Conformal Parametric Microstructure Synthesis for Boundary Representations

Q Youn Hong[1]* and Youngjin Park[2] and Gershon Elber[1]

[1]Computer Science Dept., Technion, Israel Institute of Technology, Haifa, Israel
[2]Dept. of Computer Engineering, Dong-A University, Busan, Korea

Abstract

The use of lattices and microstructures in geometric design have been recognized as potentially superior to solid structures due to the potential benefits in improved strength-to-weight ratios, better control over heat exchange and heat transfer, and so on.

In this work, we present a construction scheme to create parametric microstructures in a boundary representation (B-rep) model, $\mathcal{M}$, that are conformal to an arbitrary specification, including the boundary of $\mathcal{M}$. Given a B-rep model, $\mathcal{M}$, either a polygonal or trimmed-spline based, a cage, $\mathcal{T}$, is constructed around $\mathcal{M}$ to guide the synthesis of the microstructures in $\mathcal{M}$. Micro-elements are synthesized following $\mathcal{T}$, and verified to be inside $\mathcal{M}$ while bridging tiles are added as necessary. These parametric micro-elements can be heterogeneous in their material content, as well as locally vary in their geometric properties.

We demonstrate these abilities with example microstructures synthesized from both polygonal B-rep models and spline-based B-rep solids, including 3D printed parts.

1 Introduction

Consider a watertight boundary representation (B-rep) model, $\mathcal{M}$. Existing methods for synthesizing microstructures inside B-reps are, for the most part, assuming an axis parallel grid of micro-elements that is clipped to the shape in hand, $\mathcal{M}$. As a result, micro-elements near the boundary of the model are left clipped in an arbitrary way. A tiling scheme that is conformal to boundary of $\mathcal{M}$, $\partial\mathcal{M}$, while avoiding the need to clip tiles is highly desired. See Figure 1. When stress tensors or paths of heat flux of the model are prescribed, microstructure tiles better be conformal to the boundary of $\mathcal{M}$, and also possibly follow some desired directions in the interior.

Clipping based approaches, as shown in Figure 1 (a), are difficult to optimize as, typically, all tiles are identical in the initial axis-parallel 3D grid of tiles. If a region in $\mathcal{M}$ is found too weak in the analysis stage, micro-elements in (only) that region should be thickened, but this is a difficult task if the input is based on a uniform grid of tiles, clipped to the boundary of $\mathcal{M}$, $\partial\mathcal{M}$. $\partial\mathcal{M}$ is a critical zone in analyzing the expected behavior of $\mathcal{M}$. The fact that individual tiles are locally clipped in an arbitrary way near and by $\partial\mathcal{M}$, yields difficulties in predicting the local physical behaviors, in term of strength, heat transfer, etc., of $\mathcal{M}$, and $\partial\mathcal{M}$ in specific.

Recent advances in additive manufacturing (AM) technologies have enabled the fabrication of, and hence increased the demand for lattice based geometries, in design. AM is an enabling technology not only for manufacturing complex lattice structures but also for supporting (graded) heterogeneity in material representations. With AM, individual micro-elements can present different geometries and also potentially contain different (graded) materials.

*Corresponding author. email: younhong@cs.technion.ac.il

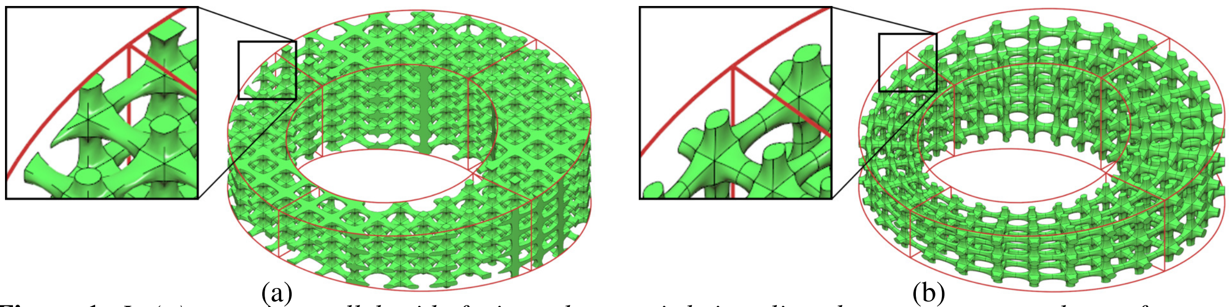

Figure 1: *In (a), an axis parallel grid of micro-elements is being clipped to a macro torus shape of a square cross section of varying size. In (b), the micro-elements follow the 'flow' of the same macro torus shape, while no tile is being clipped.*

One approach for embedding microstructured geometries in $\mathcal{M}$, while being conform to $\partial\mathcal{M}$, is based on volumetric representations [16], or V-reps: A functional composition-based framework is proposed to build microstructures in a given macro volumetric shape $\mathcal{T}$, defined as a trivariate tensor product volume. In their work, micro-element tiles, $\{\mathbf{t}_i\}_i$, are populating the box domain of $\mathcal{T}$, only to be functionally composed into $\mathcal{T}$, as $\{\mathcal{T}(\mathbf{t}_i)\}_i$. Figure 1 (b) is exploiting such an approach.

The approach of [16] is limited to a single trivariate based macro shape. In [9], an attempt was made to support conformal microstructures for V-reps that consist of several (trimmed) trivariate functions that underwent some Boolean operations. Microstructures are synthesized for each individual trivariate only to be stitched together along common zones.

In this work, we propose a method to create microstructures that are conformal to (the boundary of) a B-rep model $\mathcal{M}$ while following a prescribed vector field/tensor in the interior of $\mathcal{M}$. A cage $\mathcal{T}$ is constructed around $\mathcal{M}$ in a similar way to the creation of cages in computer graphics, toward deformations [6]. Herein, the cage is a volumetric trivariate function that prescribes the interior vector field and guides the synthesis of microstructure $\mathcal{MS}$. $\mathcal{MS}$ will be conformal to the boundary $\mathcal{M}$ and properly stitched to its boundary, while following the arbitrarily prescribed parametrization of $\mathcal{T}$. Micro-elements in $\mathcal{MS}$ are never clipped while $\mathcal{MS}$ is synthesized using a functional composition-based approach, similar to [16], but over the cage $\mathcal{T}$. As stated, the cage $\mathcal{T}$ is a trivariate function that fully contains $\mathcal{M}$ and can be created by an end user, as in computer graphics [6], following some stress tensors that result from analysis, or by interpolateing a given (vector) field, etc. Finally, being parametric, individual tiles in $\mathcal{MS}$ can also be locally controlled and modified, in their geometric properties but also in their material content.

The rest of this paper is organized as follows. In Section 2, the relevant previous work on designing, mostly conformal, microstructures is presented. In Section 3, we present our algorithm to conformally populate a B-rep model with parametric micro-elements. Then, results are presented in Section 4, and in Section 5, some extensions are discussed. Finally, we conclude this work in Section 6.

2 Previous work

We split the discussion on previous work into two. In Section 2.1, we review results with a similar aim as this work - the synthesis of conformal (parametric) microstructures over B-rep model. This work employs cages that are common in computer graphics, toward deformation mostly, and those are briefly discussed in Section 2.2.

2.1 Microstructure Construction

Populating the interior of a volumetric object with microstructures (or a set of micro-elements) has become a common practice in the modeling of 3D geometric object, while exploiting additive manufacturing (AM). One can control, with flexibility, the geometric properties of the structure while allowing the use of heterogeneous, possibly graded, materials. Microstructures have been adopted in the design of artificial bones, medical applications, or industrial designs such as airplane wings, heat exchanger, and solid rocket fuel [2]. Further details regarding the methodologies and challenges in microstructure-based modeling have complied in several survey papers, including [23, 18, 15].

Handling the micro-elements near the boundary has been one of the important issues in modeling microstructured objects. There are several different strategies to process micro-elements near or across the boundary shape: the elements can be purged as a whole, partially trimmed, or deformed with respect to the boundary shape. These strategies are chosen based on various factors in microstructure models, such as the shape of the micro-element, the representation of the microstructure and the macro (boundary) geometry, or the method used in populating or deforming the micro-elements to fit within the boundary shape.

Aremu et al. [3] define a voxel-based unit cell as a microstructure. The microstructure in a unit cell is tessellated with voxels and populated to cover the domain of the macro object. Tessellated microstructures are then trimmed by applying bitwise Boolean intersection between the microstructures and the macro shape, which is also voxelized. Each voxel is masked when the micro-element occupies the voxel, or remain void otherwise. The boundary skin is covered with a net of struts and combined with the interior microstructures using bitwise Boolean operations on voxels.

Tang et al. [24] construct lattice frames to populate uniform, conformal or random lattice microstructure in the macro volumetric object. The macro shape is voxelized according to the lattice frame and filled with strut-based micro-elements. Microstructures that are conformal to the boundary shape are fabricated by deforming voxels with respect to the lattice frame which follows the boundary shape.

Topology optimization [5] is one of the major methods in constructing microstructures with (stress) analysis and numerical optimization. In topology optimization, the microstructure is often initialized with uniform grid-like lattice or voxel-based elements. The microstructure is parameterized in terms of the radii of lattices or material distributions of the voxels, and these parameters are optimized with (stress) analysis and additional constraints. When topology optimization is used, the micro-elements near the boundary elements are aligned with the boundary shape as a result of numerical optimization. For instance, Zhu et al. [29] execute a two-stages optimization to identify the optimal material distributions and topology of the microstructures that minimize the objective energy function. Arora et al. [4] identify the optimal parameters of a truss-based lattice which aligns the lattice along the stress field of the macro object. Wu et al. [28] determine the orientation of each lattice element based on the principal directions of the macro object during the topology optimization.

Conde-Rodriguez et al. [7] models the shape of the boundary object and heterogeneous material distributions using Bézier hyper-patches. Instead of defining the geometry of the microstructure explicitly, the microstructure shape is determined by thresholding material distribution functions. Cutoff functions are used to differentiate the core material from the matrix, or separate distributions of multiple materials. Herein, the shape of the boundary object derives the boundary of microstructures.

Sosin et al. [26] viewed the construction process of the microstructure as a sphere-packing problem, where the macro object is filled with contacting spheres. To make the microstructure connected to the boundary shape, they introduced a special type of fillets to connect the outmost spheres to the boundary shape.

Some research have proposed adaptive methods to construct a microstructure. They adaptively manipulate either the macro shape or a coarse microstructure to yield a microstructure that is conformal to the boundary of the macro shape, as much as possible. For instance, Leblanc et al. [14] model a volumetric object using blocks, which are subdivided hierarchically to fit to the given macro shape. Adaptive Voids [17] fill the interior of the mesh adaptively, with tetrahedra, by tessellating the dual mesh of the boundary mesh and placing the tetrahedra hierarchically. The construction of the microstructure is started from the boundary shape, and some margins around the boundary mesh are filled with solid materials, while the interior of the shape is modeled with an adaptive size of porous cellular structures. Sitharam et al. [22] fill the boundary mesh with the microstructure of corner-sharing tetrahedra (CoSTs). Starting from the coarse CoSTs, the structure is refined with smaller tetrahedra when further details of the boundary shape must be maintained. Kambampati et al. [12] adopt function-based representation (F-rep) to represent the microstructure in the macro object, which is adaptively voxelized with different level of sparsity. A level set method is used to identify active voxels that are part of the shape. The F-rep based microstructures are filled in the macro shape and combined with the macro level surface, just as Pasko et al. [20] do.

A different approach to the construction of microstructures in a given macro shape is to apply a functional composition-based approach, where a micro-element is defined as a function in a unit box domain of a trivariate function, only to be functionally composed into the trivariate that represents the macro shape [8]. Massarwi et al. [16] propose a method to construct predefined or random microstructures in the macro object formed of trivariate splines, using functional composition. Their functional composition-based approach allows the construction of nested microstructures by multiple levels of functional compositions. Hong and Elber [9] extend Massarwi et al. [16] to populate the microstructures in a more general macro shape formed of trimmed trivariate splines, or V-reps. Toward this end, they construct the microstructures in the untrimmed trivariates and then, following the respective Boolean operation tree of the macro shape, properly trimmed the microstructures to the trimmed trivariates. As a final step, special bridging tiles are inserted to connect the existing tiles near the boundary of one trimmed trivariate to its neighboring tiles in a neighboring trimmed trivariate.

The above presented functional composition-based approaches have advantages in that they guarantee the microstructures are conformal to the macro shape, while the macro-shape is a V-rep and the resulting microstructure is parametric. In this work, we assume that the input is a watertight B-rep, and some parametric trivariate cage. Either a polygonal mesh or B-spline based B-rep is assumed. Yet, the resulting microstructure is conformal and represented as a parametric spline based. To the best of our knowledge, no microstructure construction method that is arbitrarily parametric, while conformal to the boundary of the macro shape, exists for B-reps.

2.2 Cage-based Deformation

The idea of embedding geometric objects in space and deforming the objects by deforming their embedding space, has been introduced in the 80s by Sederberg [21], and has been in a wide use in driving or manipulating skeletonal or skin based deformation in computer graphics, especially in interactive graphics [6]. The cage is a coarse net structure surrounding the geometry in hands, and each component in the driven geometry is associated with the degrees of freedom of the cage, so that high dimensional deformation is prescribed in terms of the lower dimensional changes in the cage. (A survey paper regarding the early work in cage-based deformation can be referred [19]).

Research regarding a cage-based methods often focus on the deformation of surfaces (or polygonal meshes), but there are also several publications which adopt the cage to deform not only the boundary shape but also volumetric object or other properties in the embedded model [11, 10, 25]. Our work is different from this standard usage of cages; we exploit a trivariate cage that is populated with the microstructures' tiles, while the cage is indirectly associated with the boundary shape, only to purge tiles that are not completely in the input model.

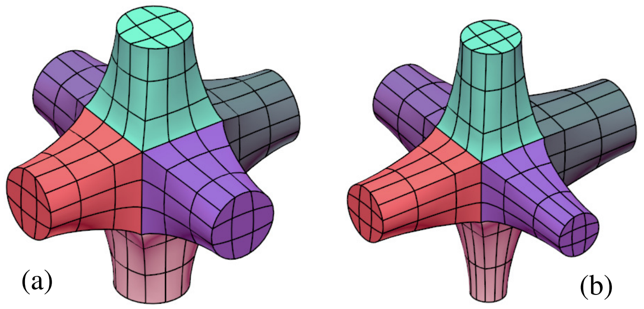

Figure 2: *The parametric tile used in this work, consisting of seven trivariates, all differently colored. In (a), the tile is uniform in all direction. However, the tile, being parametric, can, for instance, have varying arm thicknesses, as can be seen in (b).*

3 The algorithm

In order to conformally populate the volume enclosed in a B-rep model $\mathcal{M}$ with micro-elements, we augment $\mathcal{M}$ with a parametric trivariate volume $\mathcal{T}$, that is arbitrarily parameterized while fully containing $\mathcal{M}$. $\mathcal{T}$ will govern the construction of the micro-element tiles, $\{\mathbf{t}_i\}_i$, that will then be validated against $\mathcal{M}$. the construction of $\mathcal{T}$ (e.g., via a GUI in computer graphics) is beyond the scope of this work, while we will briefly discuss that, in Section 5.

With the understanding that a tile $\mathbf{t}_i$ can be of a variety of geometric types, following [8, 16], herein, for uniformity, we use the same tile, as in Figure 2, unless otherwise stated. This tile consists of seven trivariate spline functions that will be functionally composed into the different trivariate cages, $\mathcal{T}$. Being a parametric tile, its geometric properties can be modified, as is shown, for example, in Figure 2 (b).

To fully support the proposed conformal tiling with microstructures, of a B-rep model $\mathcal{M}$, this B-rep must support the following operations:

1. **PointInclusion**. Denote by $\mathcal{M}^\circ$ the interior of $\mathcal{M}$. Given a point p, is $p \in \mathcal{M}^\circ$?
2. **PointProjection**. Given a point $p \in \mathcal{M}^\circ$, what is the closest point to p on $\partial\mathcal{M}$, the boundary of $\mathcal{M}$?
3. **BrepIntersection**. Given a second B-rep model $\mathcal{N}$, is $\partial\mathcal{M} \cap \partial\mathcal{N} = \emptyset$?

See Appendix A, for some explanation how these operations can be evaluated for trimmed surfaces based B-reps as well as polygonal B-reps.

With the availability of these operations over a B-rep model, Algorithm 1 portrays the top level process. The input to the whole algorithm is the B-rep model $\mathcal{M}$, a trivariate cage $\mathcal{T}$ that fully encloses $\mathcal{M}$, and a parametric tile $\mathbf{t}$ to populate $\mathcal{T}$ with, following some xyz tile-density prescriptions $k,\ m,\ n$. Finally, input α represents a shape control over the bridging tiles that will be created in Algorithm 4. Figure 3 demonstrates each of the steps of Algorithm 1.

Following [8], we build microstructure $\mathcal{MS}_\mathcal{T}$ inside $\mathcal{T}$ using tile $\mathbf{t}$, in Line 1 of Algorithm 1. Then, in Lines 2 and 3 of the algorithm, we handle two types of tiles:

1. Tiles in $\mathcal{MS}_\mathcal{T}$ that are fully contained in $\mathcal{M}$ are assigned to $\mathcal{MS}_\mathcal{F}$, in Line 2, calling Algorithm 2.
2. New bridging tiles to $\partial\mathcal{M}$ are built between tiles in $\mathcal{MS}_\mathcal{F}$ that are close to $\partial\mathcal{M}$, and $\partial\mathcal{M}$, and accumulated in $\mathcal{MS}_\mathcal{B}$, in Line 3, calling Algorithm 3.

Then, in Line 4 of Algorithm 1, these two sets are merged into the final microstructure, $\mathcal{MS}_\mathcal{M}$.

Algorithm 2 filters out tiles in $\mathcal{MS}_\mathcal{T}$ that are not fully contained in $\mathcal{M}$. Toward this end, we exploit two operations over our B-rep: **BrepIntersection** and **PointInclusion**. Each tile, $\mathbf{t}_i$, is tested, in Line 3 of

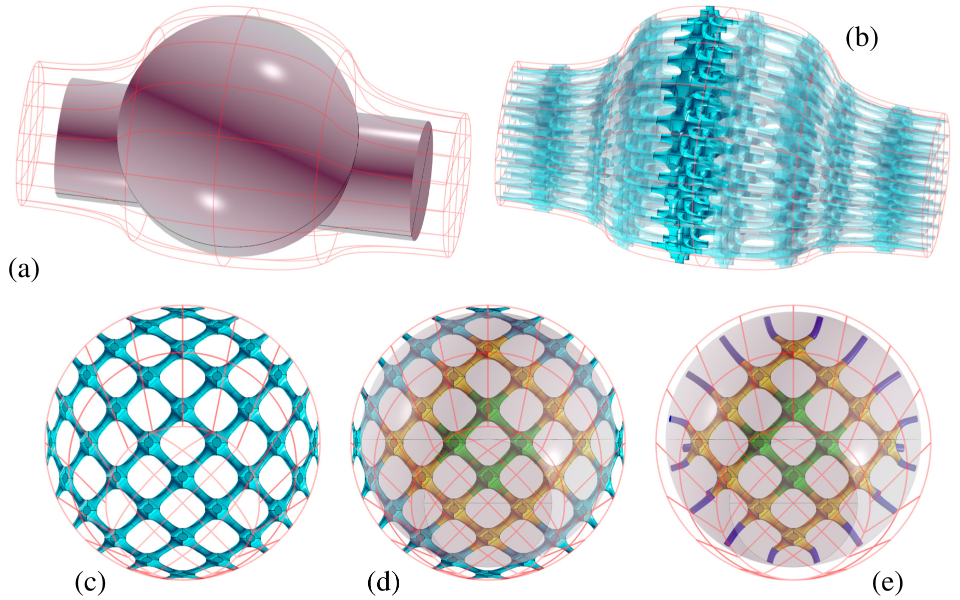

Figure 3: *An example of microstructures tiled in a B-rep trimmed surfaces model. In (a), A B-rep model $\mathcal{M}$ is drawn in gray, with a cage trivariate $\mathcal{T}$ drawn in a red wireframe. In (b), microstructure tiles in $\mathcal{MS}_{\mathcal{T}}$ are drawn in cyan. In (c), we show one layer of the tiles from $\mathcal{MS}_{\mathcal{T}}$ (the opaque tiles in (b)). In (d), the tiles in (c) are filtered, and yellow and green tiles are included in $\mathcal{MS}_{\mathcal{F}}$. In (e), yellow tiles are joined to the boundary model $\partial\mathcal{M}$ using bridging tiles (in blue).*

Algorithm 1 Microstructure tiling algorithm of $\mathcal{M}$.

```
Input:
$\mathcal{M}$: a B-rep model;
$\mathcal{T}(u, v, w)$: a trivariate volumetric cage, containing $\mathcal{M}$;
$\mathbf{t}$: a parametric tile to populate $\mathcal{T}$ (and $\mathcal{M}$) with;
$k$, $m$, $n$: controls over tiling density of $\mathbf{t}$ in $\mathcal{T}$, in $uvw$;
$\alpha$: a shape control over the created bridging tiles;
Output:
$\mathcal{MS}_{\mathcal{M}}$: a microstructure tiling of $\mathcal{M}$, using tile $\mathbf{t}$, and following the 'flow' of $\mathcal{T}$;
Algorithm:
1: $\mathcal{MS}_{\mathcal{T}}$ := a microstructure tiling of $\mathcal{T}$, using tile $\mathbf{t}$ and controls $n$, $m$, $k$, following [8];
2: $\mathcal{MS}_{\mathcal{F}}$ := FilterTilesTo$\mathcal{M}$($\mathcal{M}$, $\mathcal{MS}_{\mathcal{T}}$); // Alg. 2.
3: $\mathcal{MS}_{\mathcal{B}}$ := BridgeTilesTo$\mathcal{M}$($\mathcal{M}$, $\mathcal{MS}_{\mathcal{T}}$, $\mathcal{MS}_{\mathcal{F}}$, $\alpha$); // Alg. 3.
4: $\mathcal{MS}_{\mathcal{M}}$ := $\mathcal{MS}_{\mathcal{F}} \cup \mathcal{MS}_{\mathcal{B}}$;
5: return $\mathcal{MS}_{\mathcal{M}}$;
```

Algorithm 2, for a full containment in $\mathcal{M}$. First, we (via **BrepIntersection**) examine if $\partial\mathbf{t}_i \cap \partial\mathcal{M} \neq \emptyset$, in which case $\mathbf{t}_i$ intersects $\mathcal{M}$ and is purged. Otherwise, if $\mathbf{t}_i$ and $\mathcal{M}$ do not intersect, a single point p on $\mathbf{t}_i$ is selected, only to (via **PointInclusion**) test p for inclusion in $\mathcal{M}$. Only if $\mathbf{t}_i$ passes both tests, it is added to $\mathcal{MS}_{\mathcal{F}}$ as a fully contained tile inside $\mathcal{M}$.

Having the subset of tiles in $\mathcal{MS}_{\mathcal{T}}$ that are in $\mathcal{M}$, as $\mathcal{MS}_{\mathcal{F}}$, in Algorithm 3, a bridging tile to $\partial\mathcal{M}$ is constructed for each tile in $\mathcal{MS}_{\mathcal{F}}$ that is close to $\partial\mathcal{M}$. We start by identifying the tiles in $\mathcal{MS}_{\mathcal{F}}$ that are close to $\partial\mathcal{M}$, in Line 4 of Algorithm 3. Consider a tile $\mathbf{t}_i$ that is fully contained in $\mathcal{M}$, being in $\mathcal{MS}_{\mathcal{F}}$, while $\mathbf{t}_i$ has

Algorithm 2 FilterTilesTo$\mathcal{M}$: Filtering stage to isolate all microstructure tiles that are fully inside $\mathcal{M}$.

Input:
$\mathcal{M}$: a B-rep model;
$\mathcal{MS}_\mathcal{T}$: a microstructure formed using a trivariate that fully contains $\mathcal{M}$;
Output:
$\mathcal{MS}_\mathcal{F}$: a filtered microstructure of $\mathcal{MS}_\mathcal{T}$, with tiles completely inside $\mathcal{M}$;
Algorithm:

```
 1: $\mathcal{MS}_\mathcal{F} := \emptyset$;
 2: for each tile $\mathbf{t}_i \in \mathcal{MS}_\mathcal{T}$ do
 3:     if $\mathbf{t}_i \subset \mathcal{M}$ then
 4:         $\mathcal{MS}_\mathcal{F} := \mathcal{MS}_\mathcal{F} \cup \{\mathbf{t}_i\}$;
 5:     end if
 6: end for
 7: return $\mathcal{MS}_\mathcal{F}$;
```

an adjacent tile $\mathbf{t}_{nbr}$ that is not completely in $\mathcal{M}$. In other words, $\mathbf{t}_{nbr} \notin \mathcal{MS}_\mathcal{F}$.

Detecting a tile $\mathbf{t}_i$ close to $\partial\mathcal{M}$, that shares a boundary surface with $\mathbf{t}_{nbr}$ that is not in $\mathcal{MS}_\mathcal{F}$, we create a bridging tile from $\mathbf{t}_i$ to $\partial\mathcal{M}$. Further, this bridging tile will be created from the face of $\mathbf{t}_i$ that is shared with $\mathbf{t}_{nbr}$. This bridging tile is constructed in Algorithm 4. The function **Neighborhood** in Line 3, Algorithm 3, computes all the immediate neighbors (or adjacent tiles sharing a face) of a given tile in $\mathcal{MS}_\mathcal{T}$, a function that exploits the known topology of all tiles in $\mathcal{MS}_\mathcal{T}$.

Algorithm 3 BridgeTilesTo$\mathcal{M}$: Builds bridging tiles from $\mathcal{MS}_\mathcal{F}$ to the boundary of $\mathcal{M}$.

Input:
$\mathcal{M}$: a B-rep model;
$\mathcal{MS}_\mathcal{T}$: a microstructure of volume $\mathcal{T}$ fully containing $\mathcal{M}$;
$\mathcal{MS}_\mathcal{F}$: a filtered microstructure of $\mathcal{MS}_\mathcal{T}$, completely inside $\mathcal{M}$;
α: a shape control over the created bridging tiles;
Output:
$\mathcal{MS}_\mathcal{B}$: a set of bridging tiles from tiles in $\mathcal{MS}_\mathcal{F}$, to $\partial\mathcal{M}$;
Algorithm:

```
 1: $\mathcal{MS}_\mathcal{B} := \emptyset$;
 2: for each tile $\mathbf{t}_i \in \mathcal{MS}_\mathcal{F}$ do
 3:     for each tile $\mathbf{t}_{nbr} \in$ Neighborhood$(\mathbf{t}_i, \mathcal{MS}_\mathcal{T})$ do        // All tiles adjacent to $\mathbf{t}_i$, in $\mathcal{MS}_\mathcal{T}$.
 4:         if $\mathbf{t}_{nbr} \notin \mathcal{MS}_\mathcal{F}$ then
 5:             // $\mathbf{t}_i$ is fully in $\mathcal{M}$ while a neighbor $\mathbf{t}_{nbr}$ is not. Create a bridging tile from $\mathbf{t}_i$ to $\partial\mathcal{M}$.
 6:             $b$ := BuildBridgeTile$(\mathbf{t}_i, \mathbf{t}_{nbr}, \mathcal{M}, \alpha)$; // Alg. 4.
 7:             if TileVerify$(b)$ then        // Make sure neither deformed nor singular.
 8:                 $\mathcal{MS}_\mathcal{B} := \mathcal{MS}_\mathcal{B} \cup \{b\}$;
 9:             end if
10:         end if
11:     end for
12: end for
13: return $\mathcal{MS}_\mathcal{B}$;
```

Finally, Algorithm 4 presents one approach to creating the bridging tiles. Bridging tiles join the outermost faces of the filtered tiles in $\mathcal{MS}_\mathcal{F}$ to the boundary of $\mathcal{M}$ and enable to maintain the conformality between the microstructures and the macro object. The shared face between $\mathbf{t}_i$ and $\mathbf{t}_{nbr}$ is identified as F, in Line 1, and a central point p_F in F and an outgoing normal are computed, in Lines 2 and 3. In Line 4, we estimate the distance from p_F to the boundary (exploiting the B-rep **PointProjection** operation) and employ the shape control α to compute point q, only to project q on $\mathcal{M}$ to find the closest location to q on $\mathcal{M}$, as location r. p_F, q, and r are employed as the control points of a quadratic Bézier axis curve, C, through which a sweep trivariate bridging tile is derived, sweeping F through C, in Lines 8 and 9, that is G^1 continuous to $\mathbf{t}_i$.

Algorithm 4 shows that the bridging tiles are independently constructed by joining F's to the closest boundary surfaces of $\mathcal{M}$. Similar to the bridging tiles constructed in [9], the bridging tiles herein can also suffer from collisions between tiles. As a post-process of constructing the bridging tiles, we detect possible collisions between bridging tiles by computing the smallest distance between the axis curves. Two bridging tiles are declared too close if the sum of the radii of the shared faces is larger than the minimum distance between the axis curves. We attempt to resolve such collisions, by moving away the end point of the axis curves (r in Algorithm 4) on $\partial\mathcal{M}$. When the distance d in Line 5 in Algorithm 4 is too small, (e.g., d becomes almost zero when the filtered tile nearly contacts the boundary of $\mathcal{M}$, the resulting bridging tile is almost flat. The bridging tile, in addition, can be highly distorted, or even self-intersecting, if $\vec{n}_F$ deviates too much from the surface normal of $\partial\mathcal{M}$ at point r. See **TileVerify** in Algorithm 3, Line 7 - we do not include the bridging tiles in these extreme cases.

Algorithm 4 BuildBridgeTile: Builds a bridging tile from $\mathbf{t}_i$ along shared boundary with tile $\mathbf{t}_{nbr}$, to $\partial\mathcal{M}$.

```
Input:
$\mathcal{M}$: a B-rep model;
$\mathbf{t}_i$: a tile fully inside $\mathcal{M}$;
$\mathbf{t}_{nbr}$: a tile adjacent to $\mathbf{t}_i$, and intersecting/out-of $\mathcal{M}$;
$\alpha$: a shape control over the created bridging tile;
Output:
b: a bridging tile from $\mathbf{t}_i$, from its adjacent face with tile $\mathbf{t}_{nbr}$, to $\partial\mathcal{M}$;
Algorithm:
 1: $F$ := shared face of $\mathbf{t}_i$ and $\mathbf{t}_{nbr}$;
 2: $p_F$ := center location of $F$;
 3: $\vec{n}_F$ := outgoing unit normal of $F$ at $p_F$;
 4: $p_{\mathcal{M}}$ := closest point on $\partial\mathcal{M}$ to $p_F$, calculated via B-rep PointProjection;
 5: $d := dist(p_F, p_{\mathcal{M}})$;
 6: $q := p_F + \alpha d \vec{n}_F$;
 7: $r$ := closest point on $\partial\mathcal{M}$ to $q$, calculated via B-rep PointProjection;
 8: $C$ := quadratic Bézier curve, using $p_F$, $q$, and $r$ as its control points.
 9: b := sweep volume of surface $F$ along curve $C$;
10: return b;
```

4 Results

Unless otherwise stated, tiles used in this section are trivariates, typically tricubic. Clearly, tiles consisting of surfaces will be faster to process and more so for curves, etc. We will demonstrate that, in this section as well. We start, in Figure 4, with a simple trimmed surfaces based B-rep model that is a union of a sphere and a cylinder. The original cage trivariates and the original set of micro-elements $\mathcal{MS}_{\mathcal{T}}$, are presented in Figure 4 (i), while the final set of conforming micro-element tiles is presented in (ii) to (iv). The filtered (interior to $\mathcal{M}$) tiles, $\mathcal{MS}_{\mathcal{F}}$, are drawn in green and yellow, with the yellow tiles are those from which bridging tiles to $\partial\mathcal{M}$ are formed. Finally, the bridging tiles themselves, $\mathcal{MS}_{\mathcal{B}}$, are painted in blue. A low-resolution (recall k, m, n in Algorithm 1) tiling is presented in (a) whereas (b) presents a higher resolution version of (a) in all three axis, using the same cage trivariate. Finally, in (c), a high resolution tiling is shown, with a different, radial, cage trivariate function, $\mathcal{T}$.

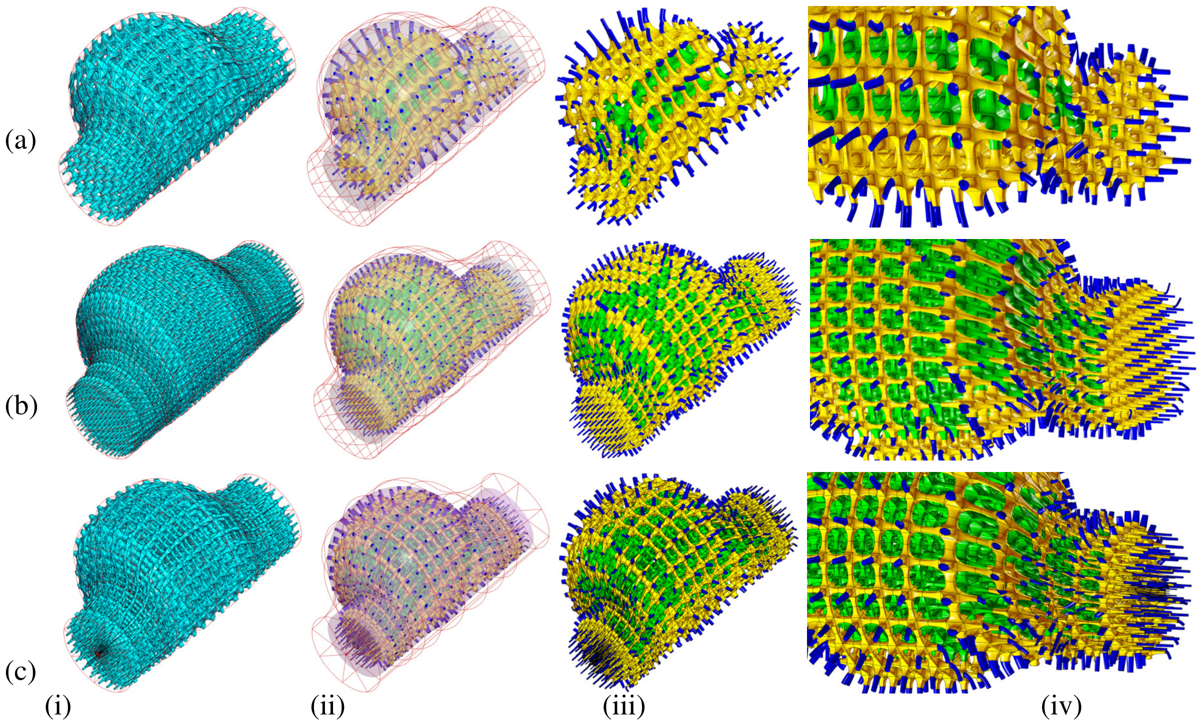

Figure 4: *Three examples of microstructures, tiled in a B-rep trimmed surfaces model. (a) shows a fairly low resolution microstructure whereas (b) presents twice higher the resolution compared to (a), in all three (u, v, w) axes. In (c), a high resolution version, similar to (b), is presented but with a different cage trivariate field, being radial. The columns, left to right, depict (i) the original cage trivariate with the full set of tiles, $\mathcal{MS}_{\mathcal{T}}$, in cyan, (ii) the model with the final set of tiles, $\mathcal{MS}_{\mathcal{M}}$, in the B-rep model (transparent), and the cage (red wireframe), (iii) the final microstructure $\mathcal{MS}_{\mathcal{M}}$, and (iv) a zoom-in on a portion of the microstructure in (iii). The filtered (interior to $\mathcal{M}$) tiles, $\mathcal{MS}_{\mathcal{F}}$, are drawn in green and yellow, with the yellow tiles are those from which bridging tiles to $\partial\mathcal{M}$ are formed. Finally, the bridging tiles themselves, $\mathcal{MS}_{\mathcal{B}}$, are painted in blue.*

The polygonal model of a bone [1] has been fitted with three trivariate cages, shown in (a) to (c) in Figure 5. Figure 6 presents the placement of conforming microstructures into another polygonal B-rep model - the Stanford Bunny [2] [3]. The tiling density (Recall k, m, n in Algorithm 1) is similar in all three cases. Three different cages are presented, two of which, in (b) and (c), have a C^{-1} discontinuity that allows the cages to split into two, near the ears of the Bunny. The trivariate cage in (c) is the tightest among the three, with respect to the bunny, and the result is clearly visible near the ears, as shown in the zoom-in, in (iii). In both the (a) and (b) cases, the ears can not be completely populated with tiles, and further, some tiles in the ear are floating and are disconnected from the rest of the microstructure.

Finally, in Figure 7, a B-rep model of a duck, formed out of trimmed surfaces is presented. Two trivariate cages were created for this model and the differences are mostly visible in the area of the head. The shape

[1]from https://www.turbosquid.com/3d-models/3d-cartoon-bone-1614756

[2]see http://graphics.stanford.edu/data/3Dscanrep

[3]downloaded from https://www.thingiverse.com/thing:3731/files

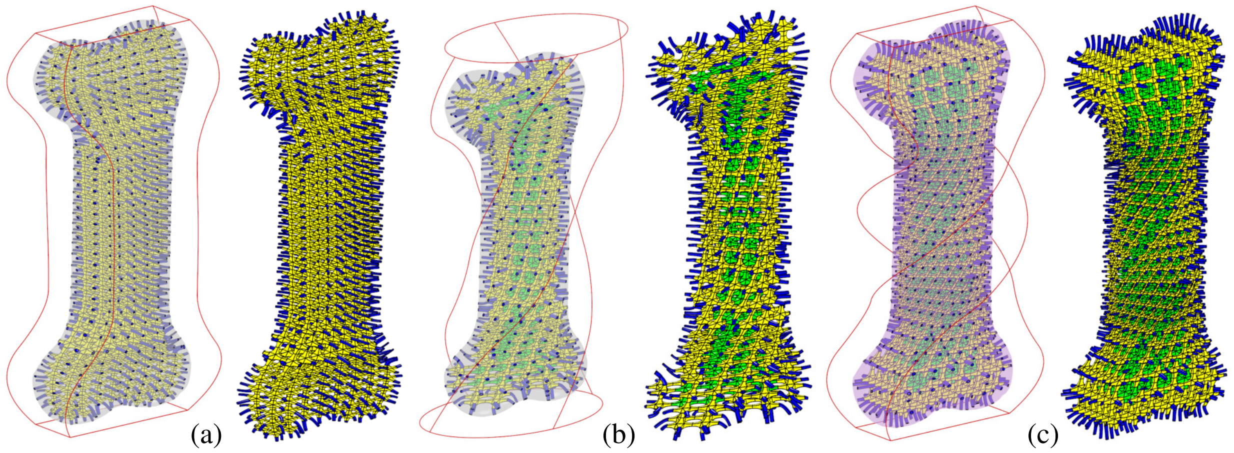

Figure 5: *A polygonal B-rep model $\mathcal{M}$ of a bone (4252 polygons) tiled with microstructures. Three different cages are shown, from (a) to (c), each of which with two images - the cage in red wireframe and the bone model transparent on the left and the final microstructure $\mathcal{MS}_{\mathcal{M}}$ on the right. The filtered (interior to $\mathcal{M}$) tiles, $\mathcal{MS}_{\mathcal{F}}$, are drawn in green and yellow, with the yellow tiles are those from which bridging tiles to $\partial\mathcal{M}$ are formed. Finally, the bridging tiles themselves, $\mathcal{MS}_{\mathcal{B}}$, are painted in blue.*

and the size of the individual microstructure depend on the Jacobian of the cage where the microstructure occupies, which is determined by geometry, parametrization of the trivariate cage, and the prescribed tiling parameters k, m, n. When the cage contains regions close to being singular, as shown near the tail of the duck in Figure 7, we can adjust k, m, n to yield less distorted tiles in near-singular areas. Alternatively, bifurcation tiles can be employed as is done in [16].

The most expensive step in these computations, especially for trimmed surfaces based B-rep, is the intersection test. Table 1 presents the computation time for some of these examples. All experimental results are measured on an Intel Core i7-7700K 4.2GHz PC with 32 GB RAM and eight cores. Construction (Line 1 in Alg. 1) and filtering (Alg. 2) computations are parallelized using eight threads, for each tile $\mathbf{t_i}$, whereas, bridging tiles (Alg. 3 and Alg. 4) are computed on a single thread, due to the collision detection between different bridging tiles.

Some results, as seeing in Table 1, should be discussed. The tiles' construction time of Figure 5 (b) is far longer compared to Figure 5 (a) and (c) due to the higher order of the cage trivariate, which is linear in u and v in (a) and (c) and cubic in (b). The same results are also observed in the Bunny model (Figure 6), in which the construction time is slower in (a) due to the higher order of the cage trivariate in (a). Filtering of polygonal B-reps is much faster due to the usage of BVH, compared with filtering of spline based B-reps that requires the computation of surface-surface intersections. Further, while we mostly employed trivariate tiles in these examples, tiles consisting of surfaces are faster to process and more so for curves based tiles. Table 1 also shows the functional composition computation times for the same tile shapes but with surfaces, which are clearly faster, mostly due to the lower dimensions of the composed results.

5 Extensions

The presented approach can be extended in several ways. To begin with, the input enclosing trivariate $\mathcal{T}$ can be prescribed directly but it can also be defined by fitting $\mathcal{T}$ to any input field (or tensor). Such a field can come, for example, from stress analysis over the geometry that will identify the principle stress directions that must be re-enforced. Further, the same analysis can also control the local shapes or material content

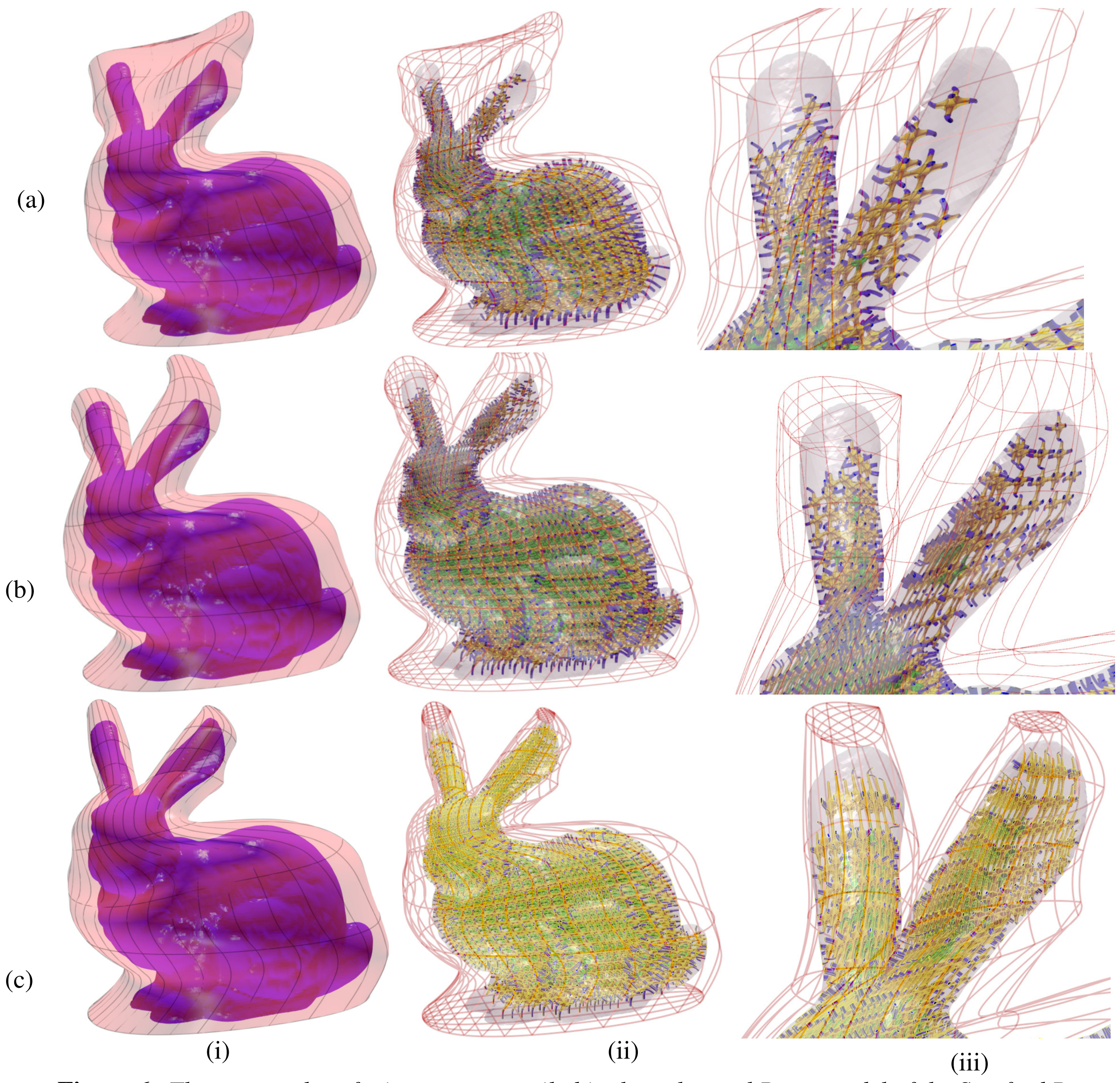

Figure 6: *Three examples of microstructures tiled in the polygonal B-rep model of the Stanford Bunny (66848 polygons). (a) to (c) show three different trivariate cages, with (c) being the tightest. The cages are depicted transparent in (i) and using wire-frames in (ii) and (iii). Note the trivariate cages in (b) and (c) have a C^{-1} discontinuity that allows its split near the ears of the bunny. The tiling density in all three examples is similar.*

of individual tiles. For example, thin tiles in low stress (or minimal heat transfer) zones and thick tiles in locations where the identified stresses are significant (or the required heat flux is considerable).

The geometry of a tile in the final result is affected not only by the shape of the tile in the domain of the cage trivariate $\mathcal{T}$, but also by the local Jacobian of $\mathcal{T}$. While we have little control over (the Jacobian of) $\mathcal{T}$, we have full control over the shape of individual tiles, as they are parametric. As long as the geometry preserves the desired continuity, any global specification can be employed here, e.g., direct prescription by the end user as a function of Euclidean coordinates in R^3 or trivariate parametric coefficients uvw, due to stress or heat transfer analyses, etc.

(a)

(b)

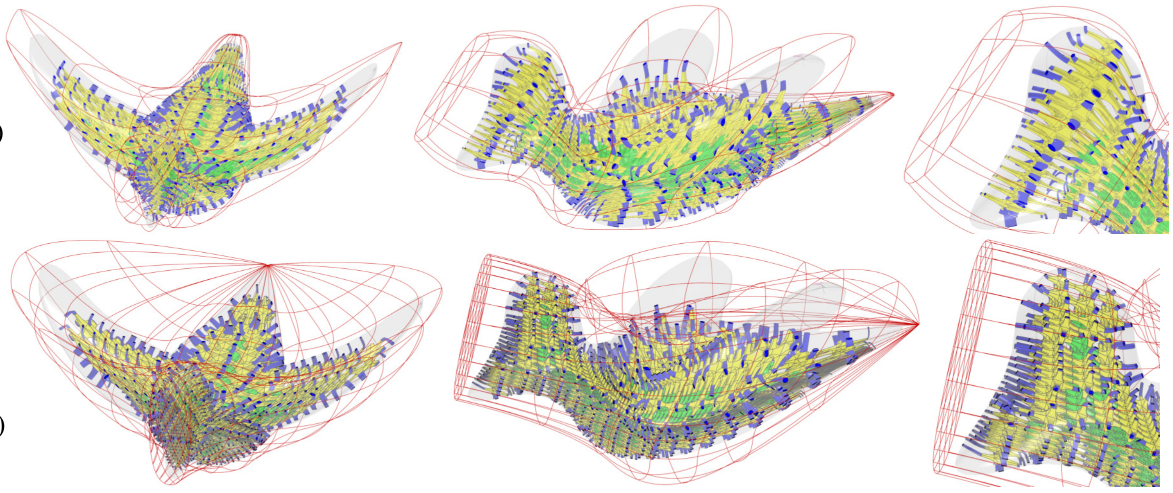

Figure 7: *A trimmed surfaces based B-rep model of a duck tiled with microstructures. Two different cages are shown, in (a) and (b), each of which with three images. The cage trivariate is shown in a red wireframe, the B-rep model is presented transparent, and the final microstructure in yellow, green (for MS_F) and blue (MS_B). The zoom-in, on the area of the head, on the right, clearly depicts the differences in the outcome.*

Figure	Time (secs.)				No. of tiles		
	Construction (eight threads)		Filtering (eight threads)	Bridging (one thread)	$\lvert\mathcal{MS_T}\rvert$	$\lvert\mathcal{MS_F}\rvert$	$\lvert\mathcal{MS_B}\rvert$
	Trivariate tiles	Surface tiles					
4 (a)	456.484	1.609	258.281	6.439	576	188	272
4 (b)	1854.469	5.937	946.63	24.26	2304	916	770
4 (c)	214.735	2.921	739.912	23.05	2304	1468	692
5 (a)	10.344	1.047	1.93	7.675	1480	531	728
5 (b)	1534.844	5.063	1.78	5.314	1920	387	510
5 (c)	12.281	1.641	3.565	10.734	2240	939	836
6 (a)	6963.937	22.328	50.667	261.711	8448	1706	1519
6 (b)	1307.093	13.141	58.611	352.709	10368	2315	2066
6 (c)	1054.328	12.906	63.94	412.034	10368	2786	2407
7 (a)	237.5	3.75	10400.717	32.138	2541	889	852
7 (b)	328.25	4.516	10783.349	46.006	3549	1287	983

Table 1: *Statistics of the test cases shown in this section. $\lvert\mathcal{MS_T}\rvert$ and $\lvert\mathcal{MS_F}\rvert$ represent the number of unit trivariate tiles before and after filtering, and $\lvert\mathcal{MS_B}\rvert$ represents the number of bridging tiles connected to $\mathcal{M}$. The number of polygons in the Bunny model (Figure 6) is 66848, and 4252 in the bone model (Figure 5), respectively. While we employ trivariate tiles throughout, the construction times for surface-based tiles are much faster and are shown for comparison.*

Figure 8 presents a few examples of the same model $\mathcal{M}$ and the same cage trivariate, as in Figure 4, while the thicknesses of individual tiles are locally modified, herein following some globally specified functions in R^3.

Clearly, any property in a tile can be similarly controlled and modified and the thickness of individual tiles is merely one example. This includes the geometry, topology and even material content of a tile. This, while typically the proper connectivity to neighboring tiles (and $\partial\mathcal{M}$) is maintained. A careful reexamination

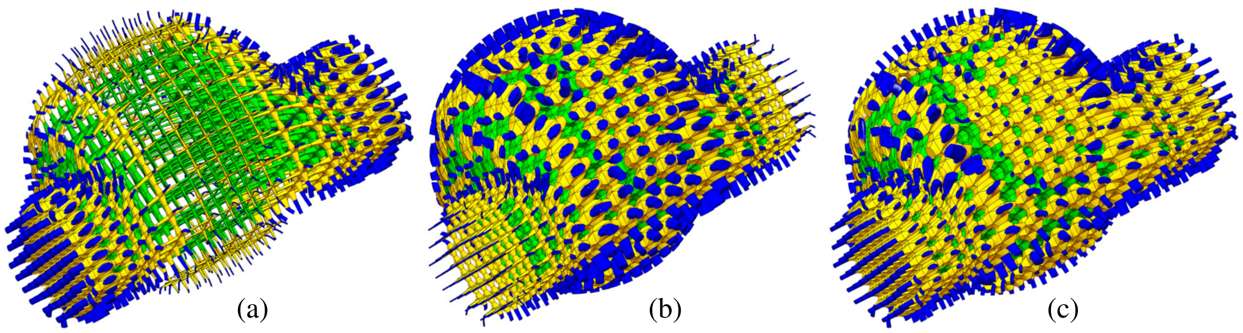

Figure 8: *Several examples of variable thickness tiles in a B-rep model are shown. In (a), the tiles are thinned at the center, in the central sphere zone, in (b), the tiles are thinned toward the bases of the cylinder, and in (c), the thicknesses are modulated to follow some sine functions. Compare with Figure 4.*

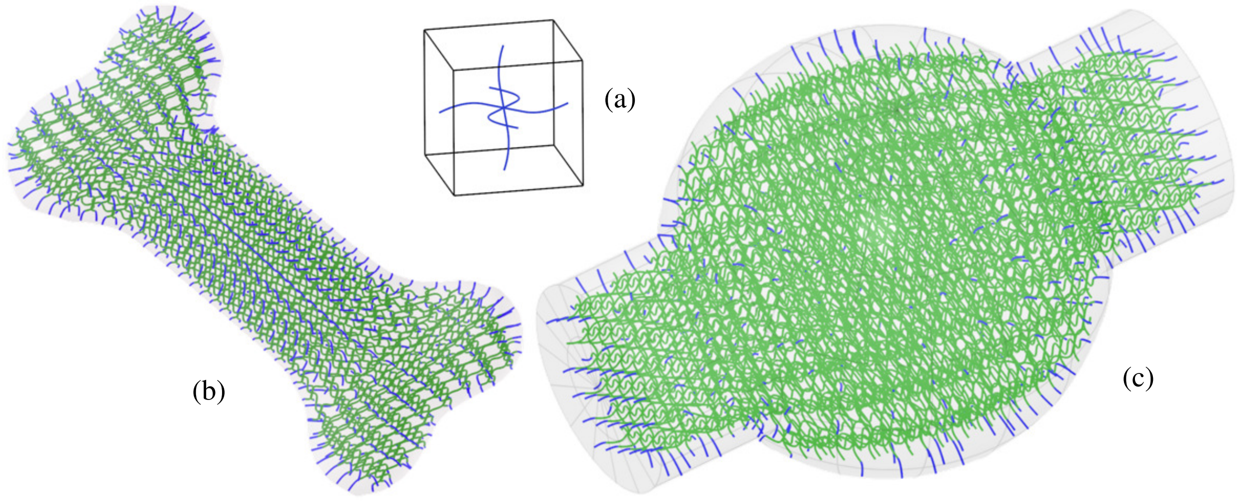

Figure 9: *The tile in (a), consisting of three periodic freeform curves in a unit cube (cube is not part of the tile), is used to tile the B-rep polygonal bone model from Figure 5, in (b), and the B-rep trimmed surfaces based model, similar to the model in Figure 4, in (c). Interior tiles are drawn in green and bridging curves are in blue.*

of Figure 8 will reveal that a single tile can possess different properties on its different (six) boundaries, herein different thicknesses, ensuring a C^0 continuity.

Further, following [16], the geometry in these tiles can consist of univariate curves and polylines, polygonal meshes, freeform (trimmed) surfaces and solids, and (trimmed) trivariates. Figure 9 demonstrates this ability for tiles formed out of freeform curves. The three operations of **PointInclusion**, **PointProjection**, and **BrepIntersection** must be supported between the B-rep model and the specific type of geometry in the tile. Herein, for a curves-based tile and a B-rep model, a test for curve-model intersection must be supported as well. For curve-based tiles, the computation is even faster than trivariate- or surface-based tiles. The curves tiles in Figure 9 (b) were computed in 0.188 Sec, whereas those in Figure 9 (c) were computed in 0.25 Sec.

Finally, Figure 10 shows the duck and the Bunny 3D printed. The duck was printed as a skeletal microstructure whereas the Bunny was printed (in two sizes) in a translucent body. Some surfaces of the microstructure of the Bunny were painted with blue dots, demonstrating the ability to prescribe heterogeneous

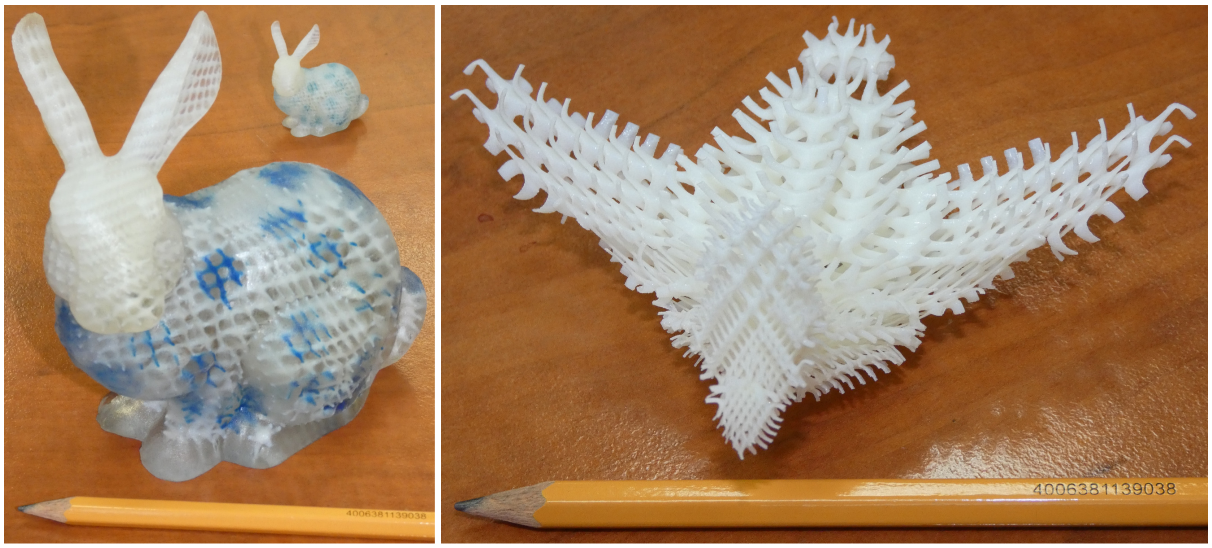

Figure 10: *3D printed versions of the Bunny (in two sizes) from Figure 6 and the duck from Figure 7. The Bunny was printed in a translucent body while for the duck, only the microstructure has been 3D printed. Printed on a J55 printer of Stratasys.*

properties in tiles in the microstructure. Following [16], the heterogeneity of the tiles in the microstructure can be reflected not only in their material properties but also in their topological and/or geometric shape. All models were printed on a J55 printer [4] of Stratasys, with water-soluble supporting structure. A single polygonal file (in STL or OBJ file format) has been created for all the B-rep tiles in the microstructure, only to be sent to the printer.

6 Conclusions and Future Work

In this work, we have presented a constructive approach to populating B-rep models, polygonal or spline-based, with microstructure tiles, along an arbitrarily specified field. This aim has been accomplished using a user-defined trivariate cage that encompasses the B-rep model. One potential benefit to be explored is the exploitation of the presented approach in placement of tiles along desired directions, like stress tensors or gradients of heat.

While tiles in this work were composed of either curves or surfaces/trivariate B-splines, nothing in this work prevents one from employing a single tile with mixed geometry type, including other geometric types, such as polygonal meshes and polylines. Yet, considering such a tile as a whole, would still entail the support of the three operations over B-reps, as discussed in Section 3.

There is also a clear room for further development. The presented scheme assumes the bridges are built from faces in tiles that are close to the boundary of $\mathcal{M}$ (Face F in Algorithm 4). Face F is typically almost planar but not exactly planar. Further, the location where the bridge contacts $\mathcal{M}$ is typically also non planar. Hence, the bridge typically contacts $\mathcal{M}$ in a non tangential way and can penetrate the boundary, instead of being precisely tangent to it. While in a minute amount, a better bridging scheme to ensure proper tangency contact might be desired. For example, if $\partial\mathcal{M}$ is a shell of some ϵ thickness and that ϵ is larger than the non-planarity of F, Boolean operations can be employed between the shell $\partial\mathcal{M}$ and $\mathcal{MS}_{\mathcal{B}}$. Further, $\mathcal{M}$ can have C^1 discontinuities (e.g., along intersection curves of two surfaces), which will require either moving the contact point of the bridge on $\mathcal{M}$ (r in Algorithm 4) away from the discontinuity, or mimicking the

[4]https://www.stratasys.com/en/3d-printers/printer-catalog/polyjet/j55-prime

discontinuity in the bridge.

In addition, the cage $\mathcal{T}$ and tiling parameters such as tiling density should be prescribed by the end user, in the proposed method. Providing a cage and tiling parameters that are satisfying physical boundary conditions and/or additive manufacturing constraints is another future work of the proposed method, incorporating analysis and optimization tools into the design loop.

The topology of a B-rep model can be arbitrary complex and herein we only employed a single caging trivariate for the entire model $\mathcal{M}$. $\mathcal{M}$ can have arbitrary number of handles coming out or have a large genus. A generalized cage based approach that is similar to polycubes [27] and/or trimmed trivariate V-reps [9], might be of value here, for a complex model $\mathcal{M}$, and needs to be further explored.

Acknowledgements

This project has received funding in part from the European Union Horizon 2020 research and innovation programme, under grant agreement No 862025 and in part by the ISRAEL SCIENCE FOUNDATION (grant No. 597/18).

References

[1] T. Akenine-Moller, E. Haines, and N. Hoffman. *Real-time Rendering*. AK Peters/crc Press, 2019.

[2] P. Antolin, A. Buffa, E. Cohen, J. F. Dannenhoffer, G. Elber, S. Elgeti, R. Haimes, and R. Riesenfeld. "Optimizing Micro-Tiles in Micro-Structures as a Design Paradigm." *Computer-Aided Design*, vol. 115, 2019, pp. 23–33. https://www.sciencedirect.com/science/article/pii/S0010448519301939.

[3] A. Aremu, J. Brennan-Craddock, A. Panesar, I. Ashcroft, R. Hague, R. Wildman, and C. Tuck. "A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing." *Additive Manufacturing*, vol. 13, 2017, pp. 1–13. https://www.sciencedirect.com/science/article/pii/S2214860416302810.

[4] R. Arora, A. Jacobson, T. Langlois, Y. Huang, C. Mueller, W. Matusik, A. Shamir, K. Singh, and D. Levin. "Volumetric Michell Trusses for Parametric Design & Fabrication." *Proceedings of the ACM Symposium on Computational Fabrication*. June 2019. pp. 1–13.

[5] M. P. Bendsoe and O. Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer, 2004.

[6] L. Chen, J. Huang, H. Sun, and H. Bao. "Technical Section: Cage-Based Deformation Transfer." *Comput. Graph.*, vol. 34, no. 2, Apr 2010, pp. 107–118.

[7] F. Conde-Rodríguez, Á.-L. García-Fernández, and J.-C. Torres. "Modeling the Internal Architecture of Composites." *Computer-Aided Design*, vol. 129, 2020, p. 102930. https://www.sciencedirect.com/science/article/pii/S0010448520301238.

[8] G. Elber. "Precise construction of micro-structures and porous geometry via functional composition." *Proceedings of the 9th International Conference on Mathematical Methods for Curves and Surfaces*. 2016. pp. 108–125.

[9] Q. Y. Hong and G. Elber. "Conformal Microstructure Synthesis in Trimmed Trivariate Based V-Reps." *Computer-Aided Design*, vol. 140, 2021, p. 103085. https://www.sciencedirect.com/science/article/pii/S0010448521000968.

[10] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. "Harmonic Coordinates for Character Articulation." *ACM Trans. Graph.*, vol. 26, no. 3, Jul 2007, pp. 7–es.

[11] T. Ju, S. Schaefer, and J. Warren. "Mean Value Coordinates for Closed Triangular Meshes." *ACM Trans. Graph.*, vol. 24, no. 3, Jul 2005, pp. 561–566.

[12] S. Kambampati, C. Jauregui, K. Museth, and H. A. Kim. “Geometry Design Using Function Representation on a Sparse Hierarchical Data Structure.” *Computer-Aided Design*, vol. 133, 2021, p. 102989. https://www.sciencedirect.com/science/article/pii/S0010448520301822.

[13] F. Klein. “A New Approach to Point Membership Classification in B-rep Solids.” *the 13th IMA International Conference on Mathematics of Surfaces XIII*. 2009. pp. 235–250.

[14] L. Leblanc, J. Houle, and P. Poulin. “Modeling with blocks.” *The Visual Computer*, vol. 27, no. 6, Apr 2011, p. 555. https://doi.org/10.1007/s00371-011-0589-4.

[15] Y. Liu, G. Zheng, N. Letov, and Y. F. Zhao. “A Survey of Modeling and Optimization Methods for Multi-Scale Heterogeneous Lattice Structures.” *The International Journal of Advanced Manufacturing Technology*, vol. 143, no. 4, 2021, p. 040803. https://doi.org/10.1115/1.4047917.

[16] F. Massarwi, J. Machchhar, P. Antolin, and G. Elber. “Hierarchical, random and bifurcation tiling with heterogeneity in micro-structures construction via functional composition.” *Computer Aided Design*, vol. 102, 2018, pp. 148–159.

[17] A. Medeiros e Sá, V. M. Mello, K. Rodriguez Echavarria, and D. Covill. “Adaptive voids.” *The Visual Computer*, vol. 31, no. 6, Jun 2015, pp. 799–808.

[18] A. Nazir, K. M. Abate, A. Kumar, and J.-Y. Jeng. “A state-of-the-art review on types, design, optimization, and additive manufacturing of cellular structures.” *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 9, 2019, pp. 3489–3510. https://doi.org/10.1007/s00170-019-04085-3.

[19] J. Nieto and T. Susin. *Cage Based Deformations: A Survey*. Springer, 01 2013. vol. 7. pp. 75–99.

[20] A. Pasko, O. Fryazinov, T. Vilbrandt, P.-A. Fayolle, and V. Adzhiev. “Procedural function-based modelling of volumetric microstructures.” *Graphical Models*, vol. 73, no. 5, 2011, pp. 165 – 181.

[21] T. W. Sederberg and S. R. Parry. “Free-Form Deformation of Solid Geometric Models.” vol. 20, no. 4, aug 1986, p. 151–160.

[22] M. Sitharam, J. Youngquist, M. Nolan, and J. Peters. “Corner-sharing tetrahedra for modeling micro-structure.” *Computer-Aided Design*, vol. 114, 2019, pp. 164–178. https://www.sciencedirect.com/science/article/pii/S0010448519301848.

[23] F. Tamburrino, S. Graziosi, and M. Bordegoni. “The Design Process of Additively Manufactured Mesoscale Lattice Structures: A Review.” *Journal of Computing and Information Science in Engineering*, vol. 18, no. 4, 2018, p. 040801. https://doi.org/10.1115/1.4040131.

[24] Y. Tang, G. Dong, and Y. Zhao. “A hybrid geometric modeling method for lattice structures fabricated by additive manufacturing.” *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 9, 06 2019, pp. 4011–4030.

[25] J.-M. Thiery, P. Memari, and T. Boubekeur. “Mean Value Coordinates for Quad Cages in 3D.” *ACM Trans. Graph.*, vol. 37, no. 6, dec 2018.

[26] B. van Sosin, D. Rodin, H. Sliusarenko, M. Bartoň, and G. Elber. “The Construction of Conforming-to-shape Truss Lattice Structures via 3D Sphere Packing.” *Computer-Aided Design*, vol. 132, 2021, p. 102962. https://www.sciencedirect.com/science/article/pii/S001044852030155X.

[27] H. Wang, Y. He, X. Li, X. Gu, and H. Qin. “Polycube splines.” *Proceedings of the 2007 ACM symposium on Solid and physical modeling*. June 2007. pp. 241–251.

[28] J. Wu, W. Wang, and X. Gao. “Design and Optimization of Conforming Lattice Structures.” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, 2021, pp. 43–56.

[29] B. Zhu, M. Skouras, D. Chen, and W. Matusik. “Two-Scale Topology Optimization with Microstructures.” *ACM Trans. Graph.*, vol. 36, no. 4, 2017. https://doi.org/10.1145/3072959.3095815.

A. B-rep queries

In this appendix, we briefly discuss how are the three key operations of **PointInclusion**, **PointProjection**, and **BrepIntersection**, evaluated, over trimmed surfaces based B-reps as well as polygonal B-reps. In both cases, we assume a water-tight model.

Having $\mathcal{M}$ as a trimmed surfaces based solids, the **PointInclusion** test for point p can be reduced to shooting a ray from p until it is outside the bounding box of $\mathcal{M}$ and counting the number of intersections - odd if $p \in \mathcal{M}$ and even if $p \notin \mathcal{M}$. This ray-shooting approach can be unstable if cracks or black holes exist along the trimmed area and an alternative and more robust approach can be used, that employs winding numbers - see [13].

The **PointProjection** query can be reduced to the closest point on $\mathcal{M}$ to p, which means the minimum between the closest point-surface tests against the trimmed surfaces in $\mathcal{M}$, closest point-curve tests against the trimming curves in $\mathcal{M}$, and closest point-point test against the intersection locations of the trimming curves. By assuming that the trimmed surfaces and trimming curves are C^1, the minimal distance queries could be reduced to algebraic constraints and solved.

The final **BrepIntersection** test can be resolved by using Boolean operation computation where we only seek to find if the given two B-reps intersect or not.

To accelerate B-rep queries on a polygonal mesh $\mathcal{M}$, we start by building a Bounding Volume Hierarchy (BVH) structure [1] for $\mathcal{M}$. An internal node in the BVH is an axis-aligned bounding box (AABB), and the leaf node contains one polygon.

For the **PointInclusion** test of a point p with respect to the polygonal mesh $\mathcal{M}$, we take the ray-shooting approach which is similar to **PointInclusion** test of the trimmed surface based model. However, the ray-shooting test is now executed while traversing the hierarchy of the BVH; We test the intersection between the ray and one AABB node on BVH structure. If the ray does not intersect the current AABB node, we do not investigate the subtree under the box anymore; if the ray does intersect with the node, then we test further for the intersection between the ray and all the children AABB's, of the current AABB.

For computing the closest point on the polygonal model to an arbitrary query point (the **PointProjection** test), we traverse the BVH by comparing the distance between AABB and the query point. After the traversal algorithm reaches a leaf node of BVH, we do point-projection to the polygon and get the closest point.

For computing the intersection between a polygonal model and a given tile surface (as part of the B-rep tile), we first make AABB for the surface and search for intersecting leaf nodes of the BVH of $\mathcal{M}$, via an hierarchical traversal, of the BVH. Internal nodes of the BVH that do not intersect with the AABB of the surface are excluded, and hence we can significantly accelerate the algorithm. We complete the algorithm by examining for intersections between the identified polygons (leaf nodes of the BVH) and the surface.