

Multi-scale space-variant FRep cellular structures

Oleg Fryazinov¹, Turlif Vilbrandt², Alexander Pasko¹

¹Bournemouth University, UK

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

Abstract

Existing mesh- and voxel-based modeling methods encounter difficulties when dealing with objects containing cellular structures on several scale levels and varying their parameters in space. We describe an alternative approach based on using real functions evaluated procedurally at any given point. This allows for modeling fully parameterized, nested and multi-scale cellular structures with dynamic variations in geometric and cellular properties. The geometry of a base unit cell is defined using Function Representation (FRep) based primitives and operations. The unit cell is then replicated in space using periodic space mappings such as sawtooth and triangle waves. While being replicated, the unit cell can vary its geometry and topology due to the use of dynamic parameterization. We illustrate this approach by several examples of microstructure generation within a given volume or along a given surface. We also outline some methods for direct rendering and fabrication not involving auxiliary mesh and voxel representations.

Keywords: procedural modeling, microstructures, FRep, direct rendering, direct fabrication

1. Introduction

Modeling of heterogeneous objects with internal multi-scale structures has recently become an important area in CAD/CAM. Traditionally, objects with geometric structures (on micro, meso and nano levels) are represented by discrete data structures such as polygonal meshes or voxels. Despite recent advancements in algorithms and hardware that allow for the manipulation, visualization and processing of large amounts of data, modeling of highly detailed and/or complex geometric models such as microstructures is still a computationally expensive task. For simplicity, we discuss microstructures here, but proposed methods due to their procedural nature can be applied on any level to produce multi-scale superimposed (nested) structures.

Recently the computational overhead and handling of complex models involving microstructures was simplified by using function-based modeling [1][2]. Precise parametrized constructive modeling based on real functions allows for the procedural definition of multi-scale microstructures, which can undergo blending, deformations, metamorphosis and other geometric operations. Furthermore, function-based models of microstructures can be directly rendered and manufactured without generating any auxiliary representations such as polygonal meshes or voxels.

In this work we further develop cellular structures presented in [2]. The main contributions of this work include outlining several methods of generating spatial variations in microstructures: parameterization with point coordinates and distance to the external shell, metamorphosis between different unit cells, transfinite interpolation between different cellular types with given space partitions and recursive multi-scale replication. We show applications of the proposed methods for generation of

variable volumetric as well as surficial cellular structures.

2. Related works

Existing approaches to modeling microstructures rely on surfaces (triangle meshes or NURBS) and voxels [3][4][5]. Many of the known problems and limitations of both representations are amplified by the geometric complexity of microstructures. These problems become unsolvable within existing approaches when nested multi-scale structures are considered. One can mention the following problems: large model size and processing time; loss of model validity (due to cracks in surfaces, for example); limited precision caused by the approximate nature of various models; limited parameterization and operations for modeling of microstructures (such as blending between structural elements and a shell); and finally issues involving digital fabrication caused by limited model resolutions and complex slicing for fabrication.

This paper deals with function-based approaches [1][2][6], which provide solutions to most of the known problems that plague the traditional approaches to modeling in commercial CAD products. An approach to modeling periodical structures based on a triangle wave space mapping was presented in [6]. This approach was only applied to algebraic surfaces and the modeling of microstructures was not considered. In [1], periodic surfaces are defined as isosurfaces of real functions involving the summation of trigonometric functions of point coordinates. Some geometric operations such as union, intersection, difference and modulation were defined using min/max functions and algebraic operations. Defining only isosurfaces instead of solid objects (with the surface separating subspaces with different function signs corresponding to each of them) and using non-differentiable functions (min and max) prevent

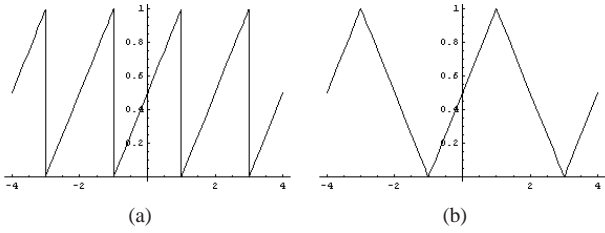


Figure 1: Replicating functions: a) Sawtooth wave, b) Triangle wave. Period $a = 2$ for both functions.

the application of additional operations on a model, such as controlled blending. This approach was further developed in [2], where microstructures are considered as solid objects defined by non-negative values of a real continuous function procedurally evaluated at any given point. The initial periodic structures, their extensions for set theoretic, controlled blending and other operations in function representation (FRep) have been combined using R-functions [7]. Regular lattices, cellular structures as well as non-regular porous structures were defined within this approach. However, methods for spatial variation of cellular structures have not been practically developed. Also, some applications require microstructures to be generated based on the proximity and features of an object surface. Below, we present methods of modeling variable volumetric and surficial cellular structures.

3. Function-based cellular structures

As it was shown in [2], cell replication similar to texture tiling can be applied to any geometric model or part of a model defined inside a bounding box (called unit cell) such that this unit cell is replicated in infinite Cartesian space. For FRep models, infinite cellular structure can be generated by applying a periodic function defining a space mapping to the FRep model of the unit cell geometry.

Given a geometric point set defined by a continuous real function $f(x, y, z)$ on the domain $I = (x_{min} \leq x \leq x_{max}, y_{min} \leq y \leq y_{max}, z_{min} \leq z \leq z_{max})$ and a periodic replicating function $g(t)$ such as $g(t) \in [0, 1] \forall t$, the cellular solid model is defined by the inequality $r(x, y, z) \geq 0$, and its surface (sometimes called implicit surface for historical reasons) is defined by the equation $r(x, y, z) = 0$ where:

$$r(x, y, z) = f(x_{min} + g(x) * (x_{max} - x_{min}), \\ y_{min} + g(y) * (y_{max} - y_{min}), \\ z_{min} + g(z) * (z_{max} - z_{min})) \quad (1)$$

The object defined by the function f on I is called a unit cell. Any periodic function can be used as a replicating function $g(t)$, however in practical modeling periodical functions with linear nature can be used. The reason for this choice is to avoid non-linear deformations of the unit cell during the replication process. In this paper we use two replicating functions: sawtooth

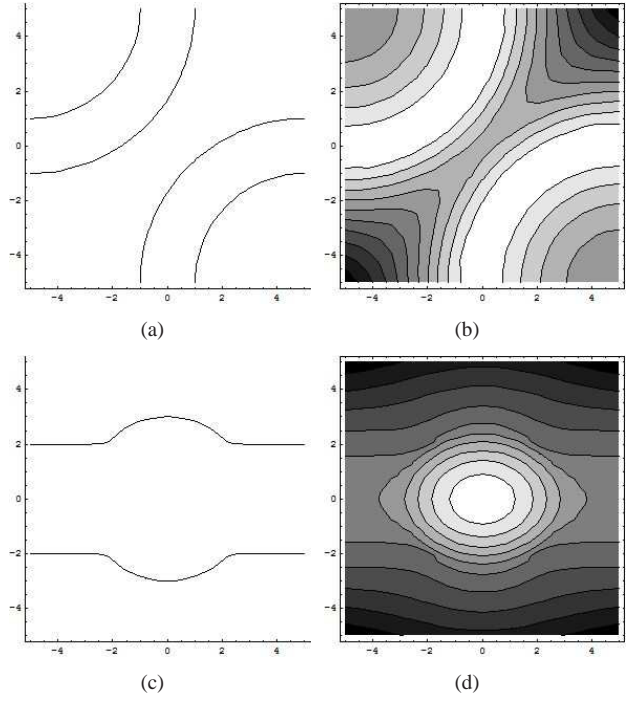


Figure 2: Connectivity types: a) Unit cell for the model with geometric connectivity, b) Function field for model in a), c) Unit cell for the model with full connectivity, d) Function field for model in c).

wave function for non-symmetric cells and triangle wave function for symmetric cells. The definition of symmetry for cells will be presented below.

A sawtooth function [8] (see Fig. 1a) can be defined by several different formulations, for example:

$$g(t) = \frac{1}{2} + \left(\frac{t}{a} - \text{floor}\left(\frac{t}{a} + \frac{1}{2}\right) \right) \quad (2)$$

A triangle wave [9](see Fig. 1b) can also be defined by using different formulations, for example:

$$g(t) = \frac{1}{2} + \frac{1}{\pi} \sin^{-1} \left[\sin\left(\pi \frac{t}{a}\right) \right] \quad (3)$$

In these functions a represents the period. Note that these functions are modified to comply with requirements of the replicating function - i.e. the value of the function lies in $[0, 1]$.

As it can be seen, the sawtooth function has periodical discontinuities, therefore the resulting cellular model has discontinuity on the faces of each cell in the cases where the unit cell has no connectivity. We distinguish two types of connectivity in function-based models:

- Geometric connectivity - the objects boundary curves at the opposite faces of the unit cell bounding box have to be equal (see Figs. 2a and 2b);
- Full connectivity - the function field is equal on the opposite faces of the unit cell bounding box (see Figs. 2c and 2d).

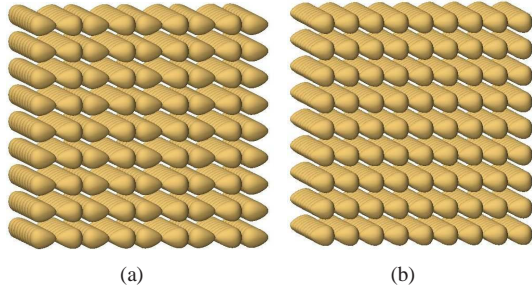


Figure 3: Non-symmetric unit cell: a) Replication with triangle wave, b) Replication with sawtooth function.

These types of connectivity can be represented formally. Given the unit cell defined by the function $f(x, y, z)$ on the domain I , the unit cell has geometric connectivity if

$$\begin{aligned}
 \{(y, z) | f(x_{min}, y, z) = 0\} &= \{(y, z) | f(x_{max}, y, z) = 0\} & y, z \in I \\
 \{(x, z) | f(x, y_{min}, z) = 0\} &= \{(x, z) | f(x, y_{max}, z) = 0\} & x, z \in I \\
 \{(x, y) | f(x, y, z_{min}) = 0\} &= \{(x, y) | f(x, y, z_{max}) = 0\} & x, y \in I
 \end{aligned} \tag{4}$$

The unit has full connectivity if the following condition is met:

$$\begin{aligned}
 f(x_{min}, y, z) &= f(x_{max}, y, z) & \forall y, z \in I \\
 f(x, y_{min}, z) &= f(x, y_{max}, z) & \forall x, z \in I \\
 f(x, y, z_{min}) &= f(x, y, z_{max}) & \forall x, y \in I
 \end{aligned} \tag{5}$$

From the practical point of view, if the unit cell has full connectivity, by using sawtooth wave we have C^0 -continuous function in the entire domain. In case of geometric connectivity we obtain geometric continuity of the entire model, however the resulting function can be C^0 -discontinuous on the faces of each cell. In case where no full or geometric connectivity conditions are met, the resulting cellular model has geometric discontinuity on the faces of the cells.

Unlike the sawtooth function, the triangle wave does not have C^0 -discontinuities. However the nature of the triangle wave function requires the unit cell to be symmetrical around the centre of the unit cell in respect to the coordinate axes. More formally, the symmetry of the cell can be defined as following:

$$\begin{aligned}
 f(x_{min} + t * (x_{max} - x_{min}), y, z) &= \\
 f(x_{max} - t * (x_{min} - x_{max}), y, z) & \quad \forall y, z \in I, t \in [0, 1] \\
 f(x, y_{min} + t * (y_{max} - y_{min}), z) &= \\
 f(x, y_{max} - t * (y_{min} - y_{max}), z) & \quad \forall x, z \in I, t \in [0, 1] \\
 f(x, y, z_{min} + t * (z_{max} - z_{min})) &= \\
 f(x, y, z_{max} - t * (z_{min} - z_{max})) & \quad \forall x, y \in I, t \in [0, 1]
 \end{aligned} \tag{6}$$

In the case where the unit cell is not symmetric in the resulting cellular structure every second cell is mirrored with respect to the centre of the unit cell (see Fig. 3a). Replication of non

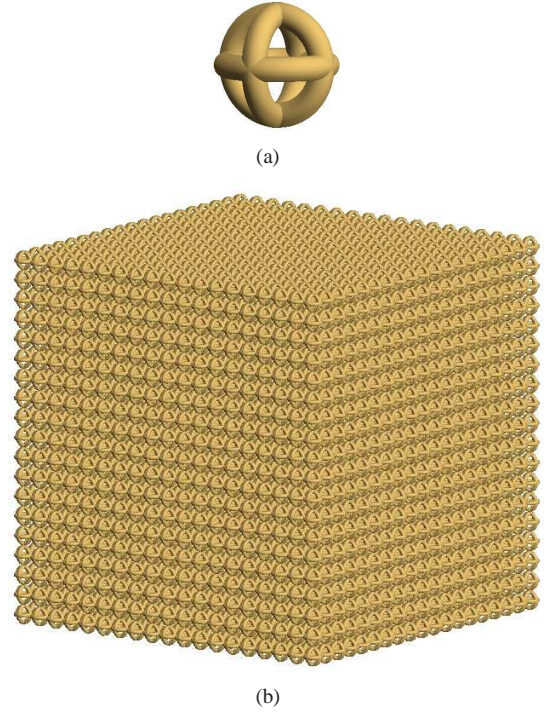


Figure 4: A regular cellular structure with union of three tori as a unit cell: a) Unit cell, b) Cellular structure.

symmetric unit cells without mirroring can be obtained with sawtooth wave (see Fig. 3b).

If we select not to use a replicating function $g(t)$ from the equation 1, but to use a linear function instead, for example $g(x) = \frac{x-x_{min}}{x_{max}-x_{min}}$, we can obtain a cellular structure where replication takes place only along some of the coordinate axes. Therefore we can distinguish several types of replication: linear replication, where only one function of $g(x)$, $g(y)$ and $g(z)$ is periodical; plane replication has two periodical functions; and volumetric replication, where all three functions are periodical.

An example of simple volumetric replication is shown in Fig. 4. In this example we take the set-theoretic union of three tori as the unit cell using R-functions [7]. Because of the symmetrical nature of the unit cell, we apply a triangle wave function as the replicating function for all three coordinate axes. The explicit defining procedure for this model can be found in the Appendix (Algorithm 1) of this paper.

4. Variable cellular structures

The mathematical nature of the definition of the cellular structures allows us to obtain a large variety of different models by replacing parameters of the cellular structure by parametric functions. This parameterization of the structure can be applied to the parameters of the replicating function, to the parameters of the unit cell or be a mixture of both.

4.1. Variable cellular structures with parameterized unit cell

We can see that the equation 1 can be rewritten in the form

$$r(x, y, z) = r(r_x(g(x)), r_y(g(y)), r_z(g(z))) \tag{7}$$

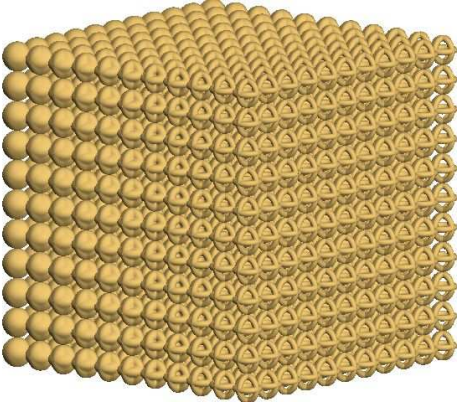


Figure 5: A variable cellular structure with the unit cell metamorphosing in space from a ball to a union of three tori.

where

$$\begin{aligned} r_x(g(x)) &= x_{min} + g(x) * (x_{max} - x_{min}) \\ r_y(g(y)) &= y_{min} + g(y) * (y_{max} - y_{min}) \\ r_z(g(z)) &= z_{min} + g(z) * (z_{max} - z_{min}) \end{aligned} \quad (8)$$

We obtain variable cellular structures by replacing linear functions f_x , f_y and f_z by arbitrary non-linear functions. Note that for cellular structures these arbitrary functions should have a replicating function over the coordinate variables as the argument. In the general case, the cellular function with the parameterized unit cell can be defined as:

$$\begin{aligned} r(x, y, z) &= r(r_x(g(x), g(y), g(z)), \\ &\quad r_y(g(x), g(y), g(z)), \\ &\quad r_z(g(x), g(y), g(z))) \end{aligned} \quad (9)$$

By using dependency of the unit cell parameters, we can obtain a different shape of the unit cell within the cellular structure. For example, in the case of a unit cell that is defined in 4D space (x, y, z, t) with the additional t parameter, we can obtain metamorphosis inside the cellular structure by using dependency of the t parameter on the coordinate values (x, y, z) (see Fig. 5 and Appendix, Algorithm 2).

4.2. Variable cellular structures with parameterized replicating function

We can vary parameters of the cellular structure by using additional parameterization of the replicating function. In general, we obtain parameterization of the replicating function by replacing arguments of the replicating function g in equation 1 by functions $g_x(x, y, z)$, $g_y(x, y, z)$ and $g_z(x, y, z)$. However, in practice the only parameter that can be variable in the replicating function is the period denoted by a in equations 2 and 3. Therefore by introducing a variable period for the cellular structure, it can be defined by the following function:

$$\begin{aligned} r(x, y, z) &= r(x_{min} + g_x(x, y, z) * (x_{max} - x_{min}), \\ &\quad y_{min} + g_y(x, y, z) * (y_{max} - y_{min}), \\ &\quad z_{min} + g_z(x, y, z) * (z_{max} - z_{min})) \end{aligned} \quad (10)$$

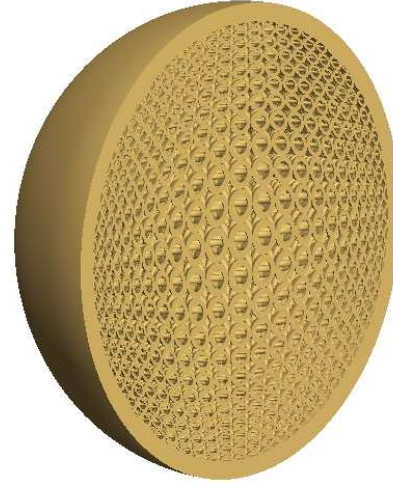


Figure 6: A variable cellular structure with the period parameterized by the distance to the external object shell.

where $g_x(x, y, z)$, $g_y(x, y, z)$ and $g_z(x, y, z)$ in the case of the saw-tooth function as the replicating function can be written as:

$$\begin{aligned} g_x(x, y, z) &= \frac{1}{2} + \left(\frac{x}{a_x(x, y, z)} - \text{floor}\left(\frac{x}{a_x(x, y, z)} + \frac{1}{2}\right) \right) \\ g_y(x, y, z) &= \frac{1}{2} + \left(\frac{y}{a_y(x, y, z)} - \text{floor}\left(\frac{y}{a_y(x, y, z)} + \frac{1}{2}\right) \right) \\ g_z(x, y, z) &= \frac{1}{2} + \left(\frac{z}{a_z(x, y, z)} - \text{floor}\left(\frac{z}{a_z(x, y, z)} + \frac{1}{2}\right) \right) \end{aligned} \quad (11)$$

where $a_x(x, y, z) > 0$, $a_y(x, y, z) > 0$, $a_z(x, y, z) > 0$, $\forall x, y, z \in \mathfrak{R}$.

As an example of a variable cellular structure with a parameterized replicating function, consider a model where the replication period depends on the distance from the given point to the boundary of the external carrying shell (see Fig. 6).

As a more complex example we consider a model obtained with the transfinite interpolation [10] of the period of the replicating function (Fig. 7d) and the transfinite interpolation of the shape of the unit cell using metamorphosis (Fig. 7e) between two unit cells (Figs. 7a and 7b). In both cases, two space partitions are defined by spheres with constant values of considered parameters inside of them. For visualization purposes in Fig. 7 we used plane replication while defining cellular structure, however the same technique can be applied for cellular structures defined by the volumetric replication.

5. Applications of volumetric microstructures

Variable cellular structures defined by function-based models are yet to be used in practical modeling and design. In this section we present some potential applications of cellular structures including variable volumetric and surficial structures.

5.1. Volumetric cellular structures in multi-scale function-based modeling

Modeling with cellular structures using variable replication takes place inside function-based modeling framework. This

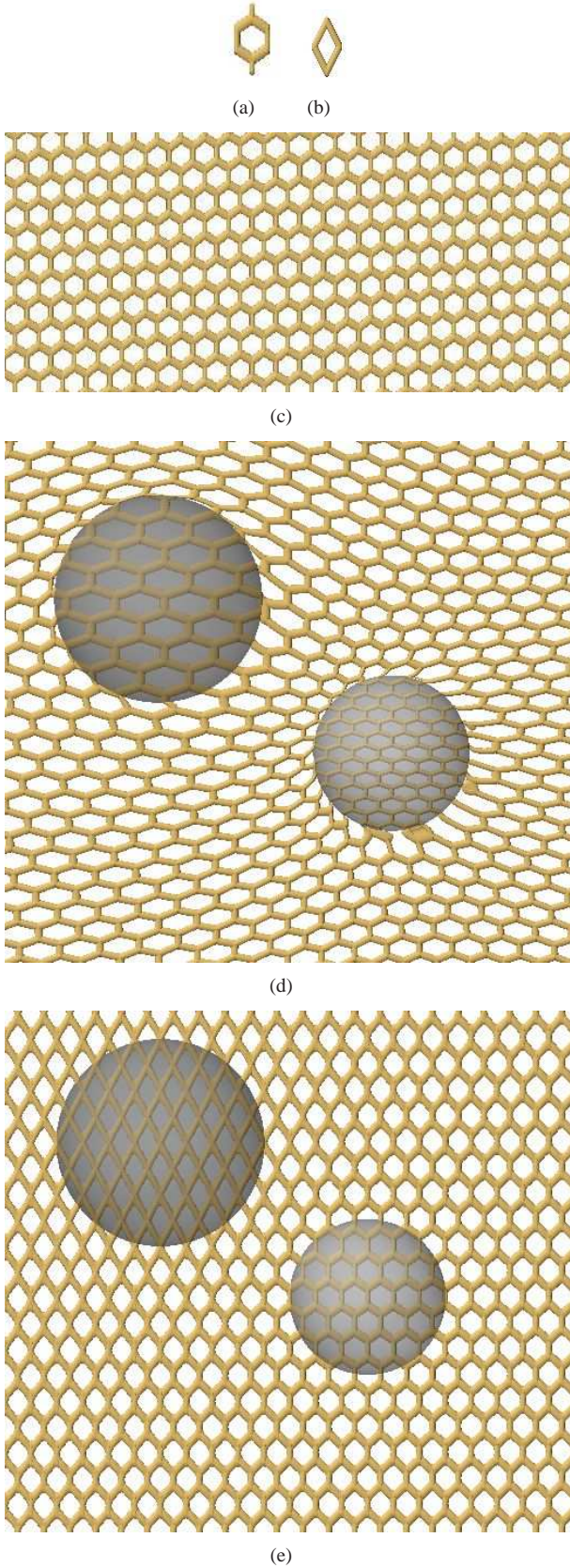


Figure 7: Transfinite interpolation in cellular structures: a) Hexagonal unit cell; b) Rhombic unit cell, c) Hexagonal cellular structure; d) Period interpolation between space partitions denoted by spheres with assigned constant periods; e) Cell shape interpolation between space partitions denoted by spheres with assigned constant cell shapes using metamorphosis between hexagonal unit cell and rhombic unit cell.

means that the resulting model is a solid object itself that can be taken as an input for another function-based operation possibly including another cellular replication.

The first example we would like to consider is metamorphosis applied to a cellular microstructure and a larger scale object (see Fig. 8 and Appendix, Algorithm 3). In this example we take the cellular structure with a unit cell as union of three tori and apply metamorphosis operation (using the linear interpolation of defining functions) between the cellular structure and another torus model on a larger scale. The application of such an operation can be useful in artistic design.

Another example of a cellular structure applied in CAD is a model of the filter with several levels of scale for the cellular replication. First, we define the cellular structure based on a basic unit cell (union of three cylinders), apply a few set-theoretic operations to the cellular structure (Fig. 9a) and then use the result as the unit cell for another cellular structure (Fig. 9b). The resulting multi-scale cellular structure also can be used for further operations including replication (Fig. 9c).

5.2. Surficial structures

Cellular structures can be used not only for modeling volumetric microstructures. Surficial or on-surface structures can be created by using cellular structures located near the surface of some solid object. Obviously, the surface in the geometric sense has zero thickness, so we are finding a way to increase the thickness and therefore we discuss on-surface structures rather than purely surface structures. Below we consider possible ways to construct surfaces microstructures based on offsetting and set-theoretic operations.

5.2.1. Surficial microstructures as feature-based volumes

We can obtain surficial microstructures by using bounding volumes that enclose the intersection curves [11]. Given the initial object defined by the function $f_{obj}(x, y, z) \geq 0$ and cellular structure defined by the function $f_{cell}(x, y, z) \geq 0$, the implicit curve that defines the intersection of the surfaces of the initial object and the cellular structure can be defined by the following:

$$g(\mathbf{x}) = (-f_{obj}^2) \& (-f_{cell}^2) \geq 0 \quad (12)$$

Here $\&$ denotes set-theoretic intersection operation using R-functions [7]. This definition arises from the fact that for any solid object defined by inequality $f(x, y, z) \geq 0$ the set of points lying on the surface is defined by inequality $-f^2(x, y, z) \geq 0$, as the given function is equal to zero on the surface of the object and is negative everywhere else.

The bounding volume for the intersection curve can be found by applying offsetting operation to the function defining the intersection curve. A number of various offset operations exists for function-based modeling. Depending on the defining functions for an initial object and the cellular structure, different offsetting operations can be used. When choosing an offsetting operation, we should balance between easiness and speed of calculations and the properties of the resulting shape. Thus, the easiest offsetting operation is the constant-value offset defined as:

$$f_{const}(\mathbf{x}) = f(\mathbf{x}) + d \quad (13)$$

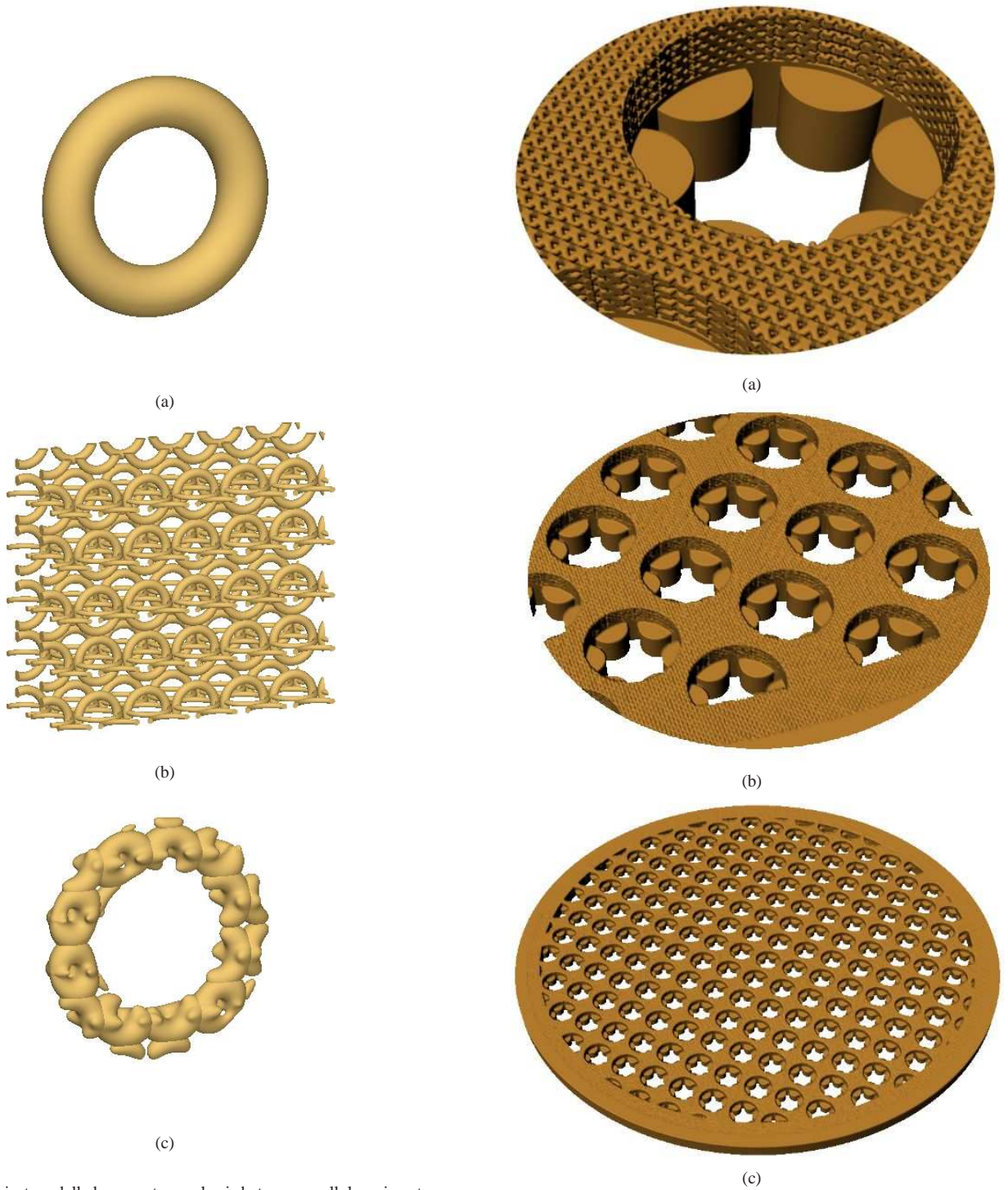


Figure 8: An object modelled as a metamorphosis between a cellular microstructure and a larger scale torus in space: a) Torus, b) Cellular structure, c) Resulting model. The formulation for this example can be found in the appendix.

Figure 9: Nested cellular structures in modeling: a) Cellular structure for the unit cell with additional CSG operations, b) Cellular structure from the unit cell defined in a) with additional CSG operations (zoom out level comparing to a) is 8x times), c) Resulting model (zoom out level comparing to b) is 6x times).

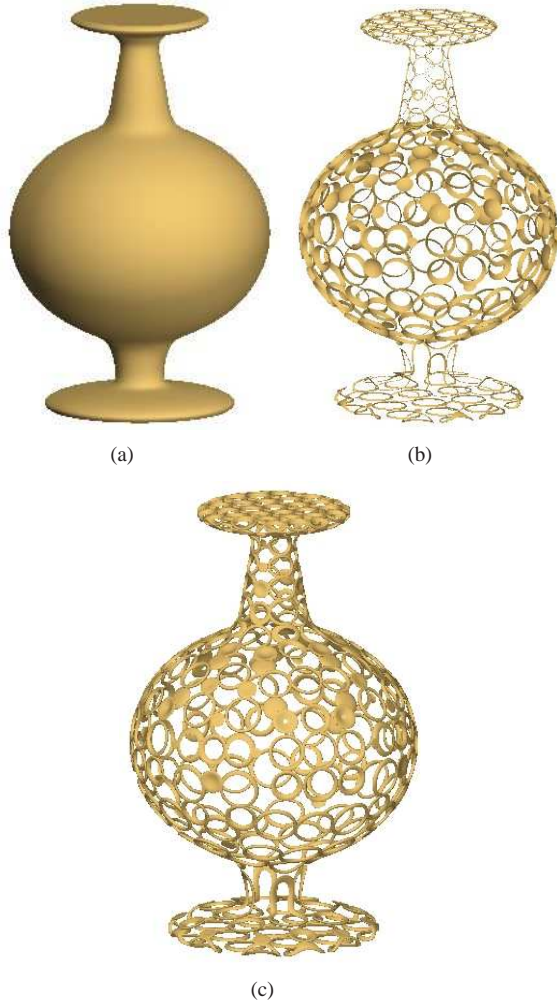


Figure 10: Surficial microstructures: a) Initial model, b) Feature-based volume after intersection with the volumetric cellular structure with the unit cell as a sphere, constant value offset, c) Feature-based volume after intersection with volumetric cellular structure with the unit cell as a sphere, offset with the function normalization



Figure 11: Set-theoretic subtraction of the volumetric cellular structure from the initial object shell.

where $f_{const}(\mathbf{x}) \geq 0$ corresponds to the solid object representing offset of the object defined by $f(\mathbf{x}) \geq 0$ and d the offsetting amount. This offsetting is simple to implement and extremely efficient, however it depends heavily on the distance property of the function and therefore it can produce unpredictable shapes.

Another approach to offsetting uses function normalization. Then the offsetting operation is:

$$f_{norm} = \frac{f}{\sqrt{f^2 + (\nabla f)^2}} + d \quad (14)$$

here $f_{norm} = f_{norm}(\mathbf{x}) \geq 0$ corresponds to the solid object representing offset of the object defined by $f(\mathbf{x}) \geq 0$ and d is the offset amount. In general, the shape after the normalization is closer to the constant-radius offset in the sense of Euclidean distance, however the normalization may produce unexpected results for functions with the not well behaved gradient.

In Fig. 10 we show a surficial structure obtained by the described method. We take the model of the vase as the initial solid object (see Fig. 10a) and the variable cellular structure where period varies over the z axis. Then the bounding volume was constructed as described above. In Fig. 10b, constant-value offset was used and in Fig. 10c offset with the normalization was used. It can be seen from the examples that the shape of the resulting surficial structures is not ideal when using a simple offsetting operation. Better shapes can be obtained using a geometric offset by applying a Minkowski sum [12] of the intersection curves with a sphere, however this operation is computationally very expensive.

5.2.2. Surficial structures after set-theoretic operations

Surficial structures can be obtained in more traditional way by using mostly set-theoretic operations. Thus, the on-surface structure can be a result of set-theoretic intersection of a shell of the initial solid object and some volumetric cellular structure. The shell here can be obtained by either using offsetting operation or by re-modeling. When using an offsetting operation we can apply one of the following approaches to create a shell:

- Take the surface of the object as $-f^2(x, y, z) \geq 0$ and then apply the offsetting operation as described above;
- Apply offsetting operation with positive value to the object and subtract the initial object from the result;

In the case of re-modeling, the same object is modeled slightly smaller or larger by copying the original object and modifying its parameters to shrink or expand the overall shape. A shell is obtained by a set-theoretic subtraction of the two objects.

Fig. 11 shows the results of an intersection between a variable volumetric cellular structure with the shell of an object. In this example, to obtain the shell we subtracted the initial object from its positive offset.

5.3. Practical modeling of volumetric microstructures and comparison with other models

Because of parametrized nature of FRep models used for cellular structures, the modeling process should not be difficult

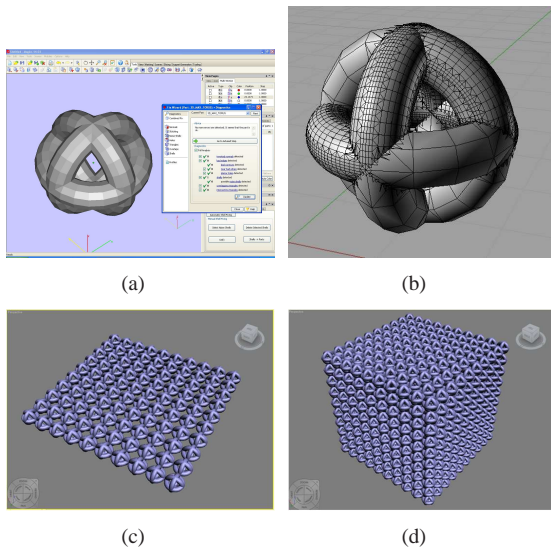


Figure 12: Results from making the torii and following construction of the replication in traditional polygonal/NURBS CAD systems.

from a user point of view. Simple identification of the spatial area that should be replicated by a unit cell is enough to make even very detailed and complex regular micro-structures such as the example complex filter given in Fig. 9, where a micro-structure is used in modelling another micro-structure. For more complex non-regular examples it is possible to simply construct any arbitrary volume (another model) and identify it as a boundary (with arbitrary soft transitions if needed) for any non-regular operation such as scale, warping, change in the microstructures parameters, metamorphosis, etc. In addition any micro-structure can be used and incorporated into any model or combined with other any other operations such as joining or blending of the micro-structure with a shell as seen again in Fig. 9. An FRep based system can easily provide a robust and dynamic framework for complex multi-scale micro-structure modeling for designers.

Using FRep to model volumetric microstructures not only performs better but can easily create models that traditional modeling system fail to create. In a simple example of the torus ball three tori are placed on each axis and unioned together into a "ball" (see Fig. 4). They are then placed in an array of $10 \times 10 \times 10$ in X, Y and Z. The "ball" is arrayed such that each ball overlaps or loses part of the the outer diameter. This allows the microstructure to be joined together as would be necessary in a real world application. This results in a microstructure block of tori balls of a thousand joined cells that form a 3-manifold and as such can be sliced for additive manufacturing. The same example was attempted with traditional software systems. The first modelling package used was unable to union three tori in the correct configuration so a test with NURBS was not performed. Several packages had great difficulty when attempting the same procedure with meshes. Only after several attempts and at low resolution (at higher mesh resolutions it failed or produced bad results) was it possible (see Fig. 12a and 12b).

The construction of $10 \times 10 \times 1$ "sheets" and the entire microstructure block is even more time- and memory consuming operation with using of meshes. The process to union the units into $10 \times 10 \times 1$ "sheets" alone took 6 minutes to complete and while it was possible to copy the sheets 10 units high it was impossible to union together the final structure. Attempting to do so resulted in the software and hardware failing when using several industrial packages of software - one designed for additive fabrication. Even so the un-unioned and therefore un-manufacturable model it was hundreds of megabytes in size. By comparison the same model can be represented by less that hundred bytes and take only a few minutes to construct using FRep modeling tools.

6. Rendering and fabrication

Application areas of multi-scale structures modeling such as material and biomedical tissue design and engineering require specific procedures for model rendering and fabrication. Currently, a function-based model has to be converted to some auxiliary representation such as a polygonal mesh or a voxel array for subsequent rendering using modern graphics hardware and additive manufacturing. The conversion to a mesh involves an isosurface polygonization (tessellation) while voxelization algorithms must be used to produce a voxel array. The known drawbacks of both these representations in the case of microstructure modeling were discussed above. Approaches to direct rendering and fabrication have to be considered to overcome these drawbacks.

6.1. Direct rendering

By direct rendering we mean accelerated ray-tracing or ray casting of function-based object surfaces without involving polygonization. The acceleration can be either by using graphics hardware (GPU)[13][14] or by using multi-threading on CPU[15]. Note that the nature of microstructure models requires rendering to be reliable. In [16], revised affine arithmetic was shown as a fast and reliable technique for ray-tracing of implicit surfaces. Also by using the same technique, we can construct reliable enumeration of the object in 3D-space and by using enumeration information decrease the number of computations by calculation ray-surface intersection only in the areas of space where points of zero value of the function can be present. Most of the pictures in this paper were obtained by using described direct rendering with modified version of Pov-Ray ray-tracing software and our own ray-casting software.

6.2. Direct fabrication

An approach to directly fabricate FRep models without auxiliary formats such as traditional STL (triangle soup) is an active direction of research. One possibility to fabricate FRep objects directly is to produce a raster image for each layer of 3D printing at the machine resolution, which is an acceptable input for some existing machines. Thus the modelled structure could be procedurally defined on a grid of voxels that corresponds directly to the layer thickness and to the pixel spacing at

the machine resolution. Another approach is to directly control the material deposition process. However, there are obstacles presented by the proprietary nature of most digital fabrication technologies, such as access to machine protocols and control commands. The wide adoption of direct fabrication obviously requires open hardware systems.

7. Conclusions

In this paper we presented an approach to modeling of variable cellular structures and their applications. The resulting models are defined procedurally within the function representation framework. This allows for further operations on the modeled structures including the creation of nested multi-scale cellular models. In practical modeling, our method has one serious restriction - there is no easy way to use existing microstructures defined using other types of representations or in traditional CAD packages. This can be resolved using polygon-to-function and voxel-to-function conversion procedures. We also outlined how traditional problems with modeling microstructures such as polygonization or voxelization for rendering and fabrication can be avoided by using direct rendering and direct fabrication.

Appendix A. Procedures defining some of the examples presented in the paper

Algorithm 1 Construction of a regular cellular structure with union of three tori as a unit cell (Fig. 4)

Procedure: `regular(x, y, z)`

Coordinate transformation by using triangle wave:

$$\begin{aligned} x_t &= \frac{1}{2} + \frac{1}{\pi} \sin^{-1}[\sin(\pi \frac{x}{2})] \\ y_t &= \frac{1}{2} + \frac{1}{\pi} \sin^{-1}[\sin(\pi \frac{y}{2})] \\ z_t &= \frac{1}{2} + \frac{1}{\pi} \sin^{-1}[\sin(\pi \frac{z}{2})]. \end{aligned}$$

Calculate replicated tori by using deformed coordinates:

$$\begin{aligned} torus_x &= 0.2^2 - x_t^2 - y_t^2 - z_t^2 - 0.8^2 + 2 * 0.8 * \sqrt{y_t^2 + z_t^2} \\ torus_y &= 0.2^2 - x_t^2 - y_t^2 - z_t^2 - 0.8^2 + 2 * 0.8 * \sqrt{x_t^2 + z_t^2} \\ torus_z &= 0.2^2 - x_t^2 - y_t^2 - z_t^2 - 0.8^2 + 2 * 0.8 * \sqrt{x_t^2 + y_t^2} \end{aligned}$$

Perform union operation over replicated tori:

$$result = torus_x \vee torus_y \vee torus_z$$

return result;

References

- [1] Y. Wang, Periodic surface modeling for computer aided nano design, *Computer-Aided Design* 39 (2007) 179–189.
- [2] A. Pasko, T. Vilbrandt, O. Fryazinov, V. Adzhiev, Procedural function-based spatial microstructures, in: *Proceedings of the 2010 Shape Modeling International Conference, SMI '10*, IEEE Computer Society, 2010, pp. 47–56.
- [3] M. W. Naing, C. K. Chua, K. F. Leong, Y. Wang, Fabrication of customised scaffolds using computer-aided design and rapid prototyping techniques, *Rapid Prototyping Journal* 11 (2005) 249–259.
- [4] W. Sun, B. Starly, J. Nam, A. Darling, Bio-cad modeling and its applications in computer-aided tissue engineering, *Computer-Aided Design* 37 (2005) 1097–1114.

Algorithm 2 Construction of a variable cellular structure with the unit cell metamorphosing in space from a ball to a union of three tori (Fig. 5)

Procedure: `variable(x, y, z)`

Coordinate transformation by using triangle wave:

$$\begin{aligned} x_t &= \frac{1}{2} + \frac{1}{\pi} \sin^{-1}[\sin(\pi \frac{x}{2})] \\ y_t &= \frac{1}{2} + \frac{1}{\pi} \sin^{-1}[\sin(\pi \frac{y}{2})] \\ z_t &= \frac{1}{2} + \frac{1}{\pi} \sin^{-1}[\sin(\pi \frac{z}{2})]. \end{aligned}$$

Calculate replicated tori and replicated spheres by using deformed coordinates:

$$\begin{aligned} torus_x &= 0.2^2 - x_t^2 - y_t^2 - z_t^2 - 0.8^2 + 2 * 0.8 * \sqrt{y_t^2 + z_t^2} \\ torus_y &= 0.2^2 - x_t^2 - y_t^2 - z_t^2 - 0.8^2 + 2 * 0.8 * \sqrt{x_t^2 + z_t^2} \\ torus_z &= 0.2^2 - x_t^2 - y_t^2 - z_t^2 - 0.8^2 + 2 * 0.8 * \sqrt{x_t^2 + y_t^2} \\ sphere &= 1 - x_t^2 - y_t^2 - z_t^2 \end{aligned}$$

Perform union operation over replicated tori:

$$tori = torus_x \vee torus_y \vee torus_z$$

Perform metamorphosis depending on z-coordinate:

$$t = (z + 10)/20$$

$$result = tori * t + sphere * (1 - t)$$

return result;

Algorithm 3 Construction of an object defined as a metamorphosis between cellular structure and a larger size torus (Fig. 8)

Procedure: `metacelltorus(x, y, z)`

Coordinate transformation by using triangle wave with affine scale and transform applied:

$$\begin{aligned} x_t &= 6.0 * (2.0 * (0.5 - \sin^{-1}(\sin(\pi/2.0 + \pi * ((x + 18.7)/12.0 - 0.5))))/\pi) - 1.0) - 18.7 \\ y_t &= 6.0 * (2.0 * (0.5 - \sin^{-1}(\sin(\pi/2.0 + \pi * ((y + 8.6)/12.0 - 0.5))))/\pi) - 1.0) - 8.6 \\ z_t &= 6.2 * (2.0 * (0.5 - \sin^{-1}(\sin(\pi/2.0 + \pi * ((z - 6.2)/12.4 - 0.5))))/\pi) - 1.0) + 6.2 \end{aligned}$$

Calculate replicated tori with additional affine transformations for each torus

$$x_{t1} = x_t + 18.7, y_{t1} = y_t + 8.9, z_{t1} = z_t - 6$$

$$x_{t2} = x_t + 18.8, y_{t1} = y_t + 0.1, z_{t1} = -z_t - 2.8$$

$$torus_x = 0.6^2 - x_{t1}^2 - y_{t1}^2 - z_{t1}^2 - 0.6^2 + 2 * 5.09 * \sqrt{y_{t1}^2 + z_{t1}^2}$$

$$torus_y = 0.6^2 - x_{t2}^2 - y_{t2}^2 - z_{t2}^2 - 0.6^2 + 2 * 5.09 * \sqrt{x_{t2}^2 + z_{t2}^2}$$

$$cellular = torus_x \vee torus_y$$

Calculate larger size torus

$$x_{t3} = x - 8.6, y_{t3} = y - 8.5, z_{t3} = z - 4.8$$

$$torus = 5.7^2 - x_{t3}^2 - y_{t3}^2 - z_{t3}^2 - 5.7^2 + 2 * 21.3 * \sqrt{y_{t3}^2 + z_{t3}^2}$$

Perform metamorphosis operation

$$result = cellular * 0.8 + torus * 0.2$$

return result;

- [5] C. Schroeder, W. C. Regli, A. Shokoufandeh, W. Sun, Computer-aided design of porous artifacts, *Computer-Aided Design* 37 (2005) 339–353.
- [6] N. Stolte, Infinite implicit replication: Case study for voxelizing and representing cyclical parametric surfaces implicitly, in: *Proceedings of the Shape Modeling International 2002 (SMI'02)*, IEEE Computer Society, 2002, pp. 105–110.
- [7] A. Pasko, V. Adzhiev, A. Sourin, V. Savchenko, Function representation in geometric modeling: concepts, implementation and applications., *The Visual Computer* 11 (1995) 429–446.
- [8] E. W. Weisstein, Sawtooth wave, *MathWorld—A Wolfram Web Resource*, 2010.
- [9] E. W. Weisstein, Triangle wave, *MathWorld—A Wolfram Web Resource*, 2010.
- [10] V. L. Rvachev, T. I. Sheiko, V. Shapiro, I. Tsukanov, Transfinite interpolation over implicitly defined sets, *Computer Aided Geometric Design* 18 (2001) 195–220.
- [11] O. Fryazinov, P.-A. Fayolle, T. Vilbrandt, G. Pasko, A. Pasko, Feature based volumes for implicit intersections, *Comput. Graph.* 35 (2011) 524–531.
- [12] A. Pasko, O. Okunev, V. Savchenko, Minkowski sums of point sets defined by inequalities, *Computers & Mathematics with Applications* 45 (2003) 1479 – 1487.
- [13] J. M. Singh, P. J. Narayanan, Real-time ray tracing of implicit surfaces on the GPU, *IEEE Transactions on Visualization and Computer Graphics* 16 (2010) 261–272.
- [14] C. Loop, J. Blinn, Real-time GPU rendering of piecewise algebraic surfaces, *ACM Trans. Graph.* 25 (2006) 664–670.
- [15] A. Knoll, Y. Hijazi, C. Hansen, I. Wald, H. Hagen, Interactive ray tracing of arbitrary implicits with SIMD interval arithmetic, in: *Proceedings of the 2007 IEEE Symposium on Interactive Ray Tracing*, IEEE Computer Society, Washington, DC, USA, 2007, pp. 11–18.
- [16] O. Fryazinov, A. Pasko, P. Comninos, Fast reliable interrogation of procedurally defined implicit surfaces using extended revised affine arithmetic, *Computers and Graphics* 34 (2010) 708–718.