

Feature Based Volumes for Implicit Intersections

Oleg Fryazinov¹, Pierre-Alain Fayolle², Turlif Vilbrandt³, Galina Pasko⁴, Alexander Pasko¹

¹Bournemouth University, UK

²University of Aizu, Japan

³Digital Materialization Group, Japan, and Uformia AS, Norway

⁴British Institute of Technology and E-commerce, UK

Abstract

The automatic generation of volumes bounding the intersection of two implicit surfaces (isosurfaces of real functions of 3D point coordinates) or Feature Based Volumes (FBV) is presented. Such FBVs are defined by constructive operations, function normalization and offsetting. By applying various offset operations to the intersection of two surfaces, we can obtain variations in the shape of an FBV. The resulting volume can be used as a boundary for blending operations applied to two corresponding volumes, and also for visualisation of feature curves and the modelling of surface based structures including microstructures.

Categories and Subject Descriptors (according to ACM CCS): Computational Geometry and Object Modeling [I.3.5]: Curve, surface, solid, and object representations—

Keywords: implicit surfaces, function representation, intersection, offsetting, bounding volumes, feature based volumes

1. Introduction

Automatic generation of bounding volumes is required in many applications in computer graphics and computational geometry. In this paper we consider the problem of automatic generation of bounding volumes or Feature Based Volumes (FBV) for intersections of implicit surfaces that can be applied for defining the bounded blending regions for two intersecting surfaces, rendering of feature curves and points in interactive function-based modelling and in other applications. By FBV we mean a volumetric object that encloses all intersection components independent of their dimensionality, number and complexity.

Intuitive shape control and feature localization are key points for widening applicability of implicit surface and function-based modelling. Using a bounded blending operation between two volumes [1] generates a blend completely localized inside an additional third bounding volume, which needs to be specified by the user. An approach for the automatic generation of such a bounding volume enclosing the intersection curve between two given implicit surfaces was proposed in [2]. A polyline, which approximates the intersection curve, is extracted and then is used as a skeleton for generat-

ing the bounding volume with a convolution surface or some other skeletal based object.

In the general case, the intersection of two implicit surfaces includes multiple components of different dimensionality: points, curves and surface patches (e.g., the intersection of two cubes partially sharing a face). It is difficult or impossible to handle such intersections using the method described in [2]. In addition, handling a large number of disjoint intersection components (points and curves) can also be problematic or impossible - consider for example the intersection of two solid noise primitives or a pair of complex metaball objects. In this paper we solve this problem by proposing Feature Based Volumes (FBV) - a general function-based definition for automatically locating and generating such bounding volumes.

2. Related Work

The main approach to defining an FBV for the surface-surface intersection curve is to approximate the curve by straight or curvilinear segments and then to use them as a skeleton of FBV. Such a skeleton can serve for a distance based volume definition (skeletal blobs [3]) or a convolution surface [4].

There are several numerical approaches to implicit surface-surface intersection. One can start with some intersection point found analytically or numerically, then trace the intersection curve by differential equation solving, see, for example [5]. Another approach is to approximate both surfaces by polygons and to intersect two obtained polyhedrons. These methods treat both surfaces equally. On the other hand, one could assume that one of the defining functions can be much more time consuming to evaluate. A polygonization-based algorithm aiming to decrease the number of evaluations of the more complex function, which substantially decreases the overall computation time, was proposed in [6].

Within the level sets framework, [7] proposed to localize blends within the regions surrounding intersection curves of two isosurfaces. Bernhardt et al. [2] introduced a method for the automatic generation of bounding volumes for intersection curves between two skeletal implicit surfaces. An intersection curve can be either approximated by a polyline or considered a set of points satisfying a system of two non-linear equations representing initial surfaces. [2] describes a hybrid numerical procedure generating a polyline approximation for one or several intersection curves with the given precision.

In this work, we attempt to avoid numerical search or approximation, and instead to provide direct function-based definition of FBVs for multiple intersections. Our proposed solution is based on treating implicit surfaces and their intersections as zero-thickness solids within the Function Representation (FRep) framework [8]. This allows for the application of set-theoretic intersection operation defined by R-functions [9, 10] as well as other FRep operations such as offsetting and bounded blending.

3. Feature Based Volume Construction

We construct FBV for the intersection of two models defined by the Function Representation as follows: first, we need to determine the point-set resulting from the intersection of two surfaces represented as FRep objects (see Figs. 1a and 1b). The next step consists in using an offset operation to define the solid object representing the FBV (see Figs. 1c and 1d). In this section we consider these steps in detail.

3.1. Construction of the intersection

For a given solid with the continuous defining function of point coordinates $f(\mathbf{x}) \geq 0$, the surface is usually

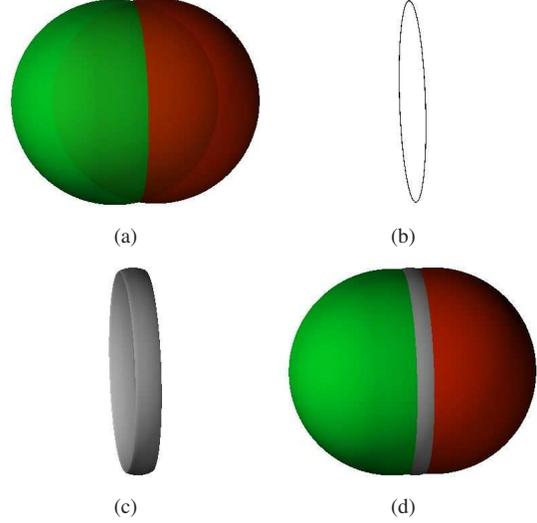


Figure 1: Feature Based Volume construction: a) Initial objects; b) Intersection curve; c) Offset of the intersection curve; d) Initial objects with Feature Based Volume.

defined by the set of points verifying $f(\mathbf{x}) = 0$. We can also use another formulation to define the surface:

$$g(\mathbf{x}) = -f^2(\mathbf{x}) \geq 0 \quad (1)$$

This function is equal to zero on the surface of the solid defined by the function f and negative everywhere else. Given two FRep objects: $f_1 \geq 0$ and $f_2 \geq 0$, by applying the intersection operation to their surfaces, we obtain the following definition for the point-set corresponding to the surface-surface intersection:

$$f_{int} = (-f_1^2) \wedge_{\alpha} (-f_2^2) \geq 0 \quad (2)$$

Here \wedge_{α} denotes the intersection operation defined by R-functions [9]. In our implementation, we use the R-function with $\alpha = 0$:

$$f_1 \wedge_0 f_2 = f_1 + f_2 - \sqrt{f_1^2 + f_2^2} \quad (3)$$

While using this R-function for the intersection, we can expect the resulting function to be C^1 -continuous in the entire space with possible exception of the intersection points. The resulting function f_{int} takes zero value at the intersection of two surfaces and negative values elsewhere. By surfaces here we consider closed surfaces, yet formally the same approach can be applied to surfaces with boundaries in the case the defining functions can be provided for them.

The volume enclosing the intersection point set can then be defined using an offset operation:

$$f_{vol} = \text{offset}(f_{int}) \quad (4)$$

This operation typically generates a new point set completely including the initial point set. In the considered problem, this is the volume enclosing the initial intersection point set between two surfaces.

3.2. Offsetting the intersection to obtain feature based volume

Here we describe several possible methods for implementing an offset operation for FRep objects. Note that in the above formulation of offsetting functions, neither dimensionality, nor the number of intersection components were taken into account. Therefore, such a defined volume has to enclose all intersection components with its shape mainly defined by the initial functions and by the type of the applied offset operation. The overall size of the FBV is controlled by the parameters of the selected offset.

3.2.1. Constant value offsetting

The simplest method for offsetting an FRep object defined by $f(\mathbf{x})$ is to change the iso-value of interest:

$$f_{const}(\mathbf{x}) = f(\mathbf{x}) + d \quad (5)$$

where $f_{const}(\mathbf{x}) = 0$ corresponds to the surface of the offset object and d is the offsetting amount. This approach is simple to implement and extremely efficient. Its obvious drawback is that it heavily depends on the properties of the function $f(\mathbf{x})$.

If $f(\mathbf{x})$ is defined by the Euclidean distance function or behaves similarly (at least in the neighbourhood of the surface of the object), then this approach will work fine. In other cases, the offset object may not have an expected shape and the offset operation may be difficult to control. Figure 2 illustrates this with an ellipse defined by: $1 - (x/5)^2 - (y/3)^2 = 0$ and the result of several offsets for values of d ranging from 0.1 to 0.5. The offset curves are flatter than the original ellipse and stretched horizontally.

Since the result of the constant value offsetting depends on whether or not $f(\mathbf{x})$ behaves like the distance function, a possible solution is to apply transformations to $f(\mathbf{x})$ to make it behave like the distance function while keeping the boundary ($f_{const} = 0$) unchanged. We discuss two methods to achieve this result in sections 3.2.2 and 3.2.3.

3.2.2. Normalization

A function f is said to be normalized to the order n if $\frac{\partial f}{\partial \nu} = 1$ and $\frac{\partial^k f}{\partial \nu^k} = 0$ for $k = 2 \dots n$ where ν is the unit normal to the surface. A normalized function behaves like the distance function near its zero set. Methods for

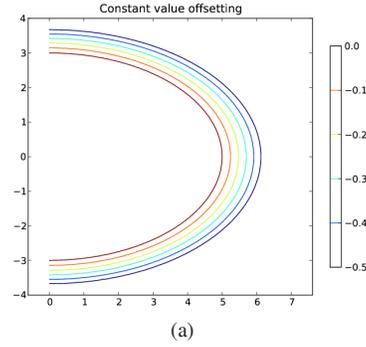


Figure 2: Constant value offsetting of an ellipse. The ellipse and five curves obtained by offsets ranging from 0.1 to 0.5 are shown. Offset ellipses are horizontally stretched.

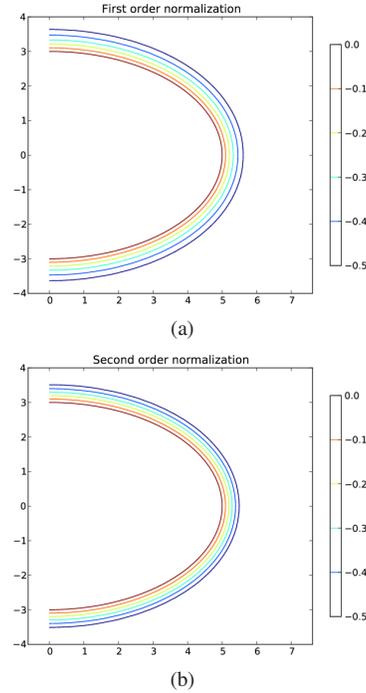


Figure 3: Normalization: a) First order normalization of f and constant value offsetting of the normalized function; b) Second order normalization of f and constant value offsetting of the normalized function.

normalizing functions have been proposed by Rvachev [9] and discussed in details by Shapiro [11, 10]. A function f with non vanishing gradient on its zero set can be normalized to the first order by the transformation:

$$f_{norm}(\mathbf{x}) = \frac{f(\mathbf{x})}{\sqrt{f^2(\mathbf{x}) + (\nabla f(\mathbf{x}))^2}} \quad (6)$$

Normalization to the order n is obtained recursively:

$$f_n = f_{n-1} - \frac{1}{n!} f_1^n \frac{\partial^n f_{n-1}}{\partial v^n} \quad (7)$$

Since R-functions preserve normalization, functions constructed by applying R-functions to primitives can be normalized by simply normalizing each primitive [9, 12, 11].

The normalization of R-functions is related to the approximation of distance introduced by Taubin in [13] for approximating the distance to the zero level-set of a function. Given an algebraic distance function: f , the Taubin approximation of distance is obtained as: $\frac{f}{|\nabla f|}$.

Figure 3a shows the previously discussed ellipse and several offsets obtained by using a first order normalization of $f(x, y) = 1 - (x/5)^2 - (y/3)^2$. Increasing the normalization order improves the approximation quality of the distance function near the zero level-set and is illustrated by Fig. 3b where the second order normalization of f is used instead. This is in the agreement with the work of Biswas and Shapiro [11]: increasing the normalization order, improves the quality of the distance function approximation near the boundary.

The normalization may produce unexpected results for functions with not well behaved gradient. As an example, please consider the following function $f(x, y) = 1 - (x/16.9)^2 - (y/0.5)^2$. Its zero level-set corresponds to a very thin ellipse. The constant value offsetting for the first order normalization of f for values of d ranging from 0.1 to 0.5 is illustrated in Fig. 4a. Artifacts can be seen for some of the offset curves. The second order normalization shown in Fig. 4b also introduces artifacts in the offset curves.

Computing the first order normalization for a given function is relatively simple. Higher order normalization requires computing higher order derivatives at runtime, which needs to be done by automatic differentiation.

3.2.3. Distance function reinitialization

Distance to the boundary $f(\mathbf{x}) = 0$ can be obtained by solving numerically the following PDE to steady state as first proposed by Sussman et al. in [14]:

$$\frac{\partial \phi}{\partial t} = \text{sign}(f)(1 - |\nabla \phi|) \quad (8)$$

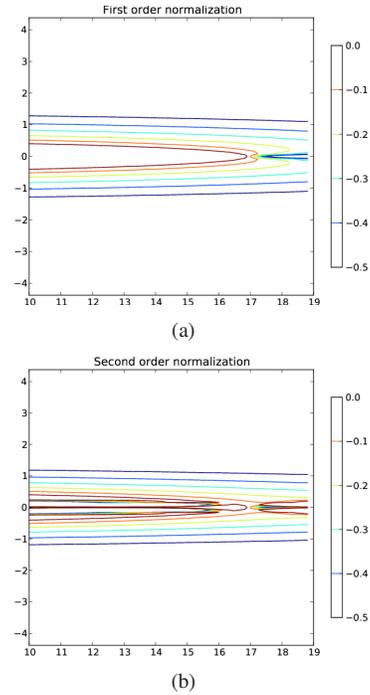


Figure 4: Artifacts after normalization of a thin ellipse: a) Artifacts on an offset thin ellipse when constant value offsetting is applied to the first order normalization; b) Artifacts are also present on the offset thin ellipse when the second order normalization is used.

where $\phi(\mathbf{x}, t = 0) = f(\mathbf{x})$ and $sign(f)$ is the sign function of f defined as:

$$sign(f) = \begin{cases} -1, & f < 0 \\ 0, & f = 0 \\ 1, & f > 0 \end{cases} \quad (9)$$

For numerical purposes, it is useful to smooth the sign function as: $sign(f) = \frac{f}{\sqrt{f^2 + \epsilon^2}}$ (see [14]); ϵ can be chosen as the length of the edge of a cell of the grid where the PDE is solved.

This approach for reinitializing a level-set to keep it a distance function was first proposed in [14]; details of the numerical implementation can be found in section 3.5 of their work. In [15] the authors show that a problem may arise when the initial function f is not close to the signed distance function and propose to solve the reinitialization problem by using a modified sign function depending on the local gradient of f : $sign(f) = \frac{f}{\sqrt{f^2 + |\nabla f|^2 \epsilon^2}}$. Further improvements for solving Eq. 8 have also been discussed in [16].

An alternative approach for computing the signed distance function to the boundary $f(\mathbf{x}) = 0$, consists in solving Eq. 10 with the Fast Marching method [17].

$$\begin{cases} |\nabla \phi| = 1 \\ sign(\phi) = sign(f) \end{cases} \quad (10)$$

Figure 5 illustrates the previous thin ellipse obtained by contouring $\phi(x, t) = 0$ and several offset curves for d ranging from 0.1 to 0.5. The PDE in Eq. 8 was solved on a grid of size 150 by 150 using 25 iterations. We follow [15] for the numerical implementation. Additionally, the first order normalization of $f = 1 - (x/16.9)^2 - (y/0.5)^2$ was used as the initial condition for $\phi(x, t = 0)$, since the first order normalization provides a better approximation to the distance function (at least close to the boundary).

Computing the distance function by the reinitialization method is more computationally intensive than normalizing the function as it requires solving numerically a PDE on a discrete grid. For the two-dimensional problem above, sampling the first order normalization of f on a 150 by 150 grid took: 0.015 sec while solving Eq. 8 on the same grid took: 0.2 sec. Both computations were run on a Sun workstation with an Intel Xeon processor (2.8 GHZ).

The solution of Eq. 8 is known on the nodes of a grid only. If a continuous function is needed in the application, then we also need to apply an interpolation or extrapolation method to the node values.

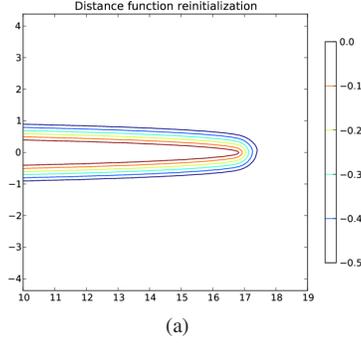


Figure 5: Correct offset curves are produced by reinitializing the distance function.

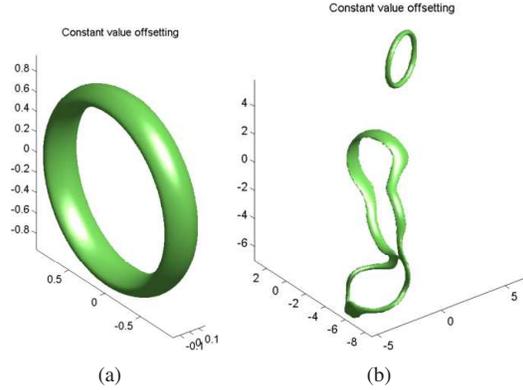


Figure 6: FBV construction by constant value offset of the intersection line: a) offset to a curve resulting from the intersection of two spheres. b) offset to curves resulting from the intersection of the surfaces of two metaball objects.

3.3. Comparison of methods for the creation of FBVs

We apply and compare the different methods proposed above for creating an FBV by applying an offset operation to the intersection of various implicit surfaces. Figure 6 corresponds to the constant value offset with value 0.1 applied to the intersection of two spheres of radius 1.0 (see Fig. 1) and to the intersection of two metaballs objects (see Fig. 10). It can be seen that for simple initial objects constant value offsetting provides good FBV, however the more complex objects we use for FBV construction the farther the result from the geometric offset.

In Fig. 7, the FBV constructed with normalization of the resulting function defined by the intersection of two implicit surfaces is shown. Figure 7a corresponds to an offset (with value 0.1) applied to the normalized intersection of two spheres and Figure 7b was obtained by offsetting (offset value 0.2) the normalized intersection of two metaball objects.

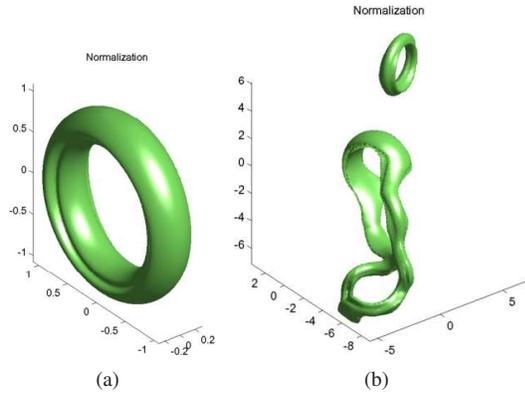


Figure 7: FBV construction by normalization of the intersection function

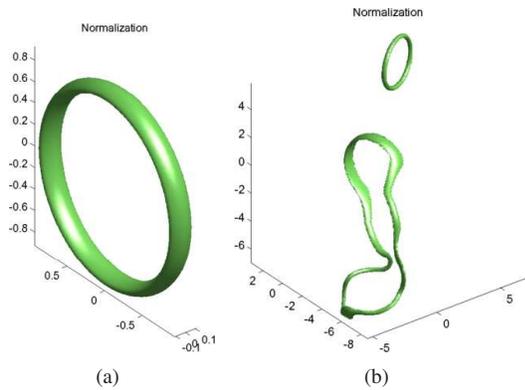


Figure 8: FBV construction by normalization of each primitive before constructing the intersection between the objects' surfaces.

For comparison, Fig. 8 illustrates the result of normalizing each primitive first by using Eq. 6 and then computing the intersection of the normalized primitives. The resulting function is normalized as a consequence of the fact that properly selected R-functions keep the normalization order of their primitives [9, 10, 11].

Finally, Fig. 9 is obtained by computing the unsigned distance to $f_{int} = 0$ by using the reinitialization method. Note that the previous approximation illustrated in Fig. 8 produces relatively similar results while being simpler to implement and faster to compute.

4. Applications and Results

In function-based modelling, FBVs can be used to define the bounding solid needed in the bounded blending operation as well as a separate solid object for other modelling purposes. In this section, we present the results of our experiments with some of the possible applications of FBVs.

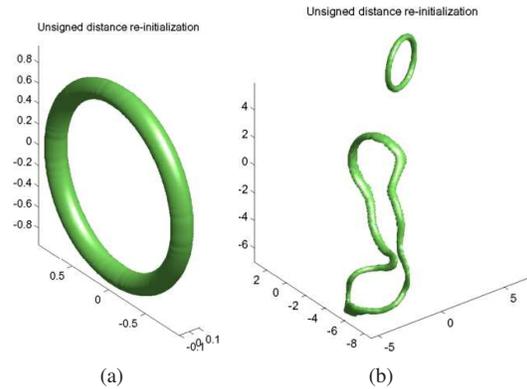


Figure 9: Computation of the unsigned distance to the intersection by the reinitialization method.

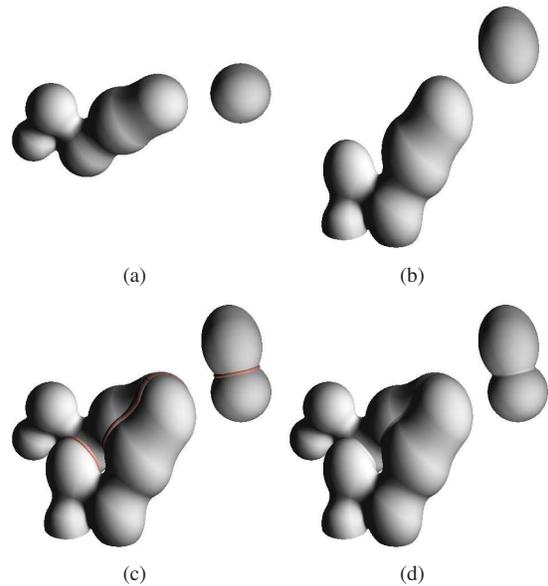


Figure 10: Bounded blending applied to two metaball models: a) and b) Initial models; c) Metaballs with the FBV obtained by offsetting the intersection curve (in red); d) Result of the bounded blending union operation using the FBV.

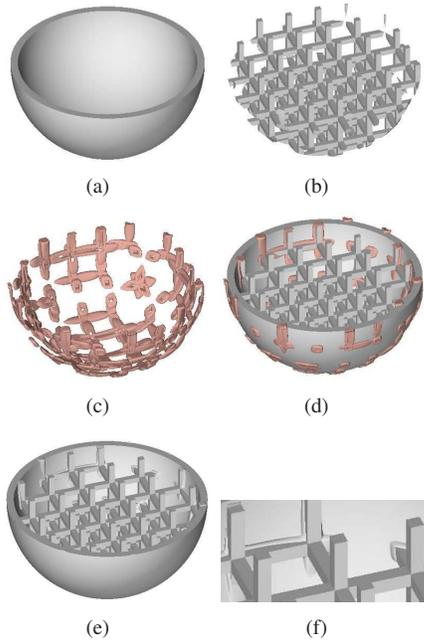


Figure 11: Bounded blending applied to two models with complex topology: a) and b) Initial models: sphere and internal procedurally defined microstructure; c) FBV; d) Initial objects together with FBV; e) Result of the bounded blending union operation using the FBV; f) Zoom to the blend area between the microstructure and the external shell.

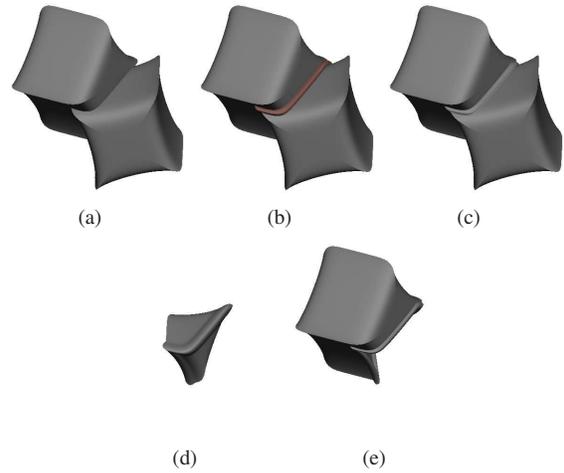


Figure 12: Using FBVs for different blending set operations: a) Two initial superellipsoids; b) Superellipsoids with the FBV obtained by offsetting the intersection curve (in red) of their surfaces; c) Result of the bounded blending union operation using the generated FBV; d) Result of the bounded blending intersection operation using the FBV; e) Result of the bounded blending subtraction operation using the FBV.

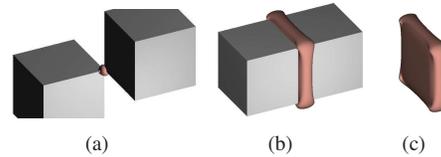


Figure 13: Feature based volumes obtained by offsetting different intersection components: a) Intersection component is a point; b) Intersection component is a face of the cube, FBV itself is shown in c).

4.1. Bounding volume for the bounded blending operation

The FBV constructed for two intersecting objects can be used to localize the bounded blending defined in [1]. The FBV can be constructed for simple objects such as superellipsoids (see Fig. 12) or cubes (see Fig. 13) as well as for more complex objects such as metaballs (see Fig. 10) and models with very complex topology, such as models with microstructure (see Fig. 11). The constant value offset was used in the example with metaballs and the normalization was used in all other examples. Any set operation (union, intersection, subtraction) with bounded blending can be applied using the corresponding R-functions [9, 8, 10] and the FBVs generated as described above. In Fig. 12d and Fig. 12e we illustrate the bounded blending intersection and bounded blending subtraction for two objects. The FBV generation is independent from dimensionality of intersection components as can be seen in Fig. 13.

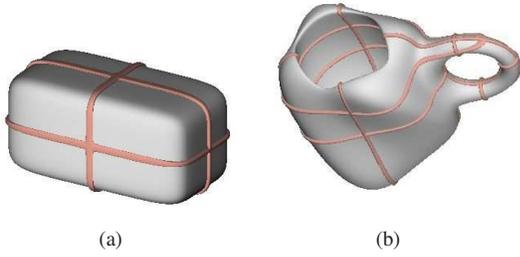


Figure 14: FBV for feature curves: a) Superellipsoid with an FBV for feature curves created by the intersection of the superellipsoid with three planes orthogonal to each axis of the coordinate system. b) Complex object with several feature curves generated by the intersection of the object with construction planes.

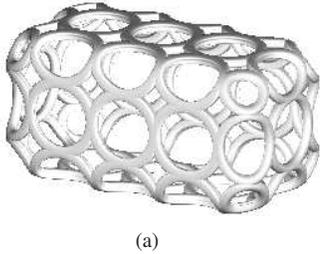


Figure 15: Modelling of on-surface microstructure: FBV created by intersection of superellipsoid with a periodical cellular structure.

4.2. Rendering feature curves

In interactive function-based modelling, it is important to provide rendering and selection of feature points and curves defined by the intersection of the modelled object with a given set of planes. For example, in Fig. 14a a superellipsoid object is rendered as well as the FBVs corresponding to an offset around the intersection of the superellipsoid with each of the three orthogonal coordinate planes. The FBV is defined as described above with $f_i \geq 0$ corresponding to the superelliptic solid and the union of three orthogonal planes defined directly by:

$$(-f_2^2) = (-x^2) \vee_{\alpha} (-y^2) \vee_{\alpha} (-z^2) \quad (11)$$

where \vee_{α} is an R-function corresponding to the union operation. In Fig. 14b a complex object constructed from several primitives with using of blending subtraction, blending union and twist operations. This example also illustrates a complex topology for the FBV that would be difficult to construct with existing methods.

4.3. Modelling of on-surface microstructures

On-surface structures and microstructures are becoming widely used in design with the development of dig-

ital fabrication hardware. Traditionally, implicit models were not popular in fabrication because of the need in an intermediate polygonization step required for the fabrication. However, this problem can be solved by using direct fabrication of FRep models. Therefore the creation of on-surface or surficial structures can also be used as an example of usage of FBVs for design purposes. We can model on-surface microstructures by intersecting the initial modelled object with periodical cellular microstructures presented in [18]. The example shown in Fig. 15 was obtained by constructing an FBV for the intersection of a superelliptic surface from Fig. 14 with a cellular microstructure surface obtained by procedurally replicating a sphere in the modelling space.

5. Conclusions and Discussion

We presented in this paper an approach to automatically constructing feature based volumes for the intersection of two implicit surfaces. We can apply different offsetting methods depending on application of the FBV. Our experiments show that in general offsetting methods can provide fast to evaluate functions by applying the constant value offset, but in this case the shape of the FBV highly depends on the distance property of the initial models and can result in an unexpected shape when a poor approximation is used. On the other hand, offsetting by using the distance function reinitialization provides good shape of the FBV, but the function is harder to evaluate. In the presented results we mostly use offsetting with normalization as the compromise between speed and quality of the shape. Other offsetting methods can also be used and provide better shape from the point of view of most CAD applications, however in general they are even harder to evaluate. Thus, the Minkowski sum of the intersection point set with a sphere, provides an ideal offset, but is more expensive to compute because of the projection operation that requires methods of global optimization for search of the global maximum of the function. A possible solution can be to use primitives with distance property and operations maintaining this distance property. However, this would restrict the number of possible operations and primitives to be used in the modelling process.

Methods for computing an offset operation for implicit surfaces that can provide good and intuitive results from the CAD point of view and can be relatively fast to compute is a task for future research. Also in some cases we need to control the shape of the FBV, especially in cases when the resulting FBV is far from the distance offsetting of the intersection curve because of

the restrictions of the selected offset operation. In this case we need to perform fitting of the parameters of the offset operation to obtain the needed shape, and this also can be a direction for future research.

References

- [1] G. I. Pasko, A. A. Pasko, T. L. Kunii, Bounded blending for function-based shape modeling, *IEEE Comput. Graph. Appl.* 25 (2) (2005) 36–45.
- [2] A. Bernhardt, L. Barthe, M.-P. Cani, B. Wyvill, Implicit blending revisited, *Computer Graphics Forum* 29 (9) (2010) 367–375.
- [3] J. Bloomenthal, B. Wyvill, Interactive techniques for implicit modeling, *SIGGRAPH Comput. Graph.* 24 (1990) 109–116.
- [4] J. McCormack, A. Sherstyuk, Creating and rendering convolution surfaces, *Computer Graphics Forum* 17 (June 1998) 113–120(8).
- [5] N. M. Patrikalakis, *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
- [6] A. G. Belyaev, A. A. Pasko, T. L. Kunii, Ridges and ravines on implicit surfaces, in: *Proceedings of the Computer Graphics International 1998, CGI '98*, IEEE Computer Society, 1998, pp. 530–.
- [7] K. Museth, D. E. Breen, R. T. Whitaker, A. H. Barr, Level set surface editing operators, *ACM Trans. Graph.* 21 (2002) 330–338.
- [8] A. Pasko, V. Adzhiev, A. Sourin, V. Savchenko, Function representation in geometric modeling: concepts, implementation and applications, *The Visual Computer* 8 (11) (1995) 429–446.
- [9] V. Rvachev, *Theory of R-functions and Some Applications*, Naukova Dumka, Kiev, 1982, in Russian.
- [10] V. Shapiro, Semi-analytic geometry with R-functions, *ACTA numerica* 16 (2007) 239–303.
- [11] A. Biswas, V. Shapiro, Approximate distance fields with non-vanishing gradients, *Graph. Models* 66 (3) (2004) 133–159.
- [12] V. Shapiro, I. Tsukanov, Implicit functions with guaranteed differential properties, in: *Proceedings of the fifth ACM symposium on Solid modeling and applications*, ACM, 1999, pp. 269–280.
- [13] G. Taubin, Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (11) (1991) 1115–1138.
- [14] M. Sussman, P. Smereka, S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, *Journal of Computational Physics* 14 (1994) 146–159.
- [15] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A pde-based fast local level set method, *Journal of Computational Physics* 155 (1999) 410–438.
- [16] G. Russo, P. Smereka, A remark on computing distance functions, *Journal of Computational Physics* 163 (2000) 51–67.
- [17] J. Sethian, A fast marching level set method for monotonically advancing fronts, *Proceedings of the National Academy of Sciences of the United States of America* 93 (4) (1996) 1591–1595.
- [18] A. Pasko, T. Vilbrandt, O. Fryazinov, V. Adzhiev, Procedural function-based spatial microstructures, in: *Shape Modeling International Conference (SMI)*, 2010, 2010, pp. 47–56.