Implicit Functionally Graded Conforming Microstructures

Q Youn Hong, Gershon Elber, Myung-Soo Kim

Abstract

The tensor product parametric representations are the most commonly used representation in geometric modeling. Yet, other representations have advantages in certain aspects, and in this work, we focus on employing implicit representations in the construction of microstructures. An implicit function, either functionally precise, or spline trivariate-based, is used to populate a macro-shape trivariate parametric form, and construct a conforming microstructure. Either the implicit tile or the macro-shape can be functionally graded or be heterogeneous, carrying graded properties such as material, translucency, or color alongside the geometry. Further, the implicit tiles can be parametrized and hence their geometry can vary across the macro-shape. The representation is locally precise and we demonstrate that in a slicing process that employs no (piecewise-linear) approximation. Finally, we demonstrate this framework on several 3D printed heterogeneous models.

Keywords: Implicit representations; triply periodic minimal surfaces (TPMS); functionally graded materials (FGM); volumetric trivariate splines; precise slicing for additive manufacturing; efficient non-linear inversions

1. Introduction

With the advent of additive manufacturing technology, also known as 3D printing, there have been dramatic changes in the modeling of 3D solid objects for physical fabrication in reality [1, 23, 31, 37, 43]. A main challenge in 3D modeling research and development nowadays is how to fill the interior of 3D solids using inhomogeneous materials, often with highly complex microstructures for supporting various mechanical, structural, and biomedical properties [13, 15, 26, 30, 34, 41, 44].

Certain geometric features play a crucial role in meeting some of often mutually-contradicting multiple design goals by imitating the generation rules commonly found in nature [17, 37, 43]. TPMS (Triply Periodic Minimal Surfaces) has attracted considerable research attention in the microstructure generation due to the periodicity of these minimal surfaces, and closed implicit forms that are based on sine and cosine functions [14, 20, 21, 22, 25, 45, 46, 48]. Being represented in non-algebraic transcendental implicit equations, the TPMS-based surfaces and volumes are typically approximated using piecewise linear, bilinear, and trilinear elements, which introduces relatively large approximation errors compared with the level of precisions provided/required by the modern 3D printers and the 3D printing applications.

In this work, we strive to support highly precise 3D printing of functionally graded materials (FGM) [3, 4, 32]. The resolution of state-of-the-art 3D printing is in the tens of microns and even below microns [1], and the goal in this work is to fully support those resolutions and beyond.

In its basic form, trivariate $\mathcal{T}(u,v,w) = (\mathcal{T}_x, \mathcal{T}_y, \mathcal{T}_z) : D \subset I\!R^3 \rightarrow I\!R^3$ may be considered as a regular freeform deformation of the 3D volume. Moreover, the uvw-space of D can be initially populated with uniform tiles (aligned in a volumetric 3D axis-parallel grid), where each tile is defined by a periodic implicit function and thus (smoothly) connected with adjacent tiles in each direction. In the 3D printing stage of the algorithm, we consider the plane-volume intersection, known as the slicing of $\mathcal{T}(u,v,w)$, and its embedded implicit tiles, against each slicing plane $z = z_{slice}$, one slice at a time, in an ascending order.

To ensure that the computation is efficient, we employ numerical tracing while preserving the accuracy. For example, consider the precision of curve tracing [35]. On the other hand, a piecewise-linear mesh approximation of an implicit surface introduces typically a much larger error, employing the marching cubes algorithm [24] or similar. Further, the non-linear mapping of the piecewise-linear approximation of the tile through $\mathcal{T}(u,v,w)$, hence after denoted the macro-shape, might enlarge those approximation errors, in the xyz-space. Some previous methods such as Yoo [46] constructed the slicing result as an image on the plane $z = z_{slice}$. Nevertheless, it is unclear how they can possibly meet the goal of the fabrication precision required for a specific 3D printer. To answer this fundamental question on the precision of manufacturing, we propose a scan conversion algorithm that can meet the goal of arbitrary precision, up to the computational machine precision.

A main technical challenge here is how to guarantee the local and global invertibility of the freeform trivariate mapping $\mathcal{T}(u,v,w)$. In the context of our microstructure application at hand, and by construction, we assume that the Jacobian of $\mathcal{T}(u,v,w)$ never vanishes and the trivariate volume $\mathcal{T}(u,v,w)$ has no global self-intersection. Thus not only the local invertibility is guaranteed but the solution is also unique [10, 11, 16].

Once the solution (u,v,w) is computed, the trivariate scalar function $\mathcal{I}(u,v,w)$ that defines the relevant implicit

[1] https://www.nano-di.com/resources/blog/2019-3d-printing-tolerances-considerations-for-high-frequency-electronics

surface tile in $\mathcal{T}$, $\mathcal{I}(u,v,w) = 0$, is evaluated. Typically, the condition $\mathcal{I}(u,v,w) > 0$ (or $\mathcal{I}(u,v,w) < 0$) implies that the point (u,v,w) is located in the interior (or in the exterior) of the tile under consideration. By enhancing the scalar function $\mathcal{I}(u,v,w)$ into a multi-valued vector function, we can also represent graded materials and properties in each tile and hence in the entire microstructure. Using the same trivariate mapping $\mathcal{T}(u,v,w)$, we can also map these graded properties to the xyz-space, allowing for graded heterogeneity.

The main contributions of this work and the technical advantages of the proposed method can be summarized as follows:

- We propose a natural construction scheme for microstructures that conform to the macro-shape of freeform trivariate solid models, which is based on the periodic implicit tiles.
- The concept of non-linear inversion plays an important role in the design of a highly efficient and precise slicing algorithm for freeform trivariate volumetric solids whose interiors are heterogeneous with functionally graded properties such as material, translucency, or color.
- By parametrizing the implicit tiles, tiles in the constructed microstructures can also continuously (and possibly smoothly) vary in their geometry and topology across macro-shape $\mathcal{T}$.
- Based on geometric and computational tools for solving systems of non-linear polynomial equations [2, 10, 16, 38, 39], we efficiently produce solutions which are locally precise – numerical errors are bounded within machine precision.
- Finally, the fact that the representation is precise throughout makes it suitable for analysis. Aims at directly analyzing implicit forms are already commencing [36] and this work can clearly support those efforts.

2. Previous Work

Conventional methods for volumetric modeling are capable of representing solid objects with interior properties [7, 19]. Recent development of additive manufacturing technologies introduced new ways of modeling and fabricating the volumetric interior, using microstructures [1, 23, 31, 37, 43]. The main technical challenge here is how to align the microstructures naturally with the macro-shape of a given solid object, in particular with the boundary surfaces [12, 13, 15, 34, 41, 44]. There are basically two different types of construction methods: (i) the generation of microstructures, based on certain procedural rules, directly in the interior volume [15, 26, 30, 44], and (ii) the design of microstructures in some parametrization of the Euclidean space [1, 28, 40, 47].

The methods in the first type are often based on geometric operations (such as minimum distance and surface offset computations) on the macro-shape of the object under consideration. Due to certain geometric singularities near the skeleton of the object, it is quite cumbersome to deal with the arrangement of periodic microstructures in some intricate parts. On the other hand, in the second type, the volumes are typically decomposed into relatively simple trivariates such as curved cuboids [27, 29] and curved tetrahedra [12]. The freeform deformations for these trivariate primitives are assumed to be nowhere Jacobian-vanishing. Nevertheless, in some cases, it is hard to avoid large distortions of these volumes and the microstructures embedded in them. There are many recent results [18, 28], developed for handling these cases. In this work, we focus on the simple case of objects being composed of curved cuboids.

Triply periodic minimal surfaces (TPMS) are useful implicits for populating microstructures within cuboids, in a repetitive fashion along each direction. In particular, TPMS surfaces provide an additional advantage of being locally minimal in surface area, which might explain why they are often found in biological shapes with porous scaffold structure [5, 48]. Being a special type of implicit surfaces, it is conceivable that the TPMS-based modeling techniques can easily be extended to adapt the function-based procedural representation of Pasko et al. [34] for heterogeneous volumetric microstructures in a compact, precise, and arbitrarily parametrized form based on generic geometric operations such as blending and deformations [6, 8, 9, 14, 20, 25]. Nevertheless, the majority of conventional work on TPMS-based microstructures considered TPMS surfaces which are static, often with simple Boolean operations such as intersection and trimming [13, 21, 22, 45, 48].

It is only a recent development that deformations are applied to TPMS implicit models to generate heterogeneous porous scaffolds for freeform trivariate volumetric objects [20]. The surfaces under deformation do not preserve mean curvature, and thus they may not be minimal surfaces anymore. Nevertheless, the TPMS-based trivariate mapping provides many useful properties for porous scaffold design. In particular, Hu and Lin [20] demonstrated constrained optimization techniques for the generated porous scaffold so that various non-trivial design requirements can be handled in a systematic way. Based on porous synthesis and topological optimization and using implicit B-spline functions in the parameter domain of the trivariate B-spline volumes, Gao et al. [14] approximated porous structures more general than those based on TPMS models. In the current work, we also employ TPMS and/or B-spline implicit functions in the parameter domain of the trivariate mapping.

The quality of final fabrication results is highly dependent on the efficiency and precision of slicing algorithms employed in AM technologies. Though popular in many implicit modeling techniques including those for TPMS models, the conventional marching cubes algorithm [24] has limitation in meeting the high-resolution requirement for 3D printing, which is in the level of tens of microns and even below microns. Regarding the storage efficiency, Hu and Lin [20] proposed a procedural way of representing porous scaffolds in a compact way, often reducing STL files of hundreds of megabytes to a procedural format of less than one megabyte. Feng et al. [13] tried to improve the slicing precision using a refined constrained Delaunay triangulation method and constructed multiscale pores for external freeform geometries. Moreover, Yoo [46] proposed the representation of slicing result as an image on the plane of slicing. Nevertheless, even the image resolution for typical rendering applications may not be sufficient for physical fabrication, in practice. In this work, we propose a highly efficient approach to the slicing of trivariate macro-shape $\mathcal{T}$ embedded with implicit tiles, in an arbitrary precision.

Geometric tools developed for intersection problems play an important role in the acceleration of our algorithms – for example, the slicing process is essentially a plane-volume intersection. Sederberg et al. [38, 39] introduced the use of tangent and normal cones as an important geometric concept for a single-component test, in surface intersection problem. Based on a similar concept, Elber and Kim [10] presented a simple geometric test for the uniqueness of multivariate equation solver. Moreover, Barton et al. [2] developed a topologically guaranteed algorithm for computing univariate solutions when the multivariate polynomial systems are underconstrained. All these algorithms can be accelerated by techniques, such as Hanniel and Elber [16], for bounding the tangent and normal maps in multivariate equation solvers.

3. Algorithms

We are now ready to discuss the various algorithms we employ toward implicit based tilings, in microstructures. In Section 3.1 we discuss the representation and the structures we use, while in Section 3.2, precise slicing of graded heterogeneous microstructures with implicit tiles is considered.

3.1. The Implicit Microstructure Representation

Let $\mathcal{T}(u, v, w) : D \to \mathbb{R}^3$ be a regular parametric macro-shape trivariate volumetric representation (V-rep). By regular we mean that the Jacobian of the function never vanishes. $\mathcal{T}$, the macro-shape, is also assumed to be globally self-intersection free and will capture the general shape of the microstructure. Implicit tiles will be embedded in its domain, D. Now let

$$\mathcal{I}(x, y, z) : [0, 1]^3 \to \mathbb{R}, \tag{1}$$

be a scalar regular implicit tile, in a unit cube, with $\mathcal{I} \geq 0$ to be considered inside the model. The boundary (interior) of $\mathcal{I}$ can be defined as either a zero (positive) set of a parametric trivariate V-rep spline function or a zero (positive) set of a closed form, precise (transcendental), function (e.g., a TPMS function), etc.

$\mathcal{I}(x, y, z)$ will typically be periodic. That is, its X_{min} boundary will identify with its X_{max} boundary and same for Y and Z. In many cases, we might require smoothness along these periodic boundaries or even higher continuity. By populating D, the domain of $\mathcal{T}$, with a 3D grid of size (L, M, N), of implicit tiles, $\mathcal{I}(x, y, z)$, we create the microstructure $\mathcal{M}$.

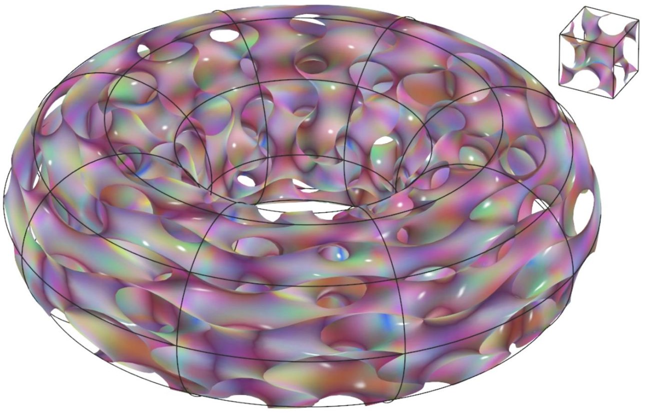

Figure 1: A marching cubes' polygonal approximation of an implicit heterogeneous Gyroid microstructure in a torus trivariate macro-shape (shown in block wireframe). A Gyroid function is defined as the zero set of $\mathcal{I}(x, y, z) = \sin(2\pi x)\cos(2\pi y) + \sin(2\pi y)\cos(2\pi z) + \sin(2\pi z)\cos(2\pi x)$, where $x, y, z \in [0, 1]^3$. One tile is shown on the top right corner.

Now, reconsider the tile from Equation (1) as a vector function:

$$\mathcal{I}(x, y, z) : [0, 1]^3 \to \mathbb{R}^k, \quad k > 1, \tag{2}$$

where the vector of $\mathcal{I}$ contains the scalar implicit function in its first coordinate and the rest are properties, like material or color in the tile. Alternatively, the macro-shape $\mathcal{T}$ can also hold additional material properties:

$$\mathcal{T}(u, v, w) : D \to \mathbb{R}^k, k > 3, \tag{3}$$

where the coordinates of $\mathcal{T}$, for $k > 3$, will hold the property values, over the macro-shape.

Back to tile $\mathcal{I}$, one can clearly construct an approximation for the boundary surface of the implicit S_0 : $\mathcal{I}(x, y, z) = 0$ in $[0, 1]^3$, using some variation of Marching Cube [24]. S_0 will now be mapped through $\mathcal{T}$, as $\mathcal{T}(S_0)$, for each position in D of the $(L \times M \times N)$ tiles in the microstructure. Then, and if $\mathcal{I}$ (or $\mathcal{T}$) is heterogeneous, the material properties of $\mathcal{I}$ (or $\mathcal{T}$) are evaluated at the vertices of the polygons from the marching cubes' approximation, only to determine the properties (colors) of those vertices. One can rely on the (color) interpolation abilities of the graphics library to interpolate the (color) properties between vertices, inside each polygon. Figure 1 shows one example of a marching cubes approximation of an implicit tile (that is computed once), conformingly embedded in a torus macro-shape, $\mathcal{T}$, of a torus. Heterogeneity is prescribed here by the tile (i.e., Equation (2)).

While we do support this approach and provide the end user with a (polygonal) approximation of microstructure $\mathcal{M}$, mostly toward visualization, we also aim at a more precise alternative. As we seek to support graded heterogeneity in the interior of the model, interior that is not represented by this polygonal approximation and because this polygonal approximation has a limited accuracy, we also support a deferred (volumetric) evaluation where we very efficiently support point by point evaluation, including at the interior of the model, as is discussed in the coming Section 3.2, toward slicing for additive manufacturing.

3.2. Precise Point Evaluation and Slicing of Implicit Microstructures

Traditionally, the 3D printing process of 3D models, is performed one planar section or slice after another. For homogeneous materials, the outline of the, typically boundary representation (B-rep), model is computed, only to fill in the interior with a homogeneous material. However, in order to support graded heterogeneity, that, for example, can be encoded in the regular V-rep macro-shape $\mathcal{T}$ and/or in the regular V-rep tiles $\mathcal{I}$, an outline is no longer sufficient.

Each planar slice must prescribe the material at any interior location and at any desired resolution. Toward this end, we follow the Voxel Printing [2] interface of Stratasys, where each slice is an image at a desired resolution, and each pixel in a slice is assigned with a color that reflects the desired material in that pixel, at that Z level.

Recall that the microstructure $\mathcal{M}$ is represented as a parametric trivariate V-rep macro-shape, $\mathcal{T}$, tiled with $(L \times M \times N)$ implicit tiles $\mathcal{I}_{lmn}$. We intersect $\mathcal{M}$ with a slicing plane P_s, and represent the result as an image at the desired resolution. W.l.o.g., assume P_s is parallel to the xy-plane. For each pixel $\mathbf{p} \in P_s$, one needs to perform the following operations:

1. Compute the (u, v, w) parameters in $\mathcal{T}$ of $\mathbf{p}$. That is, solve for the (u, v, w) parameters for which $\mathcal{T}(u, v, w) = \mathbf{p}$, if any.
2. Derive the specific implicit tile, $\mathcal{I}_{lmn}$ that these specific (u, v, w) parameters are in its domain.
3. Evaluate the sign of $\mathcal{I}_{lmn}$ at this location, to determine if we are inside this implicit tile or outside.
4. If found inside:
    - If either the tile $\mathcal{I}_{lmn}$ (with higher priority) or the macro-shape $\mathcal{T}$ have additional material properties, employ those properties to set the heterogeneous property value of $\mathbf{p}$.
    - Otherwise, set $\mathbf{p}$ to be of homogeneous build material.

   Otherwise, set $\mathbf{p}$ to be of no-material property (background).

Algorithm 1 details this process. Lines 3 and 4 in Algorithm 1 traverse pixel after pixel so we can compute the steps just described for all pixels in the image. Line 7 computes the inverse of $\mathcal{T}$, deriving the (u, v, w) parameter values so that $\mathcal{T}(u, v, w) = (x_i, y_j, z_{slice})$. As pixels in the slicing image might be outside $\mathcal{T}$, in Line 8, we identify such invalid cases and skip them. If valid, the right implicit tile, $\mathcal{I}_{lmn}$ that contains these parameters is identified in Line 10, that also maps these parameters to $\mathcal{I}_{lmn}$'s local domain. If we are inside the implicit ($\mathcal{I}_{lmn}(u_l, v_m, w_n) \geq 0$ in Line 11), we set the right color/material, in Lines 12 to 20. If tile $\mathcal{I}_{lmn}$ has material information (its dimension is greater than one), it is used to prescribe the material. Otherwise, if the macro-shape $\mathcal{T}$ has material information (its dimension is greater than three), it is employed. Then, if neither $\mathcal{I}_{lmn}$ nor $\mathcal{T}$ has material information, a homogeneous materials color is set. Finally, if we are outside the implicit ($\mathcal{I}_{lmn}(u_l, v_m, w_n) < 0$), a background color is assigned to that pixel.

Function **Dimension** in Algorithm 1 returns the dimensions of the given function - one for a scalar field. Then, function **HeterogenousMaterialColor** maps the material properties, encoded in the (higher dimensions of the) given function(s), at the given parameter location, to an RGB color.

Clearly, by far, the most challenging task in Algorithm 1 is the solution of the inverse of $\mathcal{T}$, in Line 7. Remember that $\mathcal{T}$ has at least three Euclidean coordinates (and possibly some additional heterogeneous properties' dimensions), and denote these three Euclidean coordinates by $\mathcal{T} = (\mathcal{T}_x, \mathcal{T}_y, \mathcal{T}_z, ...)$. Computing this inverse amounts to solving three nonlinear equations in three unknowns, (u, v, w):

$$\begin{aligned} \mathcal{T}_x(u, v, w) &= x_i, \\ \mathcal{T}_y(u, v, w) &= y_i, \\ \mathcal{T}_z(u, v, w) &= z_{slice}. \end{aligned} \tag{4}$$

We follow [11] that solves these nonlinear constraints using a subdivision solver [10, 16]. We will be computing this inverse function billions of times, millions of times per image. Hence, the efficient evaluation of this inverse is crucial for reasonable computation times. Yet, we have two potential advantages we can employ here:

- Macro-shape trivariate $\mathcal{T}$ and the set of implicit trivariate tiles $\{\mathcal{I}_{lmn}\}$ are all assumed locally and globally self-intersection free. Therefore, if we found a solution to the inverse problem, this is it - it is unique.

- We traverse the pixels one at a time, along the image plane. Hence, a valid (u, v, w) solution of the next pixel (x_i, y_j) in the P_s plane at Z level z_{slice} is going to be very similar to a previous nearby valid solution, for example a valid solution of pixels (x_i, y_{j-1}) or pixel (x_{i-1}, y_j).

  Following [11], we exploit this neighborhood coherence and employ numerical marching from one inverse solution to the next, refraining from the need to employ a full subdivision based solution, for each pixel. This results in one or two orders of magnitudes of improvement in computation times, compared to the application of the subdivision solver on every inverse computation, on each pixel.

Indeed, every application of the subdivision solver on a pixel, solving Equations (4), is, time-wise, several orders of magnitudes longer, compared to numerical marching. However, we are required to examine all pixels in the image, including those that are outside $\mathcal{T}$. $\mathcal{T}$ is rarely a rectangle that completely fills the image and hence many outside pixel, with no valid neighboring solutions, will force a full subdivision solution, a solution that is bound to fail.

To remedy this difficulty, we first compute the outline of the domain of $\mathcal{T}$, as is done in B-rep based 3D printing of homogeneous materials. This outline is derived by intersecting P_s with the boundary surfaces of $\mathcal{T}$. Then, we

[2] https://www.stratasys.com/en/resources/videos/voxel-the-3d-printed-pixel

Algorithm 1 SliceImplicitMicrostructure - Slicing an implicit microstructure at $Z = z_{slice}$ level.

Input:

$\mathcal{T}(u, v, w) : D \to \mathbb{R}^k$: A parametric volume trivariate of the macro object;
$\mathcal{I}(x, y, z) : [0, 1]^3 \to \mathbb{R}^k$: An implicit function defined over $[0, 1]^3$, that prescribes an implicit tile;
(L, M, N): Number of repeated $\mathcal{I}$ tiles in the u, v, w directions of $\mathcal{T}$;
z_{slice}: The Z-level of the plane to slice the implicit microstructures;
W, H: The dimensions of the sliced image;

Output:

Img: A sliced image;

Algorithm

```
$(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$ := Dimensions of $D$;
$\{\mathcal{I}_{lmn}\}$ := set of tiles populating $D$, $0 \le l, n, m \le L, M, N$;   // 3D grid of implicit tiles.
for $i := 0$ to $W - 1$ do   // Scan all pixels in the image of sliced plane, $P_s$.
    for $j := 0$ to $H - 1$ do
        $x_i := x_{min} + i\frac{x_{max}-x_{min}}{W}$;   // Derive $\mathbb{R}^3$ location of current pixel.
        $y_j := y_{min} + j\frac{y_{max}-y_{min}}{H}$;
        $(u, v, w) := \mathcal{T}^{-1}(x_i, y_j, z_{slice})$;   // Inverse of $\mathcal{T}$ - Parametric location in $\mathcal{T}$ for this $\mathbb{R}^3$ location.
        if $(u, v, w)$ invalid then
            continue;   // No valid solution found.
        $(u_l, v_m, w_n) := (u, v, w)$ in local parameters of tile $\mathcal{I}_{lmn}$;   // $(u, v, w)$ is in the domain of $\mathcal{I}_{lmn}$.
        if $\mathcal{I}_{lmn}(u_l, v_m, w_n) \ge 0$ then   // Inside the tile.
            if Dimension($\mathcal{I}_{lmn}$) > 1 then   // $\mathcal{I}_{lmn}$ is heterogeneous.
                $Img(i, j)$ := HeterogenousMaterialColor($\mathcal{I}_{lmn}$, $(u_l, v_m, w_n)$);
            else   // Map property coordinates into RGB color.
                if Dimension($\mathcal{T}$) > 3 then   // $\mathcal{T}$ is heterogeneous.
                    $Img(i, j)$ := HeterogenousMaterialColor($\mathcal{T}$, $(u, v, w)$);
                else
                    $Img(i, j)$ := Homogeneous material color;
        else
            $Img(i, j)$ := background color;
Return $Img$;
```

only consider the pixels (and evaluate their inverse) that are inside this outline. With all these improvements, we typically employ less than a handful of subdivision based solutions per slice, a slice that can hold millions of pixels.

Graded heterogeneity can be introduced by different material values at different tiles or macro-shape locations but can also be reflected in different tile geometries, in different tiles in the microstructure. In [32], it was suggested to adjust the threshold value of the implicit function as a mechanism to locally adjust the tile's geometry:

$$\mathcal{I}_{lmn}(u_l, v_m, w_n) = C(u_l, v_m, w_n). \tag{5}$$

$C(u_l, v_m, w_n)$ is a function of the local tile coordinates (u_l, v_m, w_n) but it can also be a function of the global macro-shape coordinates (u, v, w) or a function that depends on stress or heat transfer analysis over the structure, etc. In some previous work, the value of C is fixed per implicit tile, which results in minute C^0 discontinuities between adjacent tile. However, one can control C in Equation (5) so that adjacent tile will present the same C values over their shared face, preserving the continuity.

An implicit function can clearly present different boundaries in its six faces. Assume $\mathcal{I}_{lmn}(u_l, v_m, w_n)$ is made into a function that controls the shape of the six boundaries:

$$\mathcal{I}_{lmn}(u_l, v_m, w_n) \\ (X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}). \tag{6}$$

The six faces of each individual implicit tile in $\mathcal{M}$ can be all different while continuity could still be fully preserved. Recall Equation (3). By having an additional scalar continuous field $\mathcal{T}_s$ in $\mathcal{T}$, as

$$\mathcal{T} = (\mathcal{T}_x, \mathcal{T}_y, \mathcal{T}_z, \mathcal{T}_s),$$

$\mathcal{T}_s$ can be evaluated at the center of each face of each implicit tile to prescribe a parameter that controls the geometry of that face. Common faces of adjacent tiles will share the same $\mathcal{T}_s$ value, being at the same location in the domain of $\mathcal{T}$. For example, the X_{max} face of $\mathcal{I}_{lmn}$ identifies with the X_{min} face of $\mathcal{I}_{l+1,m,n}$, and hence will share a $\mathcal{T}_s$ value. Hence, in all, as long as $\mathcal{T}_s$ is continuous, the entire created model will be continuous. In the next section, we demonstrate all the options and abilities presented in this section, on some examples.

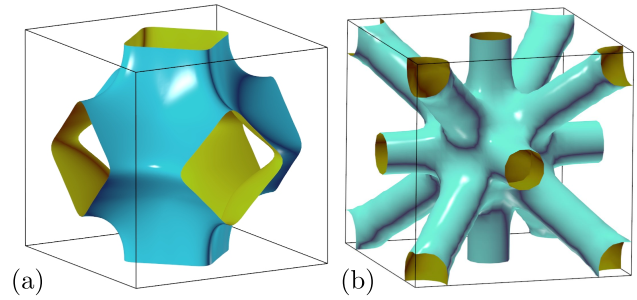

Figure 2: Two examples of implicit trivariate tiles. The left shows an implicit 3D cross trivariate tile, with arms with rounded square cross sections. The right presents an implicit trivariate tile and this tile has both diagonal and axis-parallel arms. Shown are marching cubes approximation of the geometry.

4. Experimental Results

We now present several examples of microstructures using 3D V-rep macro-shapes and implicit tiles. All presented examples were 3D printed on a J55 printer of Stratasys [42]. In the forthcoming examples, we present polygonal approximations that resulted from a marching cubes algorithm applied to the models as well as slices of the models, using precise slicing, as discussed in the previous section, and the final 3D printed results.

Figure 2 shows two implicit tiles. To define these tiles, we employed a set of user-prescribed curves in the unit cube, and constructed an implicit trivariate spline that approximates the distance field to the given set of curves. Then, a constant distance/level set of the field defines the implicit. The tile in Figure 2 (a) is a 3D cross tile with arms with a rounded square cross, whereas the tile in (b) has both diagonal arms as well as axis-parallel arms. The tile in (a) is a tri-quadratic B-spline trivariate with a mesh size of $(5 \times 5 \times 5)$ whereas the tile in (b) is tri-quadratic B-spline trivariate with a mesh size of $(12 \times 12 \times 12)$. Modeling the topology of such a tile as in (b), using parametric forms, can be quite challenging.

The implicit tile from Figure 2 (a) is embedded (many times) in the domain of a macro-shape wing trivariate V-rep model, in Figure 3. This wing resides in $\mathbb{R}^6$, for XYZ but also RGB colors. The wing is a Bézier trivariate of original orders $(2 \times 2 \times 4)$ but it was refined to allow for a better capture of the gradation of colors along it.

Figure 4 employs the implicit tile shown in Figure 2 (b). The tile is embedded in a macro-shape twisted trivariate V-rep torus of a square cross section that also resides in $\mathbb{R}^6$, for XYZRGB. This torus is of orders $(2 \times 2 \times 3)$ and a mesh size of $(2 \times 2 \times 42)$. Figure 5 shows a pair of slices out of the around 1600 slices computed to 3D print this model. Each slice is about (1600×1600) pixels[3]. It took about an hour to compute all those slices on a modern PC workstation, using eight threads (in parallel computation), and over ten times that much time to 3D print it. The final model is about 140mm in diameter.

Figure 6 shows a duck macro-shape model with a Gyroid implicit tile. The duck is a tricubic trivariate V-rep with a mesh size of $(13 \times 8 \times 17)$. Close to 4000 slices were computed for this duck model, in several hours, each slice of a little over (2000×1000) pixels. The model itself is about $(150 \times 75 \times 70)$ mm in size. Heterogeneity was again prescribed in this model via the macro-shape trivariate that resided in $\mathbb{R}^6$, for XYZ and RGB, and it can be seen that colors are varying across the entire duck model. Figure 7 portrays a few snapshots out of these almost 4000 slices.

The most time-consuming task of the slicing algorithm is in the computation of (u, v, w) parameters in $\mathcal{T}(u, v, w)$ for each pixel coordinate (x, y, z_{slice}) in the image slice. In our computations, almost 90% of the execution time was spent on the task of solving these nonlinear equations identifying (u, v, w) parameters for the given (x, y, z_{slice}) coordinates, even with outline tests and considering local coherence in the marching process. The complexity of the slicing algorithm depends on the number of evaluated pixels, and the computation time of the slicing algorithm decreased by about an order of magnitude by using outline tests and locality, to vastly reduce the number of subdivisions' based solutions. Further, the proposed slicing algorithm is also simple to parallelize, as the algorithm can be executed simultaneously for each slice, and even can be parallelized within a slice.

Figure 8 presents another trivariate wing V-rep macro-shape model with an implicit trivariate tile that also holds color material. The wing is the same Bézier trivariate from Figure 3. The tile here is a tri-quadratic trivariate of mesh size $(5 \times 5 \times 5)$ in $\mathbb{R}^4$, with coordinates representing the implicit function and RGB. One tile is shown in Figure 9.

Our last example employs tiles that are size-parametrized (Recall Equation (6)). Two examples of size-parametrized 3D cross tiles are shown in Figure 10, where the six arms of the tiles are all of different diameters. These tiles are exploited in the wing that is shown in Figure 11. Here, a V-rep trivariate model of a wing is shown, that resides in $\mathbb{R}^4$, prescribing the 3D geometry but also a scalar size field that controls the diameters of the six arms of individual tiles in the wing. Each tile in this V-rep model receives six parameters prescribing the six diameters of its arms, from the size scalar field of the wing. Color property is also added locally here for the tiles, via a second RGB field that is provided alongside the parametrized tile. Two slices out of the around 1600 slices to 3D print this wing microstructure, each of size (2500×1800), are shown in Figure 12.

5. Conclusions and Future Work

In this work, we have presented a geometric modeling framework for constructing conforming microstructures that consist of implicit tiles. Graded heterogeneity can be encoded via the V-rep macro-shape or augment the implicit tile, tiles that can be of different types, closed form or trivariate-based, and arbitrary shapes and/or topologies.

In the current work, we assumed no local and global self-intersection of the trivariate volumes. No local self-intersection can be guaranteed by verifying that the Jacobian of trivariate mapping never vanishes, which can be done quite efficiently by conventional techniques for multivariate equation solver [2, 10, 16]. On the other hand,

[3] note the XY resolution of the printer we employed, the J55, is 300 DPI but the Z resolution is 0.01875mm [42]

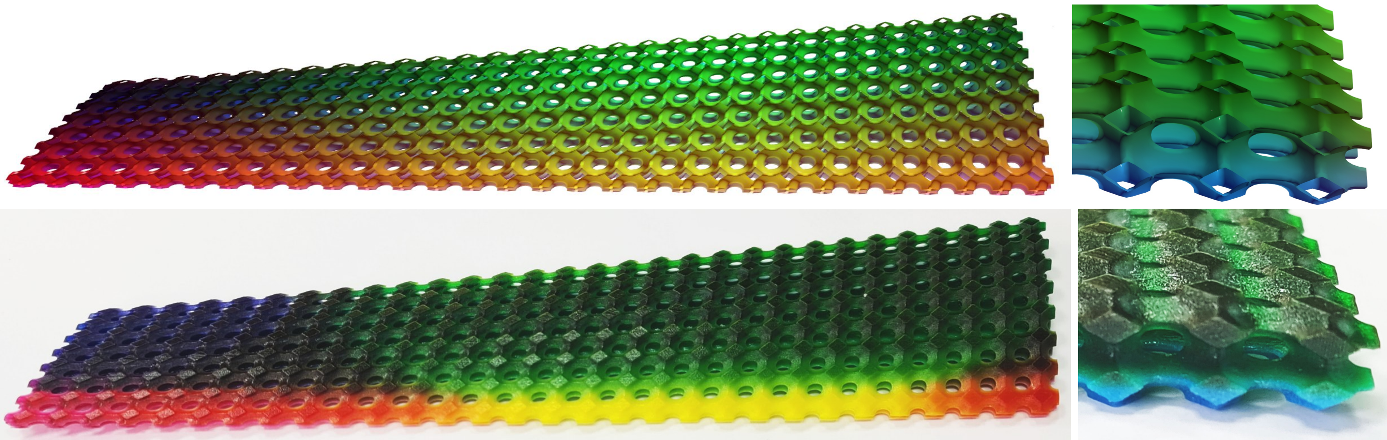

Figure 3: A 3D printed model of an heterogeneous wing tiled with the trivariate implicit cross tile from Figure 2 (a). Top row shows the computer 3D model (marching cubes) whereas the bottom row shows the 3D printed model.

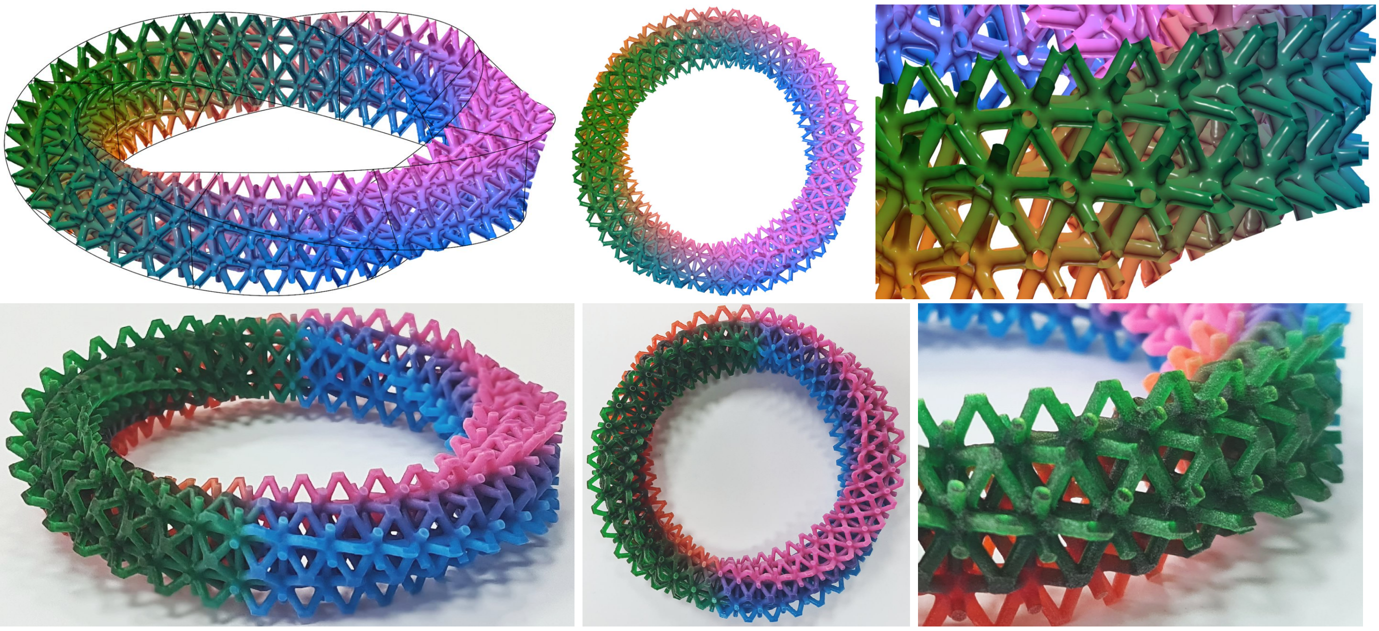

Figure 4: A twisted torus heterogeneous V-rep trivariate model in $\mathbb{R}^6$ using the implicit trivariate tile in Figure 2. Top row shows the computer model whereas the bottom row shows the 3D printed result. The twisted torus macro-shape is also shown on the top left in black wireframe.

the condition of no global self-intersection is considerably more difficult to deal with in an efficient manner. In a recent work, Park et al. [33] introduced a new approach to the surface self-intersection computation that is based on a ternary tree structure for the surface bounding volume hierarchy. We may extend this basic approach to the global self-intersection test for trivariate volumes, the technical details of which are beyond the technical scope of the current work and thus will be explored in future work.

Throughout this work, we assumed that the domains of implicits are cubes. However, one can restrict the implicit to any desired domain, much like trimming. Then, any regular or semi-regular or general tiling of a region of 3-space (the domain D of $\mathcal{T}$) can be employed, populating each element in this generalized tiling with an implicit form, while making sure the tiling is continuous.

Direct analysis of implicits is an emerging topic [36] and the representation here can directly interface with such analysis tools, including the support of heterogeneity. In fact, with the aid of analysis and optimization tools, one can (automatically) adjust the geometry and topology as well as material context of individual tiles in the microstructures, especially using implicit forms that can be parametrized to adjust their geometry and material content.

While TPMS implicits are quite common in the state-of-the-art, mostly due to their periodicity and simplicity of use, the physical advantages of TPMS, in a variety of applications, are unclear. The presented framework offers an alternative that can employ functionally graded heterogeneous implicit tiles of arbitrary shape and topolo-

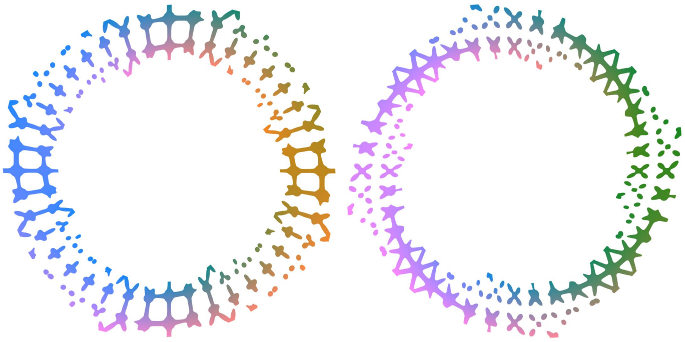

Figure 5: A few slices of the model shown in Figure 4.

gies. We conclude with the hope that with proper analysis and (topological) optimization, the solutions that this framework can offer will be of some superior value. Finally, it is plausible that with the aid of deep learning, efficient otimization of such heterogeneous geometries can be achieved.

Acknowledgments

This research was supported in part by the European Union Horizon 2020 research and innovation programme, under grant agreement No 862025, in part by the Israel Science Foundation (grant No. 597/18). and in part by the National Research Foundation of Korea (No. NRF-2019K1A3A1A78112596).

References

[1] P. Antolin, A. Buffa, E. Cohen, J. Dannenhoffer, G. Elber, S. Elgeti, R. Haimes, R. Riesenfeld: Optimizing micro-tiles in micro-structures as a design paradigm. *Computer-Aided Design*, **115**:23–33, 2019.

[2] M. Bartoň, G. Elber, I. Hanniel. Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. *Computer-Aided Design*, **43**(8):1035–1044, August 2011.

[3] D. Brackett, I. Ashcroft, R. Hague: A dithering based method to generate variable volume lattice cells for additive manufacturing. *22nd Annual International Solid Freeform Fabrication Symposium (SFF)*, Austin, TX, pp. 8—10, 2011.

[4] D. Brackett, I. Ashcroft, R. Wildman, R. Hague: An error diffusion based method to generate functionally graded cellular structures. *Comput. Struct.*, **138**:102—111, 2014.

[5] S. Callens, N. Tümer, A. Zadpoor: Hyperbolic origami-inspired folding of triply periodic minimal surface structures. *Applied Materials Today*, **15**:453—461, 2019.

[6] E. Cirillo, G. Elber: Handling heterogeneous structures and materials using blending schemes in V-reps. *Computer Aided Geometric Design*, **83**:101942, 2020.

[7] E. Cohen , R. Riesenfeld, G. Elber: *Geometric Modeling with Splines: An Introduction*, AK Peters, Wellesley, MA, 2001.

[8] S. Dahiya, A. Shein, G. Elber: Shell-lattice construction based on regular and semi-regular tiling via functional composition. *Proc. of SMI'2021 Fabrication and Sculpting Event (FASE)*, pp. 12–23, November 2021.

[9] G. Elber: Geometric texture modeling. *IEEE Computer Graphics and Applications*, **25**(4):66–76, 2005.

[10] G. Elber, M.-S. Kim, Geometric constraint solver using multivariate rational spline functions. *Proc. of ACM Symposium on Solid Modeling and Applications*, pp. 1–10, Ann Arbor, MI, June 4–8, 2001.

[11] B. Ezair, D. Dikovsky, G. Elber: Fabricating functionally graded material objects using trimmed trivariate volumetric representations. *Proc. of Shape Modeling International*, 2017.

[12] L. Feng, P. Alliez, L. Busé, H. Delingette, M. Desbrun: Curved optimal Delaunay triangulation. *ACM Trans. on Graphics*, **37**(4), Article No. 61, 2018.

[13] J. Feng, J. Fu, C. Shang, Z. Lin, X. Niu, B. Li: Efficient generation strategy for hierarchical porous scaffolds with freeform external geometries. *Addit Manuf*, **31**:100943, 2020.

[14] D. Gao, J. Chen, Z. Dong, H. Lin: Connectivity-guaranteed porous synthesis in freeform model by persistent homology. *Computers & Graphics*, **106**:33—44, 2022.

[15] A. Gupta, G. Allen, J. Rossignac: QUADOR: Quadric-of-revolution beams for lattices. *Computer-Aided Design*, **102**:160—170, 2018.

[16] I. Hanniel and G. Elber. Subdivision termination criteria in subdivision multivariate solvers. *Computer-Aided Design*, **39**:369–378, 2007.

[17] M. Helou, S. Kara: Design, analysis and manufacturing of lattice structures: an overview. *Int'l J of Computer Integrated Manufacturing*, **31**(3):243–261, 2018.

[18] Q Hong, G. Elber: Conformal microstructure synthesis in trimmed trivariate based V-reps. *Computer-Aided Design*, **140**, Article 103085, 2021.

[19] J. Hoschek, D. Lasser. *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley, MA, 1993.

[20] C. Hu, H. Lin: Heterogeneous porous scaffold generation using trivariate B-spline solids and triply periodic minimal surfaces. *Graphical Models*, **115**:101105, 2021.

[21] J. Hu, S. Wang, Y. Wang, F. Li, Z. Luo: A lightweight methodology of 3D printed objects utilizing multi-scale porous structures. *The Visual Computer*, **35**:949—959, 2019.

[22] J. Iamsamang, P. Naiyanetr: Computational method and program for generating a porous scaffold based on implicit surfaces. *Computer Methods and Programs in Biomedicine*, **205**:106088, 2021.

[23] Y. Liu, G. Zheng, N. Letov, Y.F. Zhao: A survey of modeling and optimization methods for multi-scale heterogeneous lattice structures. *Int J Adv Manuf Technol*, **143**(4):040803, 2021.

[24] W.E. Lorensen, H.E. Cline: Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics (SIGGRAPH'87 Proc.)*, **21**(4):163–169, 1987.

[25] S. Ma, K. Song, J. Lan, L. Ma: Biological and mechanical property analysis for designed heterogeneous porous scaffolds based on the refined TPMS. *J of the Mechanical Behavior of Biomedical Materials*, **107**:103727, 2020.

[26] J. Martínez, J. Dumas, S. Lefebvre: Procedural Voronoi foams for additive manufacturing. *ACM Trans. on Graphics*, **35**(4), Article No. 44, 2016.

[27] F. Massarwi, G. Elber: A B-spline based framework for volumetric object modeling. *Computer-Aided Design*, **78**:36–47, 2016.

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

[29] F. Massarwi, P. Antolin, G. Elber: Volumetric untrimming: Precise decomposition of trimmed trivariates into tensor products. *Computer Aided Geometric Design*, 2019.

[30] A. Medeiros e Sá, V.M. Mello, K. Rodriguez Echavarria, D. Covill Adaptive voids. *The Visual Computer*, **31**(6):799–808, 2015.

[31] A. Nazir, K. Abate, A. Kumar, J.-Y. Jeng: A state-of-the-art review on types, design, optimization, and additive manufacturing of cellular structures. *Int J Adv Manuf Technol*, **104**(9):3489—3510, 2019.

[32] A. Panesarm, M. Abdi, D. Hickman, and I. Ashcroft. Strategies for functionally graded lattice structures derived using topology optimisation for additive manufacturing. *Additive Manufacturing*, **19**: 81–94, 2018.

[33] Y. Park, Q Hong, M.-S. Kim, G. Elber: Self-intersection computation for freeform surfaces based on a regional representation scheme for miter points. *Computer Aided Geometric Design*,

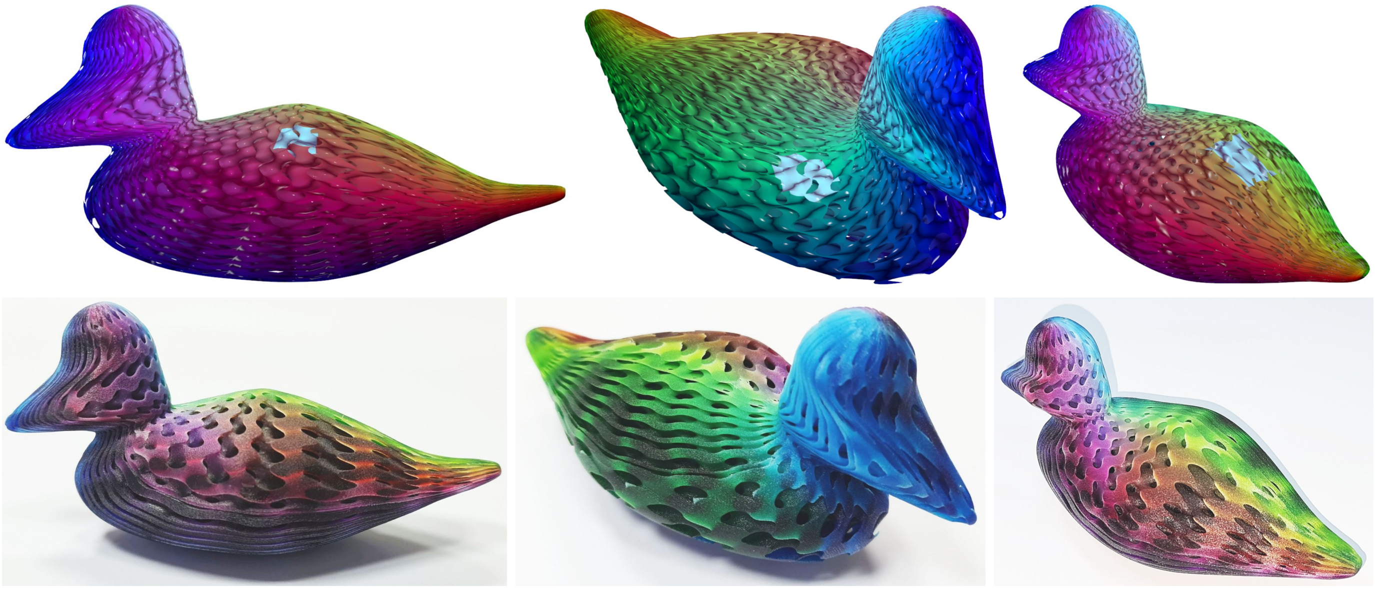

Figure 6: The top row shows a marching cubes' polygonal approximation of a heterogeneous duck model tiled with Gyroid tiles. One tile is highlighted in cyan. Bottom row shows the same model, 3D printed, from similar views.

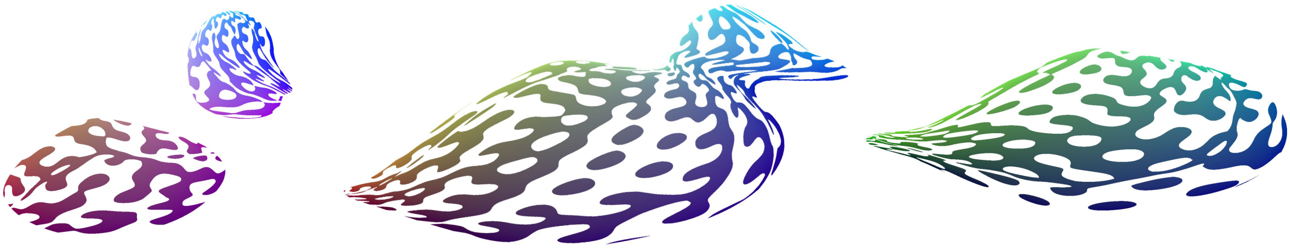

Figure 7: A few slices of the model shown in Figure 6.

86:101979, 2021.

[34] A. Pasko, O. Fryazinov, T. Vilbrandt, P.-A. Fayolle, V. Adzhiev: Procedural function-based modelling of volumetric microstructures. *Graph Models*, **73**(5):165—181, 2011.

[35] N. Patrikalakis. Surface-to-surface intersections. *IEEE Computer Graphics and Applications*, **13**(1):89–95,1993.

[36] R.I. Saye: High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles *SIAM Journal on Scientific Computing*, **37**(2):A993-A1019, 2015.

[37] T. Schaedler, W. Carter: Architected cellular materials. *Annu Rev Mater Res*, **46**(1):187—210, 2016.

[38] T. Sederberg, R. Meyers: Loop detection in surface patch intersections. *Computer Aided Geometric Design*, **5**:161–171, 1988.

[39] T. Sederberg, H. Christiansen and S. Katz. An improved test for closed loops in surface intersections. *Computer-Aided Design*, **21**(8):505–508, 1989.

[40] M. Sitharam, J. Youngquist, M. Nolan, J. Peters: Corner-sharing tetrahedra for modeling micro-structure. *Computer-Aided Design*, **114**:164–178, 2019.

[41] B. van Sosin, D. Rodin, H. Sliusarenko, M. Bartoň, G. Elber: The construction of conforming-to-shape truss lattice structures via 3D sphere packing. *Computer-Aided Design*, **132**:102962, 2021.

[42] Stratasys: *The J55 printer*.
`https://www.stratasys.com/en/3d-printers/printer-catalog/polyjet/j55-prime`

[43] F. Tamburrino, S. Graziosi, M. Bordegoni: The design process of additively manufactured mesoscale lattice structures: A review. *Trans of the ASME: J of Computing and Information Science in Eng*, **18**(4):040801, 2018.

[44] K. Vidimče, A. Kaspar, Y. Wang, W. Matusik: Foundry: Hierarchical material design for multi-material fabrication. *UIST 2016*, October 16–19, 2016, Tokyo, Japan.

[45] J. Walker, E. Bodamer, A. Kleinfehn, Y. Luo, M. Becker, D. Dean: Design and mechanical characterization of solid and highly porous 3D printed poly (propylene fumarate) scaffolds. *Prog Addit Manuf*, **2**:99—108, 2017.

[46] D. Yoo: New paradigms in hierarchical porous scaffold design for tissue engineering. *Materials Science and Engineering C*, **33**:1759—1772, 2013.

[47] J. Youngquist, M. Sitharam, J. Peters: A slice-traversal algorithm for very large mapped volumetric models. *Computer-Aided Design*, **141**, Article 103102, 2021.

[48] X. Zheng, Z. Fu, K. Du, C. Wang, Y. Yi: Minimal surface designs for porous materials: from microstructures to mechanical properties. *J Mater Sci*, **53**:10194—10208, 2018.

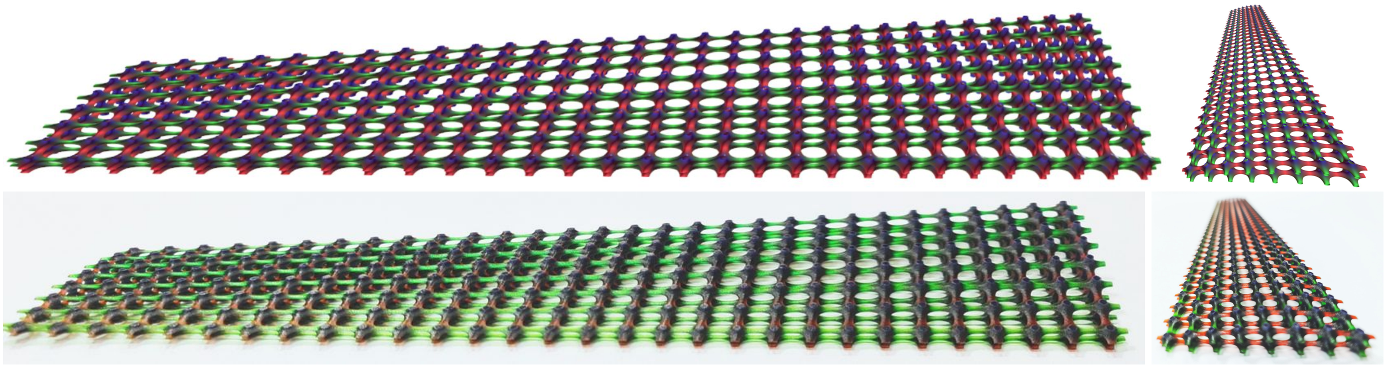

Figure 8: A 3D printed model of wing, tiled with a heterogeneous trivariate implicit cross tile that is shown in Figure 9. The geometry of this wing macro-shape model is the same model as in Figure 3.

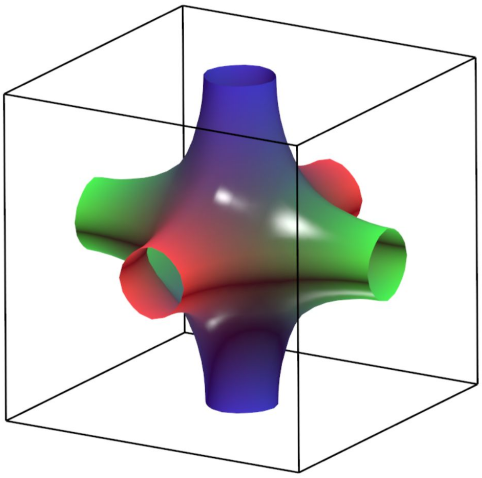

Figure 9: The implicit trivariate tile used in Figure 8.

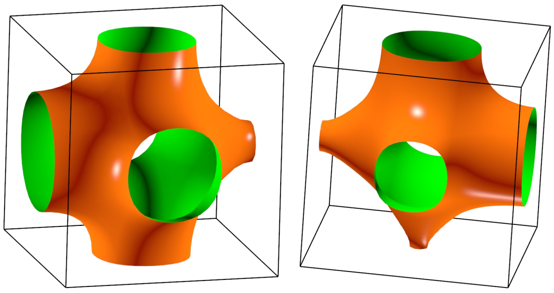

Figure 10: Parametrized implicit tiles that are used in the wing model in Figure 11. This implicit trivariate tile gets six parameters that control the diameters of the six arms of this cross shape, in U_{min}, U_{max}, V_{min}, V_{max}, W_{min}, and W_{max}. Two examples are presented.

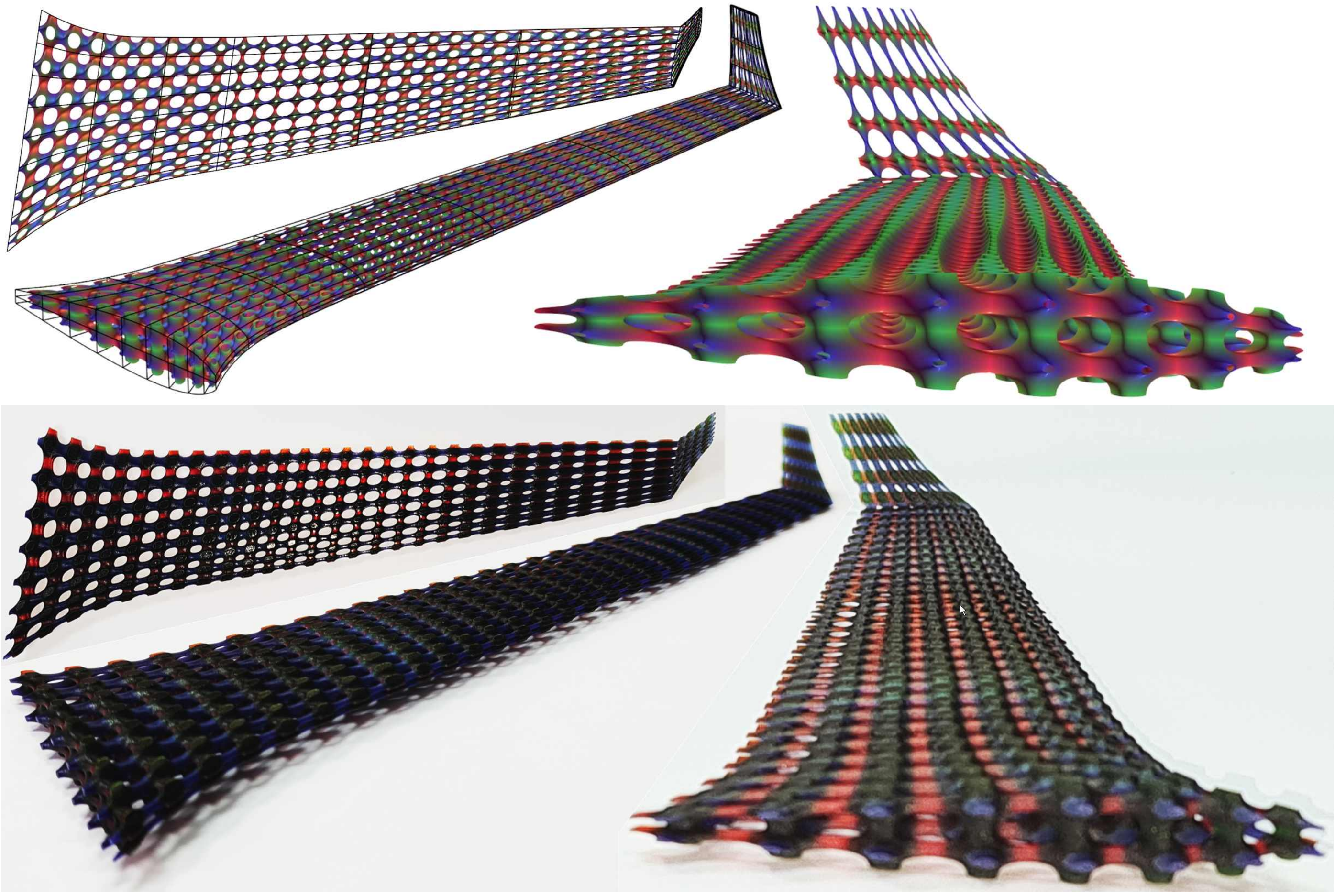

Figure 11: The top row shows a smooth heterogeneous wing model tiled with parametrized implicit trivariate tile, in three different views. Wireframe drawing of the wing macro-shape itself is also shown on the left. The bottom row shows the same wing model, 3D printed, from similar views. The arms of the 3D cross tile are of varying diameters across the wing. See also Figures 10 and 12.

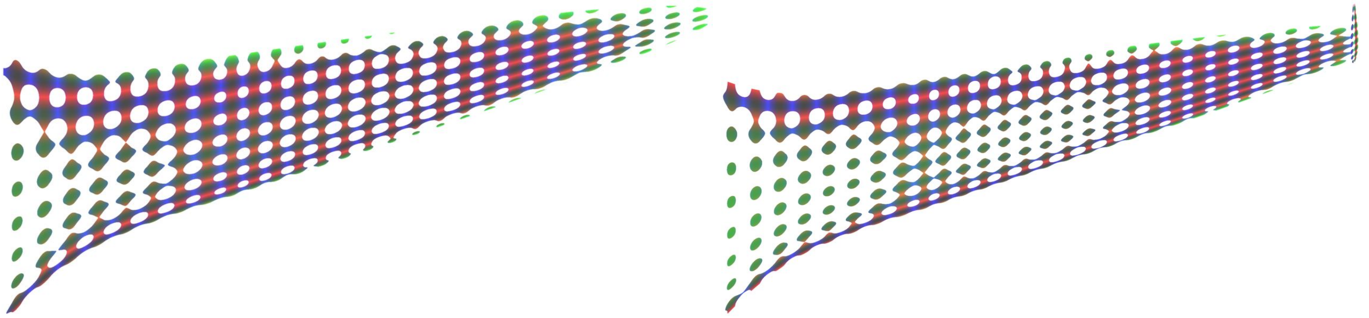

Figure 12: Two slices out of around 1600 used to 3D print the wing with parametrized implicit tiles, shown in Figure 11.