Adequate Inner Bound for Geometric Modeling with Compact Field Functions

Florian Canezin[a], Gaël Guennebaud[b], Loïc Barthe[a]

[a] *IRIT - Université de Toulouse*
[b] *Inria Bordeaux*

Abstract

Recent advances in implicit surface modeling now provide highly controllable blending effects. These effects rely on the field functions of $\mathbb{R}^3 \to \mathbb{R}$ in which the implicit surfaces are defined. In these fields, there is an outside part in which blending is defined and an inside part. The implicit surface is the interface between these two parts. As recent operators often focus on blending, most efforts have been made on the outer part of field functions and little attention has been paid on the inner part. Yet, the inner fields are important as soon as difference and intersection operators are used. This makes its quality as crucial as the quality of the outside. In this paper, we analyze these shortcomings, and deduce new constraints on field functions such that differences and intersections can be seamlessly applied without introducing discontinuities or field distortions. In particular, we show how to adapt state of the art gradient-based union and blending operators to our new constraints. Our approach enables a precise control of the shape of both the inner or outer field boundaries. We also introduce a new set of asymmetric operators tailored for the modeling of fine details while preserving the integrity of the resulting fields.

Keywords: Geometric modeling, field functions, implicit surfaces, composition operators, CSG, blending, details

1. Introduction

Implicit surfaces were introduced in geometric modeling for their capability of being robustly combined in CSG trees with either sharp [1, 2] or smooth [3, 4, 5] transitions at the vicinity of the combined surface intersections. An implicit modeling system starts from field functions $f_i : \mathbb{R}^3 \to \mathbb{R}$ whose c-isosurfaces, $c \in \mathbb{R}$, define implicit surfaces S_i. Their combination through a given operator $g : \mathbb{R}^n \to \mathbb{R}$ yields a new implicit surface S_j at the c-isosurface of the field $f_j = g(f_0, ..., f_n)$. This process can be repeated recursively to model complex objects. An elegant example of this unified process is based on R-functions combining globally supported field functions in which implicit surfaces are 0-isosurfaces [6].

Throughout this process, no topological or geometrical assumption is made about the surface, and only properties of operators g and field functions f_i matter. It is thus possible to design a unified, robust and efficient modeling framework assuming the following three interdependent ingredients are met: an intuitive user interface with predictable responses, an implicit surface rendering algorithm with interactive and accurate enough feedback, and adequate equations for the field and composition functions.

The interface is often provided by sketch-based modeling systems in which a 3D shape is incrementally created by adding or removing 3D parts reconstructed from user-drawn 2D strokes [7, 8]. Interactive display is in general performed with ray-tracing or polygonalization. In both cases, field functions with compact supports are of primary interest since they localize the field function interactions when they are composed. This leads to more predictable and controllable shape behaviors and allows us to benefit from great accelerations based on the bounding box of their support [9, 10, 11]. This is why this work focuses on compact field functions.

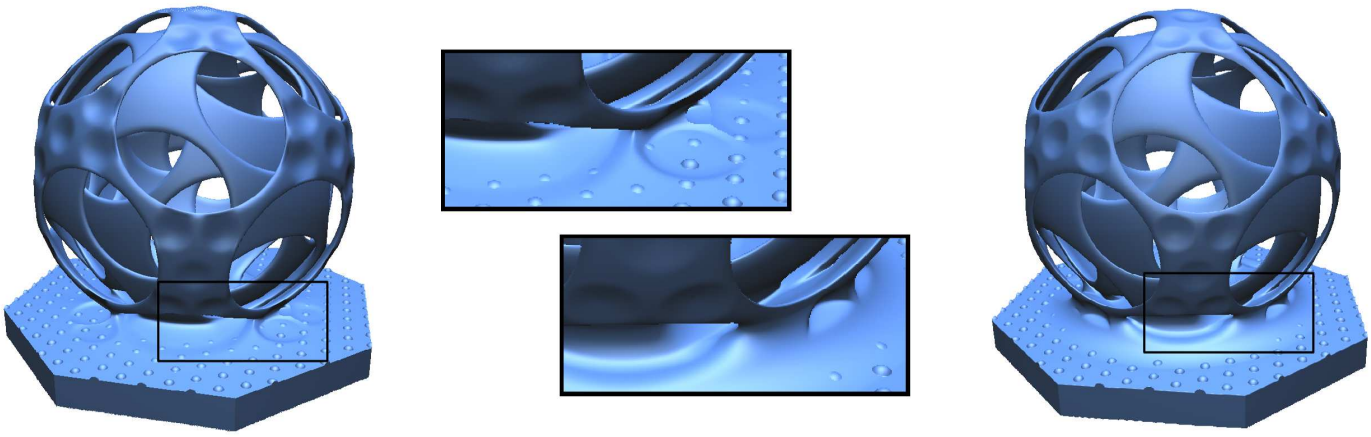
Compact field functions have been introduced by Blinn [3] using truncated Gaussian field functions, and defined with compact polynomials by Wyvill et al. [4] a few years later. By convention, a compact field function f is positive, greater than the isovalue 0.5 inside the volume delimited by the implicit surface S, and lower than 0.5 outside with decreasing values when getting further from S up to a bound. Outside this bound, the function f uniformly equals zero. Many different operators have been developed over the years, increasing their variety and improving their effectiveness [3, 12, 13, 14, 15]. While simple max-based composition functions may be enough to define n-ary Boolean operators with sharp transitions (i.e., union, intersection and difference), the most recent and advanced operators are restricted to the composition of only two fields at once [16, 17]. Such binary operators fit well most of modeling systems where object parts are incrementally combined in pairs during the modeling session. Observing that under our settings, $(1 - f_i)$ is the complement of f_i, one can easily define intersection and difference functions, $g_\cap$, $g_\setminus$, from the union function $g_\cup$ as follow:

$$g_\cap(f_1, f_2) = 1 - g_\cup(1 - f_1, 1 - f_2) \quad (1)$$

$$g_\setminus(f_1, f_2) = g_\cap(f_1, 1 - f_2) = 1 - g_\cup(1 - f_1, f_2)\,. \quad (2)$$

Email addresses: florian.canezin@irit.fr (Florian Canezin), gael.guennebaud@inria.fr (Gaël Guennebaud), loic.barthe@irit.fr (Loïc Barthe)

URL: http://www.labri.fr/perso/guenneba/ (Gaël Guennebaud), http://www.irit.fr/~Loic.Barthe/ (Loïc Barthe)



state of the art operators our operators

Figure 1: A complex model built (left) with Gourmel et al. gradient-based operators and (right) with our novel operators. As we can see in the zoom in the middle-top, details on the spheres consequently deform the blend between the spheres and the pedestal while in the middle-bottom our new operators preserve the blending shape. In addition, in the middle-top the same blending shape is smooth on the pedestal and unexpectedly sharp on the sphere, while in the middle-bottom it is nicely smooth on both objects with our operators.

We call a blend a union with smooth transitions where the combined objects intersect. If $g_\cup$ is a soft blending function then so will be $g_\cap$ and $g_\setminus$. Even though removing parts through differences is very common, these formulas enable us to treat all operators from union and blending operators only. As a consequence, since blending produces a smooth transition linking the combined objects in their *outside* volume part, little attention has been paid to definition of the part of the field function defining the *inside* of the volume. Thus only the following constraints have been considered in the design of operators on compact fields [14]:

- at the 0.5-isosurface: a sharp or smooth transition,
- at the field boundaries (0-isovalue): an adequate composition to preserve the boundaries,
- everywhere else: a smooth field composition to avoid the creation of undesired gradient discontinuities in the field function resulting from the composition.

However, as we can see in Equation 2, the difference is built from the union of $(1 - f_1)$ and f_2. Since the only requirement proposed so far for the inside part is smoothness, the field functions $(1 - f_1)$ are generally not positive, thus leading to several artifacts anytime a composition occurs in this unbounded area.

This research addresses these shortcomings. First, we propose to explicitly include the inner bound in the field definitions, thus leading to a consistent and unified compact field representation (Section 3). As for the outside, this inner bound defines the volume where the surface can be deformed by a composition operator, localizing computations in a band around the implicit surfaces. This enables enhanced and more consistent control for the creation of smooth transitions. It also increases computation optimization capabilities and lowers cost of the possible field function discrete storage [18, 19, 20, 21, 22]. This first step is performed by composing different field functions with well known step functions.

Second, we introduce new additional constraints that operators must fulfill, and show how to adapt recent union and blending operators (Section 4.1). This is a new contribution as our modified operators can be directly derived through Equations 1 and 2 to yield conforming intersection and difference operators.

Finally, we also propose a novel set of asymmetric operators tailored for the representation of small details that better preserves the shape of the outside field (Section 4.2).

Together, our novel sets of operators allow us to safely model objects composed by unions, differences and intersections, with and without smooth transitions. This is illustrated in Figure 1 where after several compositions with state of the art operators the blending shapes become uncontrollable and unsuitable while they remain controllable and as expected with our operators.

2. Related works

The simplest composition operators applying to compact field functions are the n-ary max(.) and min(.), which perform a Boolean union and intersection respectively [1, 2]. Another well known operator is the n-ary blending operator of Ricci [2]:

$$g(x_0, ..., x_n) = \left(x_0^m + ... + x_n^m\right)^{\frac{1}{m}} , \quad (3)$$

where m controls the blending size. When $m = 1$, it boils down to a simple sum, as the popular operator of Blinn [3], whose continuous extension by Bloomenthal et al. [23] yielded to convolution implicit surfaces.

While several new composition operators [24, 25, 26] and composition concepts [27, 28] are introduced for field functions with global support, operators for compactly supported fields received less attention until the last decade. Hsu et al. [13] improved the control of the blending size using adequate transfer functions modifying the slope of the composed field functions. Barthe et al. [14] defined a set of constraints for the creation

of binary composition operators on compact fields, focusing on the implicit surface and the outside bound of the field functions. Based on these constraints, a new blending operator with shape control and a *clean-union* operator (union with a smooth field everywhere except at the surface intersections) are proposed. De Groot et al. [15] used transfer functions to modify the field functions f_i so that when they are summed, the resulting n-ary operator is a blend in which the blending shape between the combined implicit surfaces is controlled pairwise.

Following the previous constraints [14], Bernhardt et al. [16] introduced locally restricted binary blending operators for compact field functions. They can interpolate between a clean-union and a blend through a parameter used to restrict the blend where the combined implicit surfaces intersect only. The blending size is automatically adapted to the size of the combined implicit surfaces.

Gourmel et al. [17] proposed a binary blending operator parametrized by the angle between the gradient of the composed field functions. This gradient-based operator automatically localizes the blend, adapts the size of the blend to the size of the combined implicit surfaces, and avoids unwanted bulging.

In all the aforementioned approaches, no specific treatment is done for the inside part of the field functions. A marginal exception is the work of Hsu et al. [13] where an inner bound is introduced to directly control the blending size when material is removed. In this work, this bound is not related to any operator conformity property. In summary, all these advanced binary operators exhibit the consistency problem and the lack of field variation control raised in Section 1.

3. Compact field function representation

Before studying the composition operators in the next section, we present our consistent representation of compact fields and its properties. As we observed in Equations 1 and 2, the complement $(1 - f)$ of a compact field function f is composed in a union or blending operator $g_\cup$ for the definition of both intersection and difference operators. Therefore, $(1 - f)$ must satisfy all the properties of compact field functions on which the definition of composition operators rely: it must be positive, greater than the isovalue 0.5 inside the volume delimited by the implicit surface, and lower than 0.5 outside with decreasing values when getting further from the surface up to a bound. Outside this bound, the function $(1 - f)$ must uniformly equal zero.

This is not the case in general, but all these properties are automatically satisfied as soon as we set an inner bound to 1 in the field of f. This manipulation is simple and as shown below the different families of field functions used in geometric modeling of 3D objects can be easily adapted to satisfy this additional requirement.

Special care has to be taken on the size of the band in which the field function varies. Indeed, outside this band, it is impossible for any composition operator to produce shapes that are not already part of the input surfaces. This band has to be large enough so that any such additional shape that a user would like to generate can be generated. In general, composition behaviors in inner and outer field parts are expected to be symmetric and a default solution is the generation of symmetric field functions. If required, the widths of the inner and outer bands can be set freely to accommodate for any specific constraint.

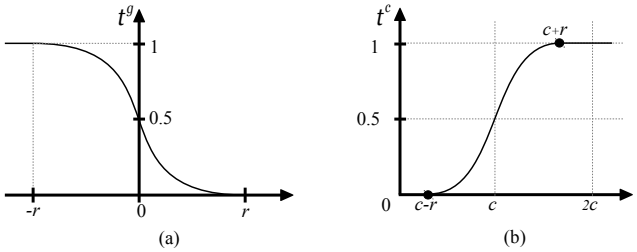


Figure 2: (a) Function t^g transferring a global support field function from $]-\infty, +\infty[$ to our compact support field representation $[0, 1]$ and (b) function t^c adjusting the bounds of a compact representation in $[0, 1]$.

Handling global distance fields. Several families of global support field functions f^g are useful for shape representation. Among them, we can cite polynomials, Radial Basis Functions [29, 30], Point Set Surfaces [31, 32, 33], volumetric diffusion [34, 35] and others. Without lack of generality, the standard convention for them is to consider the 0-isovalue as the implicit surface where the set of points $\mathbf{p} \in \mathbb{R}^3$ for which $f^g(\mathbf{p}) < 0$ defines its inner part, and the set of points for which $f^g(\mathbf{p}) > 0$ defines its outside.

In order to benefit from all compact representation advantages, the field functions f^g are composed with a transfer function t^g such that the implicit surface is not modified but the field, inside and outside, is bounded. As depicted in Figure 2 (a), we suggest defining t^g as $t^g(x) = \varphi(x/r)$ where φ is a smooth-step function, and r is the symmetric width of the band in which the resulting compact field $f = t^g \circ f^g$ varies. This width r is defined with respect to the input field metric. For the choice of φ, any smooth-step functions, such as those proposed by Li et al. [36] and Li [26], can be selected. In this paper we used the following popular C^2 polynomial function for its simplicity:

$$\varphi(x) = \begin{cases} 1 & \text{if } x \leq \text{-}1 \\ 0 & \text{if } x \geq 1 \\ -\frac{3}{16}x^5 + \frac{5}{8}x^3 - \frac{15}{16}x + 0.5 & \text{otherwise.} \end{cases} \quad (4)$$

This mapping is symmetric, but if required, individual controls on the inner and outer widths can be achieved using, for instance, Hsu's et al. step function [13]. The use of transfer functions for similar conversions is actually common. For instance, transfer functions have also been used for adapting blending operators, that were designed for compactly supported fields, to globally supported ones [37]. In this work the conventions for global field functions are preserved and only the outside part is considered for controlling the blend.

Handling non-consistent compact distance fields. As for global support fields, any compact field function f^c can be adapted to undertake our constraints by applying an adequate

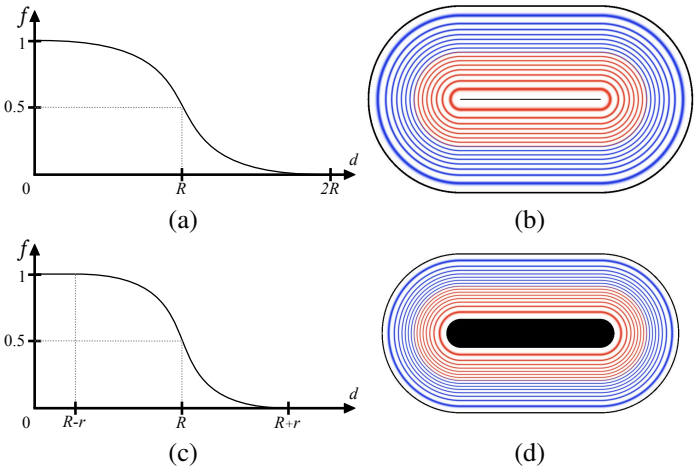


Figure 3: Compact field functions defined from the distance to a linear segment skeleton. In the first row, $r = 0$. The field is bounded in $[0, 1]$ and varies inside the volume up to the skeleton. In the second row, $0 < r < R$. The field is a narrower band around the implicit surface and there is an inner bound within which the field function uniformly equals 1. Note that the 0.5-isosurface is the same in both cases.

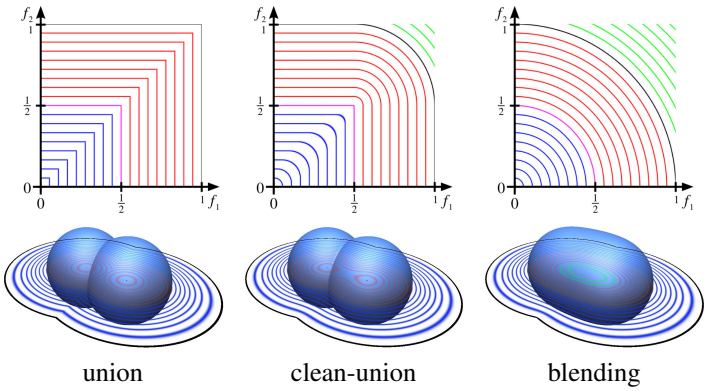


Figure 4: Illustration of three binary composition operators. The first row shows the isocurves of the 2D scalar field of each operator, with in green the values greater than 1. The second row shows their respective effect on two spherical bounded objects. The transparent surfaces are the 0.5-isosurface of the resulting fields, and their variation is illustrated in a planar section in which the lines correspond to different isovalues.

inner bound. For instance, f^c might come from skeleton-based soft-objects [4], convolution surfaces [23], or objects resulting from non conforming compositions [12, 14]. To this end, we use the transfer function:

$$t^c(x) = \varphi\left(\frac{c - x}{r}\right), \tag{5}$$

where c is the isovalue of the implicit surface in f^c, and $r \in [0, c]$ is the width of the symmetric band defined with respect to the input field metric. The shape of the resulting compact field $f = t^c \circ f^c$ is shown in Figure 2 (b). As for the transfer function t^g, t^c could easily be adapted to offer individual control on both the inner and outer bounds.

When applied on a skeleton-based soft-object, our representation results in the field function presented in Figure 3, where R is the distance from the skeleton to the surface, and r the width of the inner and outer bands. In all our field visualizations, the inner field is colored in red and the outer one is in blue. The outer bound (where $0 \leq f < \epsilon$) and the region inside the inner bound (where $1 - \epsilon < f \leq 1$) are colored in black.

4. Composition operators

Now that the field functions representing the objects to be combined are adequately defined to support the operator constructions presented in Equations 1 and 2, we address the composition operators. We focus on binary operators represented by functions $g : \mathbb{R}^2 \to \mathbb{R}$.

In order to better understand the way operators are built, it is convenient to consider $g(f_1, f_2)$ as a 2D field function and visualize its isocurves as in the first row of Figure 4 following the previous color code. Being of particular interest, the 0.5-isocurve is drawn in magenta and values greater than 1 are in green. In this representation, vertical (respectively horizontal) parts of isocurves of operator g correspond to the set of points for which $g(f_1, f_2) = f_1$ (resp. $g(f_1, f_2) = f_2$), i.e., it represents the isosurfaces of field function f_1 (resp. f_2).

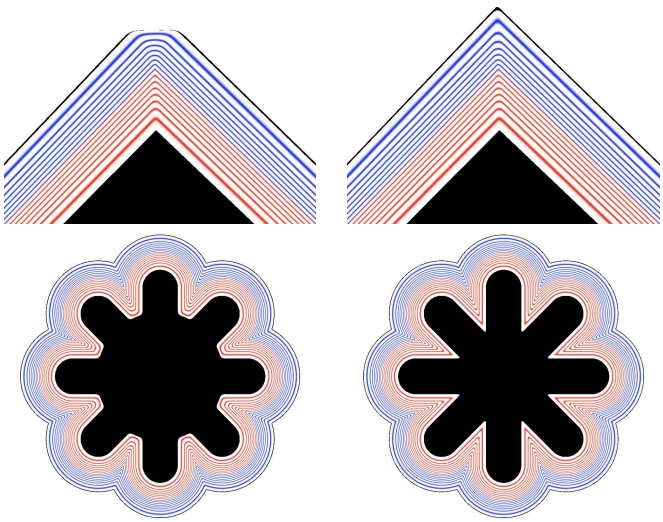
Figure 4 shows, by column and from left to right, the max operator resulting in a union of the implicit surfaces, a clean-union operator also implementing a union of the implicit surfaces but with a smooth field elsewhere, and a C^1 blending operator which smoothly links values of f_1 to those of f_2. This last operator corresponds to Ricci's function of Equation 3 with $m = 2$.

Following our boundary settings, an adequate operator g must satisfy several constraints. Firstly, as explained by Barthe et al. [14], it must guarantee the continuity of the field function $f = g(f_1, f_2)$ at the support boundaries of f_1 and f_2. This is done by imposing $g(f_1, 0) = f_1$ (along the abscissa axis) and with $g(0, f_2) = f_2$ (along the ordinate axis). In particular, $g(0, 0) = 0$. Equivalent constraints must now be added along the inner boundary. This is done by imposing $g(f_1, 1) = g(1, f_2) = 1$. In particular, $g(1, 1) = 1$. In previous works, g was only constrained to be positive on the $\mathbb{R}^+ \times \mathbb{R}^+$ domain. However, in order to keep the resulting field f consistent during composition, any operator must now remain between the inner and outer bounds (i.e., 1 and 0 respectively). More formally this means that $g(f_1, f_2) \in [0, 1]$ for all $(f_1, f_2) \in [0, 1]^2$ and g is a function $g : [0, 1]^2 \to [0, 1]$.

A naive way to enforce the above constraints is to take existing union or blending operators, and to clamp their value to the $[0, 1]$ range. However, this introduces several unwanted artifacts in the resulting field $f = g(f_1, f_2)$.

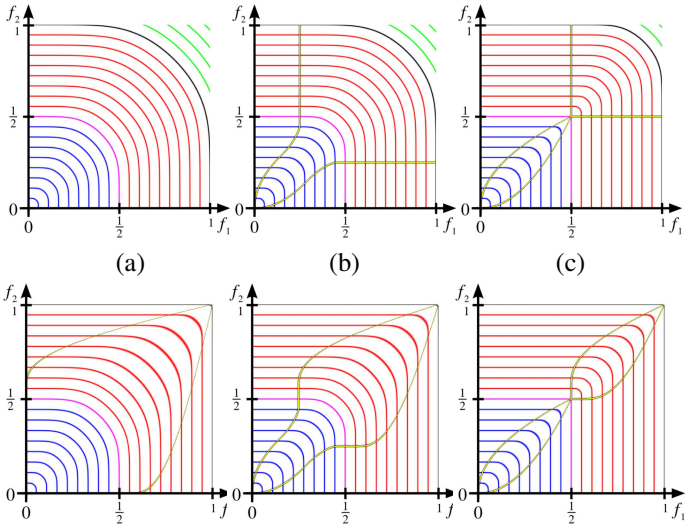
Firstly, it creates gradient discontinuities in the field f along the boundary where values are clamped, regardless of the boundary continuity of the input fields. Such discontinuities prevent the use of the resulting field with operators that exhibit a low degree of continuity along their axes. For instance, this excludes the popular sum operator that would produce at most C^0 only surfaces, as well as the circular blending operator of Figure 4 that would lack curvature continuity.

Secondly, clamping the resulting field arbitrarily truncates



(a) state of the art (b) ours

Figure 5: Intersecting 2 planes (first row) and composing 4 cylinders to build a star (second row) using a clamped version of the clean union operator (a) leads to uncontrollable bound cutting, while with our operators (b) the bounds of the planes and cylinders are preserved.



(a) (b) (c) (d) (e) (f)

Figure 6: Top-row, the state of the art gradient-based blending operators [17] produce values greater than one (green part). These operators are constructed from yellow boundary curves k_θ determining the blending radius of the operator isovalues. Three different values of θ are shown with (a) full blending, (b) intermediate blending, and (c) clean-union. Bottom-row shows our respective "closed" versions of these operators.

the field at its 1-isosurface, whereas, as explained in Section 3, the width of the inner band of a field function has to be correctly set in order to fulfill the desired modeling properties. In addition, this undesirable behavior increases with the number of overlapping compositions as shown in Figure 5.

We present two sets of operators avoiding the aforementioned problems. The adaptation of state of the art binary operators (Section 4.1) and new operators allowing us to model small details by composition without introducing field depressions (Section 4.2).

4.1. State of the art binary operators

Union and blending. Among the various operators developed for compact field functions, we present the adaptation of Gourmel et al. [17] operators. These operators are the more general and the most challenging to handle. The same modification procedure can be applied to other state of the art families of binary operators such as those of Bernhardt et al. [16] and Barthe et al. [14].

As illustrated in Figure 6 top-row, Gourmel et al. [17] propose a continuous set of C^∞ operators g_θ interpolating between a clean-union and a very smooth blend according to a parameter θ. Each operator g_θ is built from a profile curve k_θ (shown in orange) that is symmetric with respect to the $f_1 = f_2$ diagonal. Outside this profile, the operator simply returns the maximum of f_1 and f_2. Inside, a blend is realized by instancing iso-curves $\rho(\phi)$ defined in polar coordinates. Intuitively, they mimic circular arcs but with C^∞ continuity at junctions. The set of operators is thus produced by continuously varying the profile curves k_θ with respect to θ. Since this construction does not yield closed-form formulas, operators are computed numerically and baked into 3D textures parameterized in f_1, f_2, θ.

As can be seen in Figure 6, these operators, as all other existing blending operators, are not bound to 1. To overcome this issue without introducing the aforementioned problems, an effective solution consists of ensuring that along the $f_1 = 1$ and $f_2 = 1$ axes, $g_\theta(1, f_2) = g_\theta(f_1, 1) = 1$.

This is automatically achieved by designing the profile curves k_θ such that they meet at $f_1 = f_2 = 1$, thus "closing" the blending region of the operator as depicted in Figure 6 bottom-row. To this end, we modify the boundary functions k_θ as follow:

$$k_\theta(f) = \begin{cases} k_\theta^{base} & \text{if } f \leq 0.5 \\ \frac{1}{2}\left(\left(\frac{\tau(f)}{\tanh(1)} + 1\right)(2 - \tan(\theta)) + \tan(\theta)\right) & \text{otherwise} \end{cases}$$

where $\tau(f) = (\tanh \circ \tanh \circ \tan)(\pi(f - 1))$, $f \in [0, 1]$, and k_θ^{base} is the original boundary function proposed by Gourmel et al. [17]. The intuition behind the construction of these functions is the use of trigonometric and hyperbolic functions for their natural C^∞ continuity and the composition for controlling the slope of there shape.

The practical effect of our new operators is depicted in Figure 7 on a clean union, a blend and a gradient-based blend of two cylindrical primitives. Observe how the depression in the inner part (shown in green) is effectively removed. In the work by Gourmel et al., the parameter θ can be automatically adjusted from the angle between the gradients of f_1 and f_2 through a user defined controller. For instance, this allows us to localize the blending effect as with the "camel" controller that has the property to remove unwanted bulge as in Figure 7 (c). Note the additional distortions introduced by this operator in Figure 7 (c)-left that are removed by our improvements in Figure 7 (c)-right. This is particularly interesting for subsequent gradient-

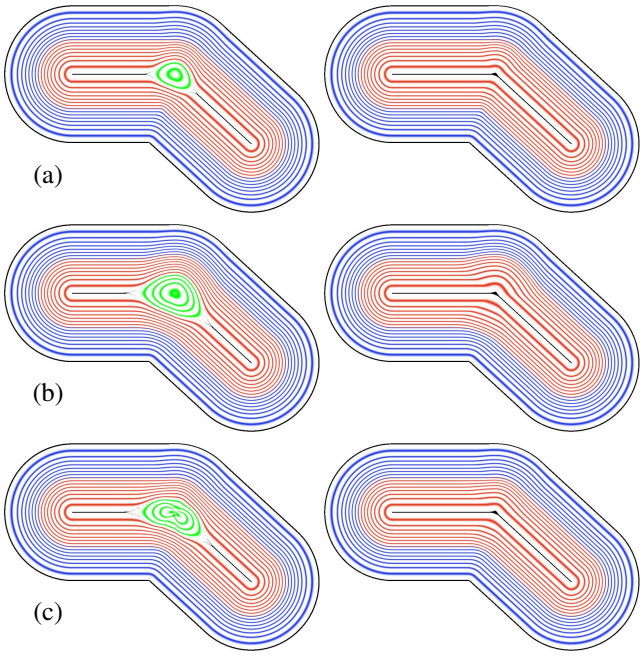


Figure 7: Closing the boundary functions of a gradient-based blending operator leads to better shaped resulting potential fields. Applying previous operators (left) and our operators (right) using (a) clean union, (b) blending and (c) "camel" blending.

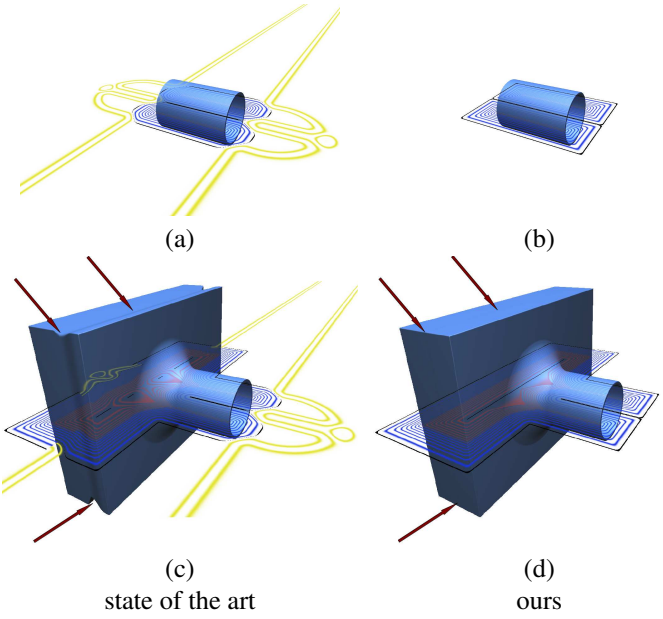


Figure 8: Top row: a hollow cylinder is built by removing a cylinder from another larger and then intersecting with 2 planes. The use of Gourmel's operators [17] on the left produces negative field values that lead to an inadequate deformation of the blended cuboid on its side, where pointed in (c) by the red arrows. This misbehavior is naturally avoided by the use of our operators as shown in (d).

based compositions in which these field distortions would introduce artifacts due to unpredictable gradient variations.

Intersection and difference. We now have all the ingredients to build artifact-free intersection and difference operators by combining our modified union and blending operators following Equations 1 and 2. The benefits of our approach are depicted in Figure 8. As we can see, the negative values (shown in yellow) produced by state of the art operators when building the cylinder in 8 (a) are avoided by our operators in 8 (b). This prevents the parallelepiped from being unexpectedly deformed by the presence of negative values when it is blended with the cylinder as in 8 (c) and to obtain the expect result in 8 (d).

4.2. Operators for details

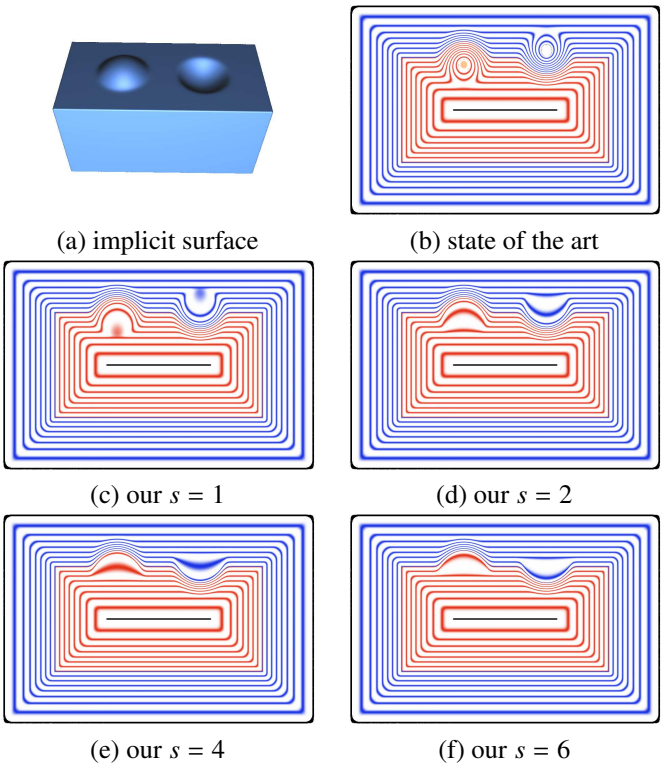
In this Section we study the application of union, blending, and difference operators when they are used to add thin details onto an existing surface. In this case, very small objects are added or removed to significantly larger ones. As shown in Figure 9 (b), classically designed operators, including the ones of the previous Section, introduce field depressions of the shape of the combined small object (a sphere in this example) where it has been added or removed. When field functions are composed, the resulting field function is expected to approximate a distance field to the implicit surface with some additional continuity and boundary constraints. The metric of this resulting field should correspond to the ones of the operands. These depressions are thus undesired from both the theoretical point of view and the practical point of view as they introduce unpredictable shape behavior when they are crossed by a blend. This is clearly illustrated in the close-up of Figure 1 (left) in which small spheres removed to deform a large one generate holes in a subsequent blend.

This leads us to the definition of a very particular operator. Indeed, when modeling small details, the resulting field is expected to progressively vary from the detailed implicit surface to the field of the large object, as shown in Figure 9 (e). Assume f_1 is the initial large scale field and f_2 is the field of the detail. This means that while the operator has to preserve the field properties of f_1 (i.e., $g(f_1, 0) = f_1$ and $g(1, f_2) = 1$), it must modify those of the field f_2 so that it is smoothly absorbed by the field of f_1. This means that in this special case, the constraints $g(0, f_2) = f_2$ and $g(f_1, 1) = 1$ do not have to be respected for $f_2 > 0.5$. To achieve this behavior, we have to build a new set of gradient-based operators $\tilde{g}_\theta$ that reproduce g_θ in the outer part of both field functions f_1 and f_2, and which progressively reproduce the field of f_1 when moving away from the 0.5 isovalue of operator g_θ.

In our representation of binary composition operators (Figure 10), this means that the operator $\tilde{g}_\theta$ must be a standard blending operator for $\tilde{g}_\theta < 0.5$. Outside this region, isocurves of $\tilde{g}_\theta$ must smoothly vary for increasing isovalues to become a straight vertical line at $f_1 = 1$. This operator is depicted in Figure 10 (a).

Our first attempt to design such an operator was to simply perform a linear interpolation between g_θ and f_1 according to a parameter $\alpha(f_1, f_2)$. However, designing α to obtain the expected operator appeared to be very challenging. Instead, we propose to precisely define the shape of each individual isocurve of $\tilde{g}_\theta$ using the following construction.

We define the isocurves of the detail operator $\tilde{g}_\theta$ by modify-



(a) implicit surface (b) state of the art

(c) our $s = 1$ (d) our $s = 2$

(e) our $s = 4$ (f) our $s = 6$

Figure 9: (a) A parallelepiped on which a sphere has been added (blending) and a sphere has been removed (difference with smooth transition). While this surface is the same for all operators, the resulting field has different local variations where the spheres are combined. This is illustrated in the planar section of the resulting field function passing by the middle of the added and removed spheres. (b) State of the art operators create field depressions. Our new detail operator (c),(d),(e),(f) absorbs the blended/subtracted field functions. As we can see, the absorption is modulated by the parameter s.

ing the ones of g_θ that are greater than 0.5 such that they become straight at an angle $\phi(\tilde{g}_\theta)$ in a local polar coordinate system as depicted by the red curve in Figure 10 (b). When $\phi = \pi/2$ we exactly reproduce g_θ, and when $\phi = 0$ we obtain the straight vertical isocurves reproducing f_1. Then, ϕ smoothly varies between $\pi/2$ and 0 when $\tilde{g}_\theta$ varies from 0.5 to 1 as:

$$\phi(t) = \frac{\pi}{2}(2-2t)^s \quad . \tag{6}$$

Here the exponent s adjusts the interpolation speed between the 0.5-isovalue and the field f_1.

As for g_θ in Gourmel et al. [17], this definition of $\tilde{g}_\theta$ does not allow its analytical evaluation. It is evaluated as follows. Given a point $\mathbf{x} = (f_1, f_2)$, $c = \tilde{g}_\theta(\mathbf{x})$ is the value we want to compute. In this explanation, we follow the illustration of Figure 10 (b). If $\mathbf{x}$ is below the 0.5-isocurve of g_θ shown in magenta (i.e., $g_\theta(\mathbf{x}) \leq 0.5$) then $c = g_\theta(\mathbf{x})$. Likewise, if $\mathbf{x}$ is below the profile curve k_θ shown in yellow (i.e., $f_2 < k_\theta(f_1)$) then, by construction, $c = f_1$. Otherwise c is numerically evaluated with a dichotomic search in the range $[0.5, 1]$, starting with $c_d = 0.75$ as the initial guess. If $\mathbf{x}$ is below the c_d-isocurve shown in red, we continue the search within the range $[0.5, c_d]$. Otherwise, we continue the search within the range $[c_d, 1]$. To determine the position of $\mathbf{x}$ with respect to the c_d-isocurve, we express its

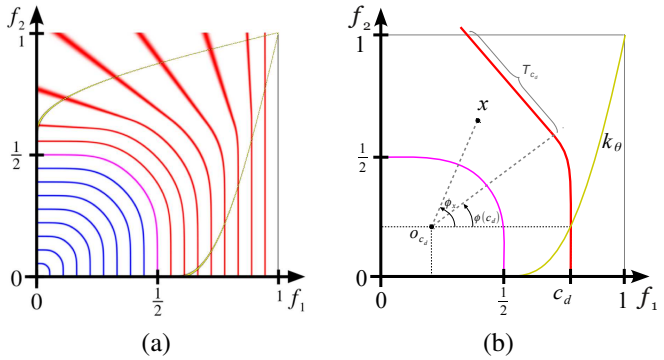


Figure 10: (a) Illustration of our asymmetric operator for details $\tilde{g}_\theta$ with $\theta = 0$ and $s = 1$. (b) Illustration of the construction of this operator. The c_d-isocurve shown in red is obtained by cutting the original isocurve of g_θ at a polar angle $\phi(c_d)$ and prolonging it by a straight line T_{c_d}. The evaluation of $\tilde{g}_\theta$ at an arbitrary position $\mathbf{x}$ is performed by iteratively determining whether $\mathbf{x}$ is above or below the isocurve of the current guess c_d. The profile curve k_θ is shown in yellow, and the 0.5-isocurve in magenta.

position in polar coordinates $(\phi_\mathbf{x}, \rho_\mathbf{x})$ in the local frame centered at $\mathbf{o}_{c_d} = (k_\theta(c_d), k_\theta(c_d))$. Then three cases occur:

- If $\phi_\mathbf{x} \leq 0$, then $\mathbf{x}$ is below the c_d-isocurve.
- If $\phi_\mathbf{x} \leq \phi(c_d)$, $\mathbf{x}$ is below the c_d-isocurve if $\rho(\phi_\mathbf{x}) < \rho_\mathbf{x}$, where $\rho(\phi)$ is the profile curve used to define the blending operator g_θ.
- Otherwise we directly check whether $\mathbf{x}$ is above or below the tangential half-line T_{c_d}.

As for g_θ, this operator is precomputed into a 3D grid, and its gradients are computed by means of finite differences. Note that precomputations in grids are easier to perform as the range of evaluation for (f_1, f_2) is now restricted to $[0, 1]^2$.

This operator is illustrated in Figure 9 for different values of the parameter s which modulates the absorption of the blended/subtracted field. In practice, we suggest using $s = 4$ which has been used for all the examples in this paper.

5. Results and discussions

The precomputation of our operator with inner and outer bounds and its partial derivatives in 128^3 grids takes about 0.9s on a Core I7 950. The same precomputations for our detail-specific operator take about 9s. Their transfer from the host memory to the device memory as 3D textures takes 3 ms. The evaluation of any operator stored in 3D textures boils down to a single texture fetch and the evaluation cost is thus irrespective to its actual equation complexity. This explains why our operators achieve the same performance as previous gradient based operators [17]. On a NVIDIA GTX 480, 100 million evaluations are done in less than 35 ms.

Using both our compact support field representation and our new operators, we can now design complex objects with adequate field variations and metrics in there inside part. These objects can be drilled with a guaranty that the resulting object is well shaped. Figure 1 illustrates a complex object built with

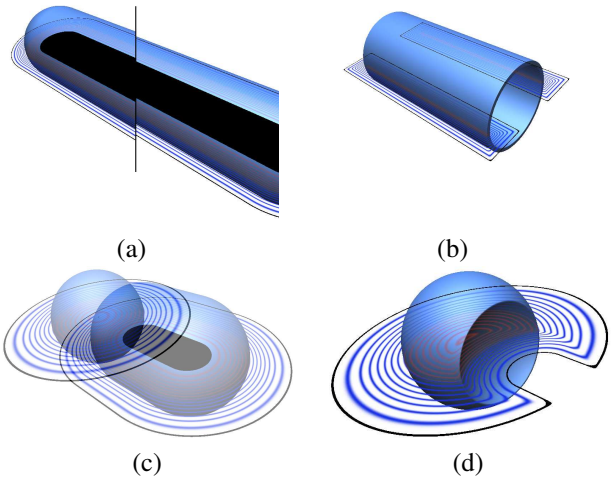


Figure 11: Adjusting the external boundary using the inner bound and subtracting field functions. Top row: (a) a cylinder is removed from a slightly larger one. Intersection with planes are used to get the final result in (b). Bottom row: (c) a capsule is removed from a sphere. Observe in (b) and (d) how the field approximates a smooth distance field around the implicit surfaces.

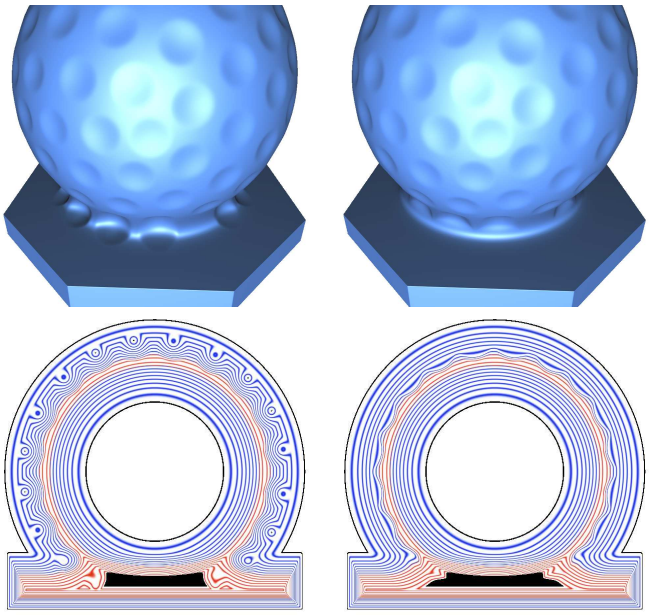


Figure 12: Illustration of our new operator for details. (a) The details are created using our difference operator and (b) using our new detail-specific difference operator. Note the field depressions introduced in (a)-bottom and the resulting blend deformation between the ball and the pedestal (a)-top that are avoided in (b) with our detail-specific operator.

several differences and blends. As we can see on the left of Figure 1, despite all there nice properties, Gourmel et al. operators fail in preserving field variations and metrics when several differences are used. The field of the large drilled spheres has been altered and a consequence is the asymmetry in the blend between the large spheres and the pedestal. As shown in the right of Figure 1, the use of our operators prevent these alterations of the field and the blend is symmetric.

When making a difference operation, a very interesting observation is that the inner field of the subtracted primitive defines a part of the outer field of the result. Another important observation is that after a composition, we expect the resulting field to follow the shape of the resulting surface as if approximating the variations of a distance field. Without inner field control, this is not the case in the inner field part when clean union and blending are used as illustrated in Figure 7 where unexpected field depressions arise. The use of an inner bound together with the adapted operators avoid this problem. This is also usually not the case for the difference. Reducing the inner radius of the combined objects enables the generation of a band around the implicit surface, as for the tubes in Figure 11 (top-row). In this band, the field smoothly approximates a distance field with the metric of the composed field functions. Figure 11 (bottom-row) shows a similar control on a capsule subtracted from a sphere.

Whereas symmetric operators are well suited for large-scale compositions (Section 4.1), when modeling small features, our detail-aware operator becomes preferable (Section 4.2). This is demonstrated in Figure 12 where a golf-ball like shape is obtained by removing small spheres from a large one, and then blending the result with a pedestal. Using symmetric difference operators, the depressions introduced in the field when removing the small spheres (Figure 12 (a)-bottom) distort the blend between the ball and the pedestal (Figure 12 (a)-top). This behavior is undesired and unexpected as these small spheres just represent a detail and the blend should mostly be as the one linking the ball (a large sphere) and the pedestal. Figure 12 (b) illustrates the improvement obtained by using our new detail specific operator. These behaviors are also illustrated in Figures 1 and 13. Figure 13 also illustrates the field variations generated when objects are built using our compact field representation together with our composition operators.

6. Conclusion

In this paper we have presented new constraints on field functions so that intersection and difference composition operators are applied in a consistent manner, avoiding field distortions and discontinuities. We also provide a method to build composition operators satisfying those constraints when intersection and difference operators are derived from union or blending. Combining these contributions allows, when applying difference operators, to dig the outer bound of the resulting field function so that its shape follows the shape of the surface.

Finally, we have introduced a new specific composition operator for the modeling of thin details on a surface. This operator smoothly absorbs the removed field, thus avoiding the introduction of undesired depressions in the resulting field function that would degrade the shape of subsequent smooth transitions.

These advanced operators do not yield analytic formulas and have to be precomputed into tables to enable fast evaluations. As future work, it might be interesting to derive analytic formulas reproducing our operators, even if that means losing C^∞ continuity. This paper highlights the importance of the quality and shape of both the inside and outside fields. In the context

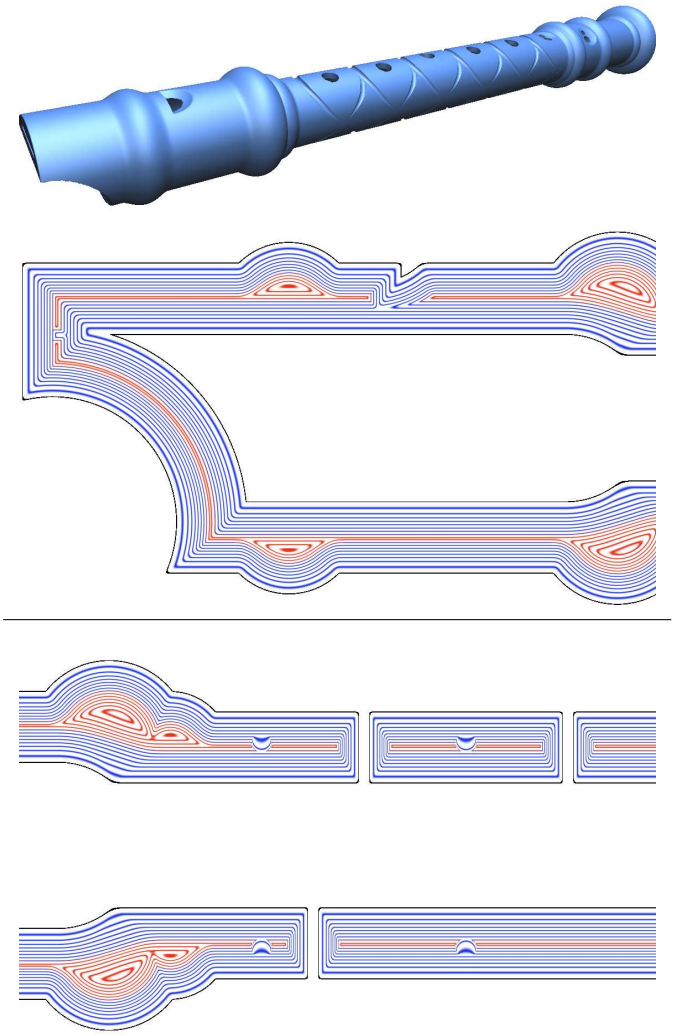


Figure 13: A flute model built using our compact field functions and our adapted composition operators including detail-specific operators. The quality of the field variations is illustrated on a vertical section of the left side of the field function. Note that unexpected depressions are avoided and the field approximates a distance field in bands located on each side of the implicit surface.

of an interactive modeling system, these observations yield interesting questions such as how to leverage a maximal control on the resulting fields? The study of interactive visualizations, especially for the creation of small details and field function based micro geometries would also be of interest. Finally, the way the details could be positioned and repeated on the surface is another direction to investigate.

7. Acknowledgments

This work has been funded by the IM&M project (ANR-11-JS02-007).

References

[1] Sabin MA. The use of potential surfaces for numerical geometry. In: Tech. Report VTO/MS/153, British Aerospace Corp., Weybridge, U.K. 1968,.

[2] Ricci A. A Constructive Geometry for Computer Graphics. Computer Journal 1973;16(2):157–60.

[3] Blinn JF. A generalization of algebraic surface drawing. ACM Trans Graph 1982;1(3):235–56.

[4] Wyvill G, McPheeters C, Wyvill B. Data Structure for Soft Objects. The Visual Computer 1986;2(4):227–34.

[5] Bloomenthal J. Bulge elimination in convolution surfaces. In: Computer Graphics Forum; vol. 16. 1997, p. 31–41.

[6] Shapiro V. Semi-analytic geometry with R-functions. Acta Numerica 2007;16:239–303.

[7] Olsen L, Samavati FF, Sousa MC, Jorge JA. Technical section: Sketch-based modeling: A survey. Comput Graph 2009;33(1):85–103.

[8] Brazil EV, Macedo I, Sousa MC, de Figueiredo LH, Velho L. Sketching variational hermite-RBF implicits. In: Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium. SBIM '10; 2010, p. 1–8.

[9] Galin E, Akkouche S. Incremental polygonization of implicit surfaces. Graphical Models 2000;62(1):19–39.

[10] Kanamori Y, Szego Z, Nishita T. GPU-based fast ray casting for a large number of metaballs. Comput Graph Forum 2008;27(2):351–60.

[11] Gourmel O, Pajot A, Paulin M, Barthe L, Poulin P. Fitted BVH for fast raytracing of metaballs. Computer Graphics Forum 2010;29(2):281–8.

[12] Wyvill B, Guy A, Galin E. Extending the CSG tree - warping, blending and boolean operations in an implicit surface modeling system. Comput Graph Forum 1999;18(2):149–58.

[13] Hsu PC, Lee C. Field functions for blending range controls on soft objects. Proc of Eurographics, Computer Graphics Forum 2003;22(3):233–42.

[14] Barthe L, Wyvill B, de Groot E. Controllable binary CSG operators for "soft objects". International Journal of Shape Modeling 2004;10(2):135–54.

[15] de Groot E, Wyvill B, van de Wetering H. Locally restricted blending of blobtrees. Computers & Graphics 2009;33(6):690–7.

[16] Bernhardt A, Barthe L, Cani MP, Wyvill B. Implicit blending revisited. Proc of Eurographics, Computer Graphics Forum 2010;29(2):367–76.

[17] Gourmel O, Barthe L, Cani MP, Wyvill B, Bernhardt A, Paulin M, et al. A gradient-based implicit blend. ACM Transactions on Graphics 2013;32(2).

[18] Barthe L, Mora B, Dodgson N, Sabin M. Interactive implicit modelling based on C1 reconstruction of regular grids. International Journal of Shape Modeling 2002;8(2):99–117.

[19] Ferley E, Cani MP, Gascuel JD. Resolution adaptive volume sculpting. Graphical Models 2002;63:459–78.

[20] Schmitt R, Wyvill B, Galin E. Interactive implicit modeling with hierarchical spatial caching. In: Shape Modelling International. 2005,.

[21] Eyiyurekli M, Breen DE. Data structures for interactive high resolution level-set surface editing. In: Proceedings of Graphics Interface 2011. GI'11; 2011, p. 95–102.

[22] Brun E, Guittet A, Gibou F. A local level-set method using a hash table data structure. Journal of Computational Physics 2012;231(6):2528–36.

[23] Bloomenthal J, Shoemake K. Convolution surfaces. SIGGRAPH Comput Graph 1991;25(4):251–6.

[24] Pasko A, Adzhiev V, Sourin A, Savchenko V. Function representation in geometric modeling: concepts, implementation and applications. The Visual Computer 1995;11(8):429–46.

[25] Barthe L, Gaildrat V, Caubet R. Extrusion of 1D implicit profiles: Theory and first application. International Journal of Shape Modeling 2001;7:179–99.

[26] Li Q. Smooth piecewise polynomial blending operations for implicit shapes. Comput Graph Forum 2007;26(2):157–71.

[27] Barthe L, Dodgson NA, Sabin MA, Wyvill B, Gaildrat V. Two-dimensional potential fields for advanced implicit modeling operators. Computer Graphics Forum 2003;22(1):23–33.

[28] Pasko GI, Pasko AA, Kunii TL. Bounded blending for Function-Based shape modeling. IEEE Comput Graph Appl 2005;25(2):36–45.

[29] Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, et al. Reconstruction and representation of 3d objects with radial basis functions. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. SIGGRAPH '01; ACM; 2001, p. 67–76.

[30] Macêdo I, Gois JP, Velho L. Hermite radial basis functions implicits.

Computer Graphics Forum 2011;30(1):27–42.
[31] Adamson A, Alexa M. Approximating and intersecting surfaces from points. In: Kobbelt L, Schrder P, Hoppe H, editors. Proceedings of the Eurographics Symposium on Geometry Processing. 2003, p. 245–54.
[32] Shen C, O'Brien JF, Shewchuk JR. Interpolating and approximating implicit surfaces from polygon soup. ACM Trans Graph 2004;23(3):896–904.
[33] Guennebaud G, Gross M. Algebraic point set surfaces. In: ACM SIGGRAPH 2007 papers. SIGGRAPH '07; New York, NY, USA: ACM; 2007,.
[34] Davis J, Marschner SR, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. In: 3DPVT. IEEE Computer Society; 2002, p. 428–38.
[35] Calakli F, Taubin G. SSD: Smooth signed distance surface reconstruction. Comput Graph Forum 2011;30(7):1993–2002.
[36] Li Q, Phillips R. Implicit curve and surface design using smooth unit step functions. In: Proc. of ACM symposium on Solid modeling and applications. SM '04; Eurographics Association; 2004, p. 237–42.
[37] Barthe L, Gaildrat V, Caubet R. Combining implicit surfaces with soft blending in a CSG tree. In: Proc. of CSG Conference Series. 1998, p. 17–31.