, J. The potential method for blending surfaces and corners. In *Geometric Modeling, Algorithms and new trends*, G. Farin, Ed. SIAM, 1987, pp. 347–365.

[13] Holmström, I. Piecewise quadric blending of implicitly defined surfaces. *Computer Aided Geometric Design 4* (1987), 171–190.

[14] Kosters, M. Quadratic blending surfaces for complex corners. *The Visual Computer*, 5 (1989), 134–146.

[15] Li, J., Hoschek, J., and Hartmann, E. G^{n-1}-functional splines for interpolation and approximation of curves, surfaces and solids. *Computer Aided Geometric Design 7* (1990), 209–220.

[16] Middleditch, and Dimas. Solid models with piecewise algebraic free-form faces. preprint.

[17] Moore, D., and Warren, J. Approximation of dense scattered data using algebraic surfaces. In *Proceedings of the 24th Hawaii Intl. Conference on Computer Systems Sciences, Kaui, Hawaii* (1991), pp. 681–690.

[18] Peters, J. C^2 surfaces built from zero sets of the 7-direction box spline. In *Mathematical Methods in Curve and Surface Design VI* (1996), G. Mullineux, Ed.

[19] Peters, J., and Wittman, M. Blending basic implicit shapes using trivariate box splines. In *The Mathematics of surfaces VII* (1997), pp. xx–xx. http://www.cs.purdue.edu/people/jorg.

[20] Powell, M. Piecewise quadratic surface fitting for contour plotting. In *Software for Numerical Mathematics* (1969), D. Evans, Ed., Academic Press, pp. 253–271.

[21] Powell, M., and Sabin, M. Piecewise quadratic approximation on triangles. *ACM Trans. of Math. Software 3* (1977), 316–325.

[22] Pratt, M. Cyclides in computer aided geometric design. *Computer Aided Geometric Design 7* (1990), 221–242.

[23] Ricci, A. A constructive geometry for computer graphics. *Computer Journal 16*, 2 (1974), 157–160.

[24] Sabin, M. A. *The use of piecewise forms for the numerical representation of shape.* PhD thesis, MTA Budapest, 1973.

[25] Sederberg, T. Techniques for cubic algebraic surfaces, tutorial part ii. *IEEE Computer Graphics and applications 10*, 5 (1990), 12–21.

[26] Vida, J., Martin, R., and Varady, T. A survey of blending methods that use parametric surfaces. *Computer Aided Design 26*, 5 (1994), 341–365.

[27] Woodwark, J. Blends in geometric modeling. In *Mathematics of surfaces II*, R. R. Martin, Ed. Clarendon Press, 1987, pp. 255–297.

[28] Wyvill, G., Pheeters, C. M., and Wyvill, B. Data structure for soft objrects. *The Visual Computer 2*, 4 (1986), 227–234.

[29] Zwart, P. Multivariate splines with nondegenerate partitions. *SIAM J. of Num Analysis 10* (1973), 665–673.

Figure 9: Uniform univariate splines

7 Box splines

Box splines represent a generalization of univariate spline theory to several variables. Since box splines were introduced by de Boor and DeVore [8] a rich theory has been developed and collected in the "box spline book" [3] which serves as reference for the following exposition of these piecewise polynomial functions.

The box spline M_Ξ in s variables is defined by the $s \times n$ matrix Ξ (pronounced Xi) with columns in $R^s \backslash 0$. For the purposes of this paper we may assume that the first s columns of Ξ form the identity matrix I. This yields the following inductive definition of the box spline. If $\Xi = I$, then M_Ξ is the function that is 1 on the unit cube and 0 elsewhere:

$$M_I(x) := \begin{cases} 1, & \text{if } x \in [0..1)^s, \\ 0, & \text{else.} \end{cases}$$

This box spline is piecewise constant, has degree zero and is discontinuous. If $\Xi \cup \xi$ is any matrix formed from Ξ by the addition of the column $\xi \in R^s$, then the box spline $M_{\Xi\cup\xi}$ is given by the convolution equation

$$M_{\Xi\cup\xi}(u) = \int_0^1 M_\Xi(u - t\xi).$$

For $s = 1$ this is exactly the B-spline construction by convolution(c.f. Figure 9).

7.1 Box spline properties

The box spline has the following properties.

(i) M_Ξ is positive and its shifts sum to one: $\sum_{\alpha \in Z^s} M_\Xi(\cdot - \alpha) = 1$.

(ii) The support of M_Ξ is $\Xi[0..1)^s$, i.e. the set sum of the columns contained in Ξ.

(iii) M_Ξ is piecewise polynomial of degree $n - s$. That is, each convolution in another direction ξ increases the degree by one.

(iv) M_Ξ is $\rho - 2$ times continuously differentiable, where ρ is the minimal number of columns that need to be removed from Ξ to obtain a matrix whose columns do not span R^s.

(v) M_Ξ reproduces all polynomials of degree $m := \rho - 1$ and none of degree higher than $n - s$.

(vi) The L^p approximation-order of the spline space $S :=$ span $(M(\cdot - \alpha))$ is ρ. That is with the refinement of the lattice $x \to hx$, $h < 1$, $\text{dist}(f, \sum a(\alpha) M_\Xi((\cdot - \alpha)/h)) = O(h^\rho)$ for all sufficiently smooth f.

Thus the n columns of Ξ, which may be interpreted as directions in R^s, determine the support of the piecewise polynomial and its continuity properties. Understanding the number ρ requires an analysis of the independent submatrices of Ξ.

7.2 Box spline examples

We develop three examples relevant to this paper.

1. The well-known univariate uniform cubic B-spline has the matrix (direction set)

$$\Xi := \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 9 shows in order the characteristic function of the 1-dimensional cube and its repeated convolution in the direction 1 yielding the linear 'hat' function, the quadratic and finally the cubic B-spline. We determine the characteristic numbers as

$$s = 1, n = 4, \text{ and } \rho = 4$$

since all elements of the set have to be removed to make it nonspanning in R^s. The degree of the B-spline pieces is $n - s = 3$ and the continuity is of order $\rho - 2 = 2$ as expected. The cubic spline formed as a linear combination of B-splines is guaranteed to at least reproduce polynomials of degree 3 and none of degree higher than 3. The approximation order is 4.

2. The bivariate box spline M_Ξ based on the matrix

$$\Xi := \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

is called Zwart-Powell element [29], [20], [21]. It is displayed in Figure 12 (lower right). The characteristic numbers are

$$s = 2, n = 4, \text{ and } \rho = 3$$

and hence the element is of degree 2 and its polynomial pieces are connected C^1. Since $n - s = 2 = \rho - 1$ linear combinations of the ZP-element reproduce exactly all

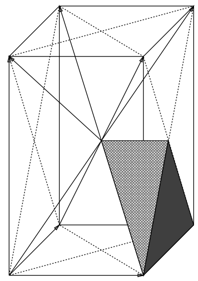

Figure 10: The 7 directions of the box spline and its domain tetrahedra

quadratic polynomials; that is any quadratic $q(x,y)$ can be written as

$$q(x,y) = \sum_{\alpha \in Z^2} a(\alpha) M_\Xi((x,y)-\alpha).$$

The Zwart-Powell element stands out among the low-degree box-splines defined over the plane, in that it has maximal smoothness equal to the degree minus one and is piecewise polynomial over a regular triangulation.

3. The 7-direction box spline is a similar serendipity element among the trivariate box splines. It is based on the direction matrix

$$\Xi := \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

The seven directions defined by the colums of the matrix cut R^3 into a symmetric regular arrangement of tetrahedra. The characteristic numbers of the 7-direction box spline are

$$s = 3, n = 7, \text{ and } \rho = 4.$$

Thus the polynomial piece defined over each tetrahedron is of degree $n-s=4$ and splines formed as a linear combination of shifts of the box spline are $C^{\rho-2} = C^2$. Elements of the spline space reproduce all cubics in three variables (and some additional polynomials of degree four) and the approximation order is 4.

7.3 Box spline subdivision

To quickly approximate any box spline we may use subdivision. Since the shifts of the box spline M_Ξ form a nonnegative, local partition of unity, a spline formed as a linear combination of shifts of the box spline is a finite convex combination of its coefficients $a(\alpha)$. To the extent that the local variation of the coefficients is small, the coefficients $a(\alpha)$ approximate the spline well. This is the basis for fast algorithms for graphic display and rendering. The key observation is that the variation of

$$\begin{array}{cc} 4a & 4b \\ & \\ 4c & 4d \end{array} \quad \rightarrow \quad \begin{array}{cccc} a & a & b & b \\ a & a & b & b \\ c & c & d & d \\ c & c & d & d \end{array}$$

$$\rightarrow_{1/2} \begin{array}{cccc} . & . & . & . \\ aa & ab & bb & . \\ ac & bc & bd & . \\ cc & cd & dd & . \end{array} \quad \rightarrow_{1/4} \begin{array}{cccc} . & . & . & . \\ . & . & . & . \\ . & aabc & abbd & . \\ . & accd & bcdd & . \end{array}$$

Figure 11: Four-direction box spline subdivision.

the coefficients is reduced when the spline is expressed in terms of box splines corresponding to the refined lattice $\frac{1}{2}Z^s$:

$$\sum_{j \in Z^s} a(j) M(x-j) = \sum_{k \in \frac{1}{2}Z^s} a_{1/2}(k) M(2(x-k)).$$

The successive computation of a sequence of refined coefficients $a_1, a_{1/2}, \ldots$ is called a **subdivision algorithm:** We compute $a_{h/2}$ from a_h for α on the finer mesh $\frac{h}{2}Z^s$. First set

$$a_{h/2}(\alpha) := \begin{cases} 2^s a_h(\alpha), \text{if } \alpha \in hZ^s \\ 0, \text{else} \end{cases}.$$

Then average in each of the directions in Ξ. That is for each $\xi \in \Xi$ compute, careful not to overwrite still needed values,

$$a_{h/2}(\alpha) \leftarrow (a_{h/2}(\alpha) + a_{h/2}(\alpha - \xi/2))/2.$$

Under mild assumptions on the matrix Ξ that are satisfied by all three box splines defined above, the sequence of control points converges quadratically to the spline [3], (30)Theorem, page 169. The sequence of array entries for the subdivision of a spline with coefficients a, b, c, d and the ZP-element as M is displayed in Figure 11.

To illustrate the effect of subdivision as approximate evaluation, we choose one box-spline coefficient (at the origin) non-zero, and all other coefficients equal zero, i.e.

$$a(\alpha) = \begin{cases} 1, \text{ if } \alpha_1 = \alpha_2 = 0 \\ 0, \text{ else} \end{cases} \qquad \alpha := (\alpha_1, \alpha_2).$$

Then

$$\sum_\alpha a(\alpha) M_\Xi((x,y)-\alpha) = M_\Xi(x,y),$$

and the spline represents just a single basis function. Figure 12 below shows four steps of subdivision on the spline coefficients. The central spike is of height 1.

In priciple, one can convert any piecewise polynomial in box-spline form into any other piecewise polynomial representation such as the power form or the Bernstein form. For example, in the Bernstein-Bézier form, the Zwart element is represented by 28 quadratic pieces with coefficients 1/2, 1/4, 1/8 and 0.

Jörg Peters, Mike Wittman
Dept of Computer Sciences
Purdue University
W-Lafayette IN 47907-1398
e-mail: jorgcs.purdue.edu

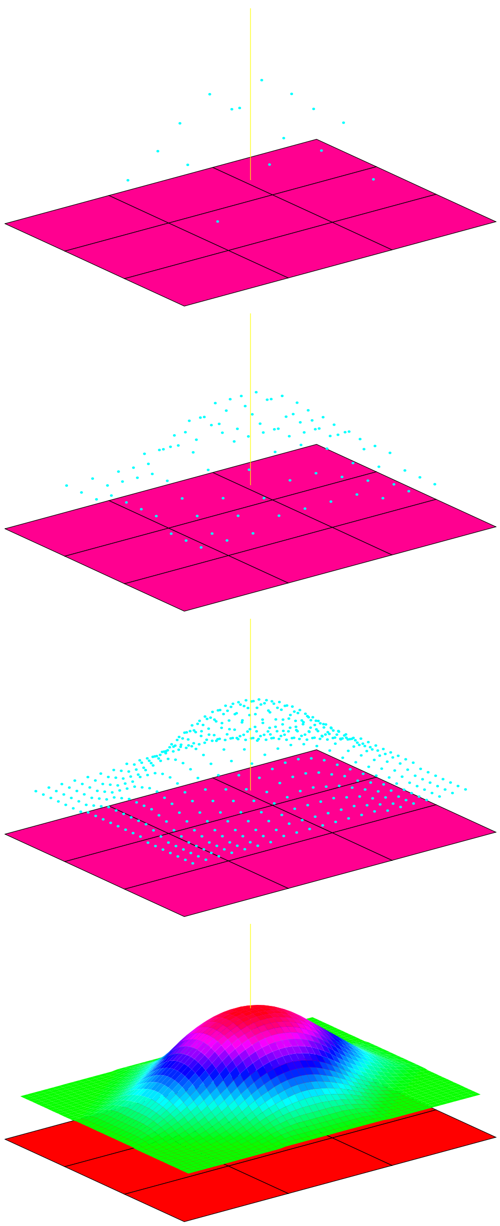

Figure 12: The Zwart-Powell element M_{Ξ} approximated using 4 steps of subdivision. The point cloud are the coefficients generated by the refinement.