

Precise Construction of Micro-structures and Porous Geometry via Functional Composition

Gershon Elber^(✉)

Department of Computer Science, Technion – IIT, 32000 Haifa, Israel
gershon@cs.technion.ac.il
<http://www.cs.technion.ac.il/~gershon>

Abstract. We introduce a modeling constructor for micro-structures and porous geometry via curve-trivariate, surface-trivariate and trivariate-trivariate function (symbolic) compositions. By using 1-, 2- and 3-manifold based tiles and paving them multiple times inside the domain of a 3-manifold deforming trivariate function, smooth, precise and watertight, yet general, porous/micro-structure geometry might be constructed, via composition. The tiles are demonstrated to be either polygonal meshes, (a set of) Bézier or B-spline curves, (a set of) Bézier or B-spline (trimmed) surfaces, (a set of) Bézier or B-spline (trimmed) trivariates or any combination thereof, whereas the 3-manifold deforming function is either a Bézier or a B-spline trivariate.

We briefly lay down the theoretical foundations, only to demonstrate the power of this modeling constructor in practice, and also present a few 3D printed tangible examples. We then discuss these results and conclude with some future directions and limitations.

Keywords: Freeform deformation · Trivariate splines · Symbolic computation · Freeform tiling

1 Introduction and Related Work

Deformations and metamorphosis captured the attention of the computer graphics and the geometric modeling communities for several decades, while in recent years this interest has reduced a bit. The idea of freeform deformations (FFD) was introduced around thirty years ago [25] as a global deformation mapping, $\mathcal{T} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, and was originally based on trivariate tensor-product Bézier vector functions. Trivariate splines were investigated by many and herein we will only survey the use of trivariates toward deformations.

A large body of work was presented on a variety of FFD techniques, following [25], including extensions that support the B-spline representation [14] and use of FFDs in animation [5]. While, in general, FFDs map a box-shaped domain into a deformed-box in Euclidean space, other topologies were considered and, for example, [4] introduces Extended FFDs to form a deformation that

better resembles the shape of the input model. [4] suggested the use of prismatic and cylindrical FFD functions that can approximate some geometric models better than box-shaped tensor product FFDs. More general FFDs suggested the use of arbitrary topology FFDs based on subdivision volumes for free-form deformation [15].

Other, more recent, variations of FFDs considered the removal of certain topological restrictions from the deformed object. [11] considered torn surfaces that incorporated non-iso-parametric curves of C^{-1} discontinuity inside B-spline surfaces. Similarly, [23] suggested the exploitation of discontinuous FFDs to induce tears in the deformed models for animation and surgery incision simulations.

While the body of FFD work is significant, FFDs were always seen as manipulation tools of existing geometry. Almost exclusively, FFDs were applied to an existing model, resulting in a modified, deformed, model, that typically, and aside from the discussed torn surfaces abilities, preserved the topology. Also, in the last decade or so, surface detailing techniques were introduced in [8] and then in [9, 22] that are volumetric but limited to a surface layer (between the surface and its small offset, with typically a linear interpolation in between) allowing the modeling of surface details like scales or thorns. Similar surface detailing abilities can also be found in commercial packages nowadays like Rhinoceros¹.

Other efforts toward the synthesis of porous geometry and modeling with porosity are known and, for example, include stochastic methods and the use of Boolean set operations [24], including voxels' based. In [26], the 3D Voronoi diagram of a set of points serves as the basis of the pore space, thickening the Voronoi edges and/or walls, and in [19], procedural (implicit) forms are employed toward the synthesis of micro-structures while also allowing for deformations and blendings. In [1], porous modeling of scaffolds is considered toward 3D printing, where a volumetric grid-like model is synthesized to follow the basic input scaffold. These methods typically synthesize piecewise constant (i.e. voxels) and linear porous geometry and are hence of limited continuity, and are further incapable of precisely controlling the geometry that is being synthesize. This, while herein the micro-structures' results can be fully piecewise-parametric.

In this work, we fuse the general FFD's idea with a surface detailing technique into a modeling constructor of porous and/or micro-structure geometry that smoothly and precisely builds the geometry. The constructor of the porous/micro-structure receives a volumetric model, \mathcal{T} , as a trivariate and a geometric tile, and paves the tile in the domain of \mathcal{T} as desired, constructing a whole new topology of porous geometry in the general shape of \mathcal{T} . The paved tiles are then mapped to Euclidean space via a composition with the volumetric model \mathcal{T} .

The trivariate function \mathcal{T} can be of any general shape. Techniques to build trivariate functions are known for a long time (i.e. [18] but also recently [16]). The vast majority of tensor product surface constructors can be made into a trivariate constructors, including volumetric extrusions, ruled volumes or vol-

¹ <http://www.rhino3d.com>.

umes of revolution, volumetric Boolean sums, and volumetric sweeps. With the clear ability to construct primitive shaped trivariate (i.e. cones and spheres), in [16] volumetric Boolean set operations over trivariate are now also demonstrated.

This work extends our previous recent work [10] that introduced the basic micro-structure construction idea, in several ways. While [10] only considered surface-trivariate compositions and 2-manifold micro-structures, here we expand and allow tiles consisting of univariate, (trimmed) bivariate, and even (trimmed) trivariate [16], including in combination thereof in the same tile (See Fig. 1). Hence, one can manage non-manifold tiles as well as tiles consisting of multi-dimensional shapes. With the additional ability of supporting trivariate-trivariate composition, we form a closure: the resulting elements of the micro-structure are again trivariate and hence, can be recursively used in the construction of nano-structure, etc. Finally, we formalize the conditions over the mapping trivariate, \mathcal{T} , and the tile so that the constructed model will be a viable k -manifold.

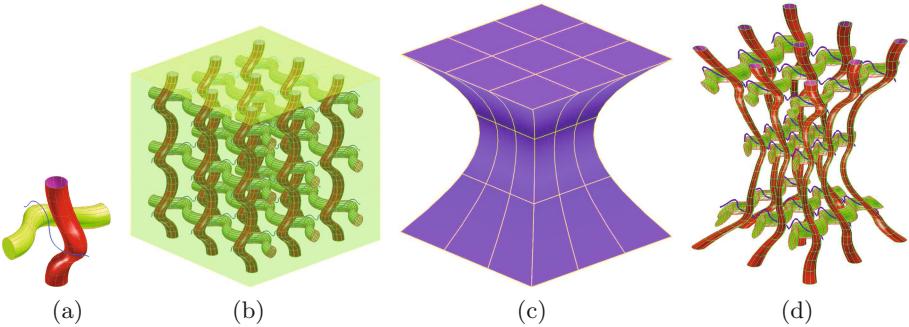


Fig. 1. A simple example of a tile, T , consisting of three different geometric types: a curve (in blue), a surface (in red), and a trivariate (in green) (a). T is paved ($3 \times 3 \times 3$) times in the domain of the mapping trivariate, \mathcal{T} , (b), that is shown in (c). (d) presents the precise smooth composition result of $\mathcal{T}(T)$. (Color figure online)

The rest of this work is organized as follows. Section 2 presents the different computational needs of this variant of a micro-structure constructor and the necessary foundations. In Sect. 3, some examples and results are presented, only to be discussed, in Sect. 4. Then, we conclude, in Sect. 5.

2 Algorithm

Let \mathcal{T} be a trivariate Bézier vector function:

$$\mathcal{T}(x, y, z) = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} P_{i_1, i_2, i_3} B_{i_1}^{n_1}(x) B_{i_2}^{n_2}(y) B_{i_3}^{n_3}(z), \quad (1)$$

where P_{i_1, i_2, i_3} are the control points of the 3D mesh of \mathcal{T} and $B_{i_1}^{n_1}(u)$ is the i_1 'th Bézier basis function, of degree n_1 .

Herein, we only discuss the necessary computation imposed by a trivariate-trivariate composition, while the cases of curve-trivariate and surface-trivariate are similar, yet obviously simpler.

Consider the trivariate-trivariate composition $\bar{\mathcal{T}} = \mathcal{T}(T)$, where \mathcal{T} is as in Eq. (1) and T is:

$$T(u, v, w) = \sum_{j_1=0}^{m_1} \sum_{j_2=0}^{m_2} \sum_{j_3=0}^{m_3} Q_{j_1, j_2, j_3} B_{j_1}^{m_1}(u) B_{j_2}^{m_2}(v) B_{j_3}^{m_3}(w). \quad (2)$$

One can map the control points, Q_{j_1, j_2, j_3} , of T through \mathcal{T} as $\mathcal{T}(Q_{j_1, j_2, j_3})$, yielding

$$\bar{\mathcal{T}} \approx \sum_{j_1=0}^{m_1} \sum_{j_2=0}^{m_2} \sum_{j_3=0}^{m_3} \mathcal{T}(Q_{j_1, j_2, j_3}) B_{j_1}^{m_1}(u) B_{j_2}^{m_2}(v) B_{j_3}^{m_3}(w), \quad (3)$$

as is typically done with the FFD of input polygonal data where only the vertices of the polygons are mapped through \mathcal{T} . However, Eq. (3) is only an approximation of $\bar{\mathcal{T}} = \mathcal{T}(T)$. Further, continuity will not be preserved and the geometry will only loosely follow the micro-shape induced by T . Alternatively, a precise mapping of T through \mathcal{T} (See Fig. 1) can be computed using function composition [6, 7]:

$$\begin{aligned} \bar{\mathcal{T}} &= \mathcal{T}(T) \\ &= \mathcal{T}(t^x(u, v, w), t^y(u, v, w), t^z(u, v, w)) \\ &= \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} P_{i_1, i_2, i_3} B_{i_1}^{n_1}(t^x(u, v, w)) B_{i_2}^{n_2}(t^y(u, v, w)) B_{i_3}^{n_3}(t^z(u, v, w)), \end{aligned} \quad (4)$$

where (t^x, t^y, t^z) are the coefficients of T . Equation (4) amounts to the computation of products of terms in the form of $B_{i_1}^{n_1}(t^x(u, v, w))$. If $B_{i_1}^{n_1}$ is a polynomial (Bézier) function and $Q_{j_1, j_2, j_3} = (q_{j_1, j_2, j_3}^x, q_{j_1, j_2, j_3}^y, q_{j_1, j_2, j_3}^z)$, then:

$$\begin{aligned} B_{i_1}^{n_1}(t^x(u, v, w)) &= \binom{n_1}{i_1} t^x(u, v, w)^{i_1} (1 - t^x(u, v, w))^{n_1 - i_1} \\ &= \binom{n_1}{i_1} \left(\sum_{j_1=0}^{m_1} \sum_{j_2=0}^{m_2} \sum_{j_3=0}^{m_3} q_{j_1, j_2, j_3}^x B_{j_1}^M(u, v, w) \right)^{i_1} \\ &\quad \left(1 - \sum_{j_1=0}^{m_1} \sum_{j_2=0}^{m_2} \sum_{j_3=0}^{m_3} q_{j_1, j_2, j_3}^x B_{j_1}^M(u, v, w) \right)^{n_1 - i_1}, \end{aligned} \quad (5)$$

where $B_J^M(u, v, w) = B_{j_1}^{m_1}(u) B_{j_2}^{m_2}(v) B_{j_3}^{m_3}(w)$.

Algorithms to directly evaluate the product (and summation) of splines, in both Bézier and B-spline forms, are known [7, 12, 13, 17], and all the above formulation can be applied to either the Bézier or the B-spline representation, with

one caution. If \mathcal{T} is a B-spline trivariate, the tile T cannot, in general, cross knot lines in \mathcal{T} . Because tensor product splines can represent finite (dis)continuities only along knots, a general crossing of a knot line of \mathcal{T} by T is likely to introduce an irrepresentable by tensor product splines diagonal (dis)continuity into $T(\mathcal{T})$. Hence, T must be divided along the knot lines of \mathcal{T} . If T consists solely of univariates, these univariates could be divided at the knot lines of \mathcal{T} , only to be re-merged into $T(\mathcal{T})$. However, if T is a surface or a trivariate it must be divided along the knot lines into smaller, not necessarily rectangular surface/trivariate patches. Further, those new patches must be again divided into rectangular/cuboid patches. While a feasible process, it is unfortunately a far more difficult process that also affects the regularity of the tiles' representation, in the micro-structure as a whole. Hence, herein we limit ourselves to bivariate and trivariate tiles that cross no knot lines in \mathcal{T} .

The continuity of $\bar{T} = \mathcal{T}(T)$ is directly governed by the lowest continuity between the continuities of \mathcal{T} and T , a result that stems directly from the chain rule of differentiation of composition functions. A function is considered regular if its Jacobian vanishes in no place. Again, and using the chain rule of differentiation \bar{T} can be shown to be regular if both \mathcal{T} and T are regular. This means that a mapping of a regular k -manifold, $k = 1, 2, 3$ through a regular trivariate \mathcal{T} will yield back a regular k -manifold, $k = 1, 2, 3$, albeit of a higher degree.

Having the ability to compute $\mathcal{T}(T)$ as well as the simpler cases of surface-trivariate (and polygon-trivariate) and curve-trivariate compositions, we consider periodic tiles in 3D that pave the domain of \mathcal{T} ($c_x \times c_y \times c_z$) times (See also Fig. 1). A tile is considered C^n periodic if the boundaries of the tile for $u_{min}, v_{min}, w_{min}$ match the boundaries of the same tile for $u_{max}, v_{max}, w_{max}$, with C^n continuity, respectively. That is,

$$\left. \frac{\partial^m t_a}{\partial p^m} \right|_{u_{min}} = \left. \frac{\partial^m t_a}{\partial p^m} \right|_{u_{max}}, \quad m = 0, \dots, n, \quad (6)$$

in all axes $a = x, y, z$ and in all permutations of parameters $p \in \{u, v, w\}$.

We also need to consider \mathcal{T} 's boundary end conditions. Consider the pavement of the domain \mathcal{T} by tile T ($c_x \times c_y \times c_z$) times. A tile T can be C^n periodic but tiles placed on the boundary of \mathcal{T} must be closed along their boundary. That is, all tiles $(i, j, 1)$ and (i, j, c_z) , $i \in (1, \dots, c_x)$, $j \in (1, \dots, c_y)$ must all be closed in the w_{min} and w_{max} directions, respectively, and the same holds for the u and v min/max boundaries. While one can consider handling these boundary openings after the mapping through \mathcal{T} , we propose a simpler remedy. Given a C^n periodic tile T , process it by computing its Boolean set operations with the six planes $u = u_{min}, u = u_{max}, v = v_{min}, v = v_{max}, w = w_{min}, w = w_{max}$, and their combination thereof, considering face, edge and vertex neighborhoods, 26 neighbors in all. As an example, tile $(1, 1, 1)$, that should be closed in u_{min}, v_{min} and w_{min} , will be applied with Boolean set operations and sealed against planes $u = u_{min}, v = v_{min}$ and $w = w_{min}$. Then, and based on the tile's indices in \mathcal{T} , the proper boundary or interior tile will be employed out of the 27 tiles we will have, in whole. While the output can include hundred of thousands if

not millions of mapped tiles, by computing the boundary tiles a-priori, only 26 different, local to the tile, Boolean set operations are required.

Then, if \mathcal{T} is a regular C^n (or better) trivariate and T is a regular C^n k -manifold periodic tile, a watertight C^n k -manifold model can be formed, except possibly at the boundaries, as the intersection curves along the Boolean set operations are typically only C^0 . In the next section, the power of this modeling constructor is fully revealed and demonstrated.

3 Results and Examples

A modeling constructor based on trivariate functions can be quite powerful. It enables the fabrication of delicate geometry that is very difficult to construct in alternative ways. In Fig. 2, the domain of a trivariate duck is paved with piecewise linear B-spline surfaces' tiles, only to be using precise surface-trivariate composition computation. Six bilinear B-spline surfaces define this hollowed tile, shown in Fig. 2(a). The result of the composition is shown in Fig. 2(b) whereas Fig. 2(c) presents a similar result when the surfaces of the tile are first converted to polygons while only the vertices of the polygons are mapped through \mathcal{T} . Note the silhouettes near the belly area, in Fig. 2(c), that are clearly C^1 discontinuous where they should have been smooth, at common boundaries between two different tiles.

We seek viable models which means they should be watertight. If each tile is watertight and closed, the result will be watertight but consisting of numerous disjoint parts. If the tiles are periodic (and possibly smoothly periodic) between $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$, the interior will be connected and hence sealed and watertight. However, we still need to close boundary openings along the boundary of \mathcal{T} . This closure is simple to achieve as explained in Sect. 2 - every tile that is a boundary tile in some direction, in the domain of the trivariate, will be sealed with the plane of that boundary, possibly using a Boolean set operation. Figure 2(d) shows a watertight porous model that resulted from applying this boundary sealing operation to the model in Fig. 2(c). As stated in the previous section, 26 such Boolean operations will be needed.

Alternatively, one can provide a-priori sealed tiles for the proper boundary. 26 such sealed tiles could be provided for all possible neighboring boundaries. Figure 3 shows a tile consisting of three Borromean rings², tailored so they can also be linked to their neighbors. In Fig. 3(a), the interior as well as the primary $u_{min}, u_{max}, v_{min}, v_{max}, w_{min}, w_{max}$ boundary tiles are shown, left to right. Figure 3(b) and (c) shows two views of a $(3 \times 3 \times 3)$ tiling using these tiles, as a full watertight and smooth model, using the interior and six boundary tiles shown in Fig. 3(a).

In Fig. 4, we pave 3D twisted tubes in a domain of a trivariate in the shape of a knot. The knot surface was created as a regular sweep of a circular cross section along a 3D knot curve. Then, volumetric Boolean sum was used to convert the

² https://en.wikipedia.org/wiki/Borromean_rings.

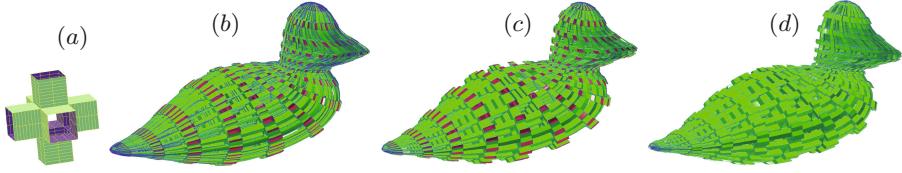


Fig. 2. Six bilinear B-spline surfaces form the tile in (a) that paves the domain of a trivariate in the shape of a duck. (b) presents the smooth and precise surface-trivar composition. In (c), the tile is converted to polygons and vertices are mapped through the trivariate, resulting in C^1 discontinuities (note the belly area). The interiors of the resulting surfaces are exposed, in magenta, in (b) and (c). (d) shows the result of the boundary sealing via Boolean set operations, creating a watertight model. See also Fig. 9(a).

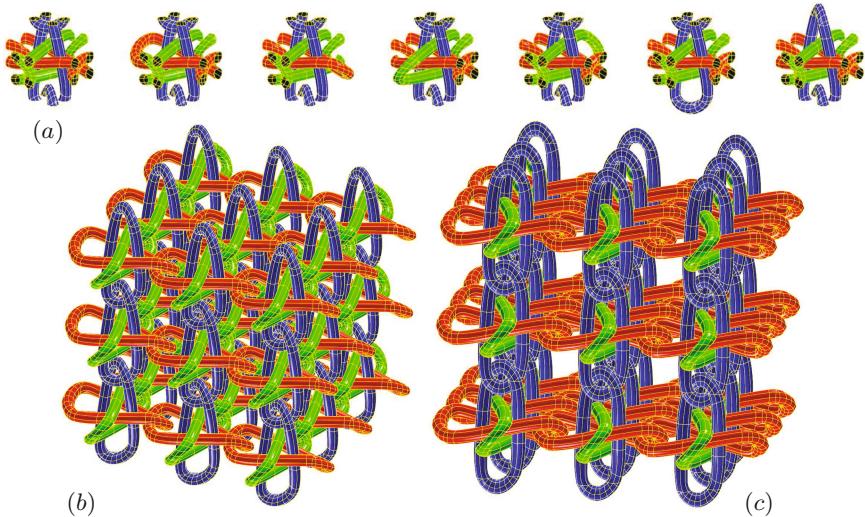


Fig. 3. One can create a sealed, watertight, model by providing sealed boundary tiles for the proper boundaries of \mathcal{T} . In (a), the interior tile and the $u_{min}, u_{max}, v_{min}, v_{max}, w_{min}, w_{max}$ tiles are provided (left to right). (b) and (c) shows two different views of a $(3 \times 3 \times 3)$ tiling, using these linked Borromean rings' tiles.

sweep surface to the trivariate that is shown in Fig. 4(b). The tile in Fig. 4(a) consists of four bicubic helical B-spline surfaces, constructed using algebraic sum [21] between a quarter of a helical curve and a circle. In this example, we pave the tiles mostly in one direction - along the axis of the knot trivariate. The tile (smoothly) shifts between the four boundary openings, bottom to top, creating the twisting effect, in the C^1 continuous final result shown in Figs. 4(c) and (d). Figures 4(e) to (h) shows the same trivariate paved using increasingly higher resolutions, in all three axes.

The complexity of the final model depends on the resolution of the pavements but also on the complexity of the individual tile. Figure 5 shows an example where

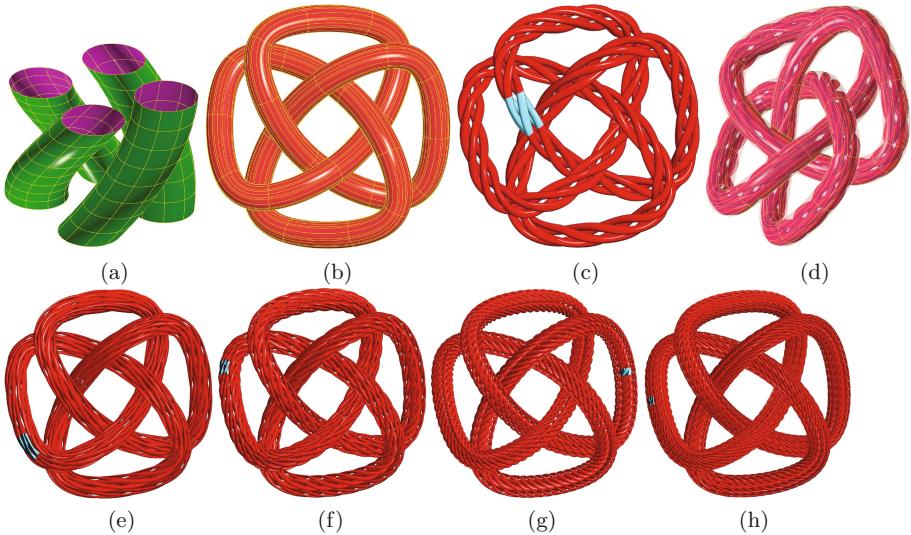


Fig. 4. A tile consisting of four B-spline helical-looking surfaces (a) is paved ($1 \times 1 \times 47$) times in the domain of a B-spline trivariate of degrees ($3 \times 3 \times 3$) and lengths ($4 \times 4 \times 50$) in the shape of a knot (b), resulting in (c) (note one tile is highlighted in cyan). (d) shows a different view of the same final result, embedded in the transparent knot trivariate. (e) to (h) present different results using increasingly higher resolutions of pavements, with one tile highlighted in cyan. (Color figure online)

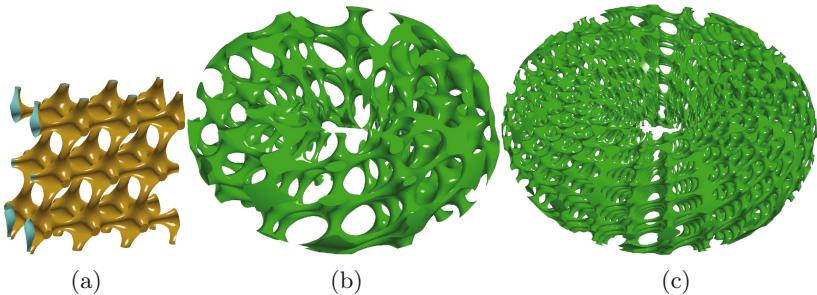


Fig. 5. A fairly complex polygonal periodic tile consisting of around 20k polygons in (a) is paving the domain of a torus trivariate using two different resolutions in (b) and (c). See also Fig. 9(b).

a fairly complex tile is exploited. The polygonal tile, in Fig. 5(a), is paving the domain of a torus trivariate in Fig. 5(b) and (c), using two different pavement's resolutions.

Because of the capability to conduct trivariate-trivariate composition, a closure is formed. A recursive application of the composition operator may be performed, and Fig. 6 demonstrates this ability. The wing model, in Fig. 6(a), is

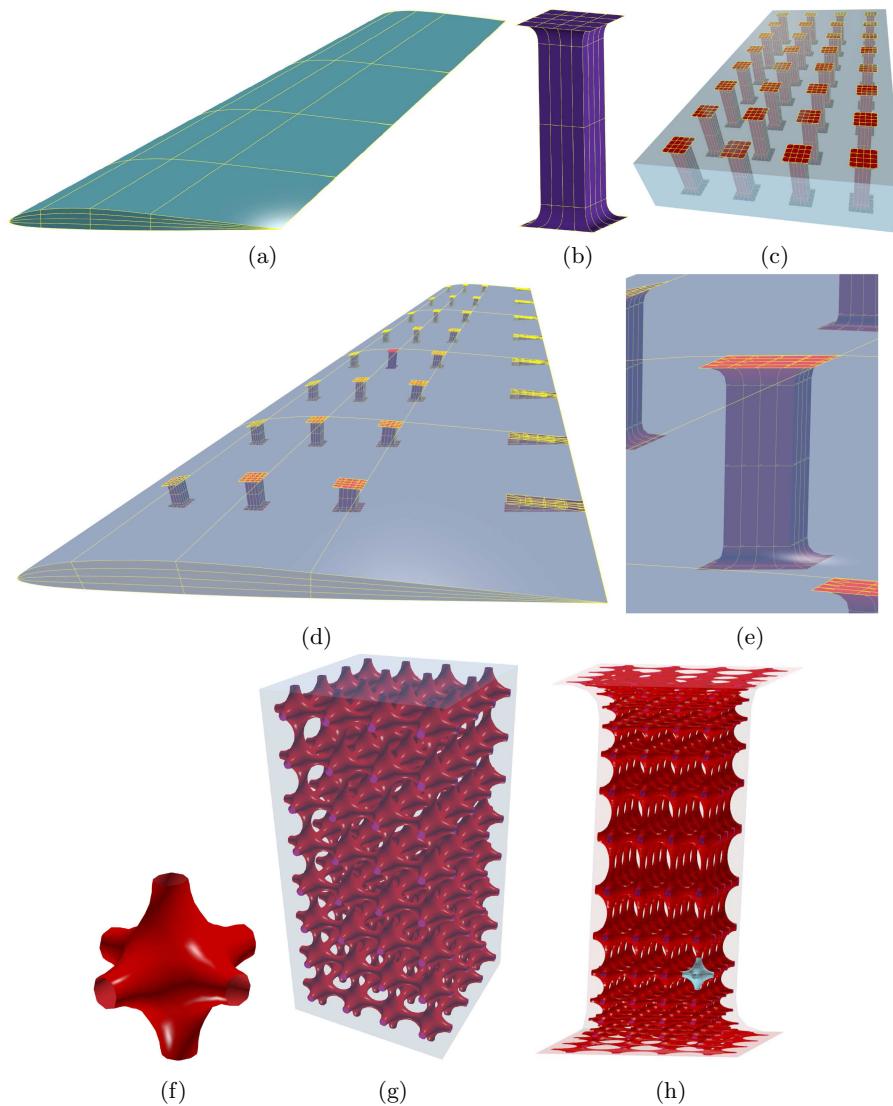


Fig. 6. A recursive application of the composition operator. The trivariate model of the wing in (a) is trivariate-trivariate composed ($8 \times 4 \times 1$) times with the vertical support trivariate pillar shown in (b). The tiling in the domain of the wing is shown in (c) and the result of the composition is shown in (d). Now each of these pillars (e), which is a zoom-in on (d), is, in turn, surface-trivariate composed with a polygonal surface tile (f). The tiling in the domain of the pillar is shown in (g) and the result of the composition is shown in (h). Note one tile is highlighted in cyan, in (h). (Color figure online)

constructed as a trivariate ruled volume. The supporting pillar structure, in Fig. 6(b), is another trivariate volume that is constructed by lofting along a set of square surfaces of different sizes. By composing the support pillar trivariate tile into the wing domain ($8 \times 4 \times 1$) times (wing's domain is shown in Fig. 6(c)), 32 deformed pillars, shown in Fig. 6(d), result. Note the 32 composed pillars are all differently shaped trivariates. In Fig. 6(e), a zoom-in on one of these pillars is shown, only to recursively compose the polygonal surface tile, in Fig. 6(f), in the pillar trivariate ($8 \times 4 \times 4$) times (pillar's domain is shown in Fig. 6(g)). The final, two-levels composition, result is shown in Fig. 6(h), with one tile highlighted in cyan.

This surface-trivariate (and trivariate-trivariate) approach can also serve to handle trimmed geometry. Herein, the tensor product geometry undergoes composition whereas the trimming information is simply propagated along, as the domain(s) of the surface(s) (or trivariate(s)) in the tile is (are) not affected. Figure 7 shows one example where three concentric trimmed through-cylinder B-spline surfaces are serving together as a tile that is composed ($4 \times 4 \times 2$) times in the mapping trivariate.

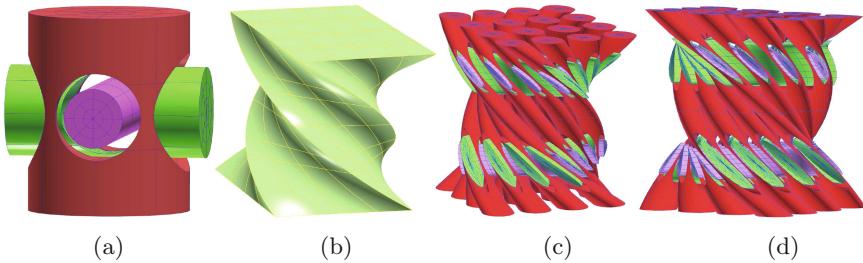


Fig. 7. Three concentric trimmed through-cylinder surfaces are serving as a tile (a) that is composed ($4 \times 4 \times 2$) times in the trivariate shown in (b). The result is shown from two somewhat different views, in (c) and (d).

We now present another example that exploits surface-surface and surface-trivariate compositions, in two composition levels, toward the precise modeling of composite materials. The domain of the B-spline duck surface shown in Fig. 8(a) is tiled in Fig. 8(b) with parallel strip surfaces that are composed with the duck surface to yield Fig. 8(c). The strip surface in Fig. 8(c) is then offset a bit to yield some real thickness and a trivariate strip is formed as a ruled volume between the original strip and its offset, and is shown in Fig. 8(d). Finally, the trivariate strip, from Fig. 8(d), is populated with tiles in the shape of stitches to model the internal stitched fibers, as is shown in Fig. 8(e) and (f) (a zoom-in).

Finally, and as a testimony for the viability of the constructed models and their watertightness, Fig. 9 presents two of the presented examples 3D-printed using additive manufacturing.

While in this work we focused on surface-trivariate and trivariate-trivariate composition, we like to add that curve-trivariate composition can also play a

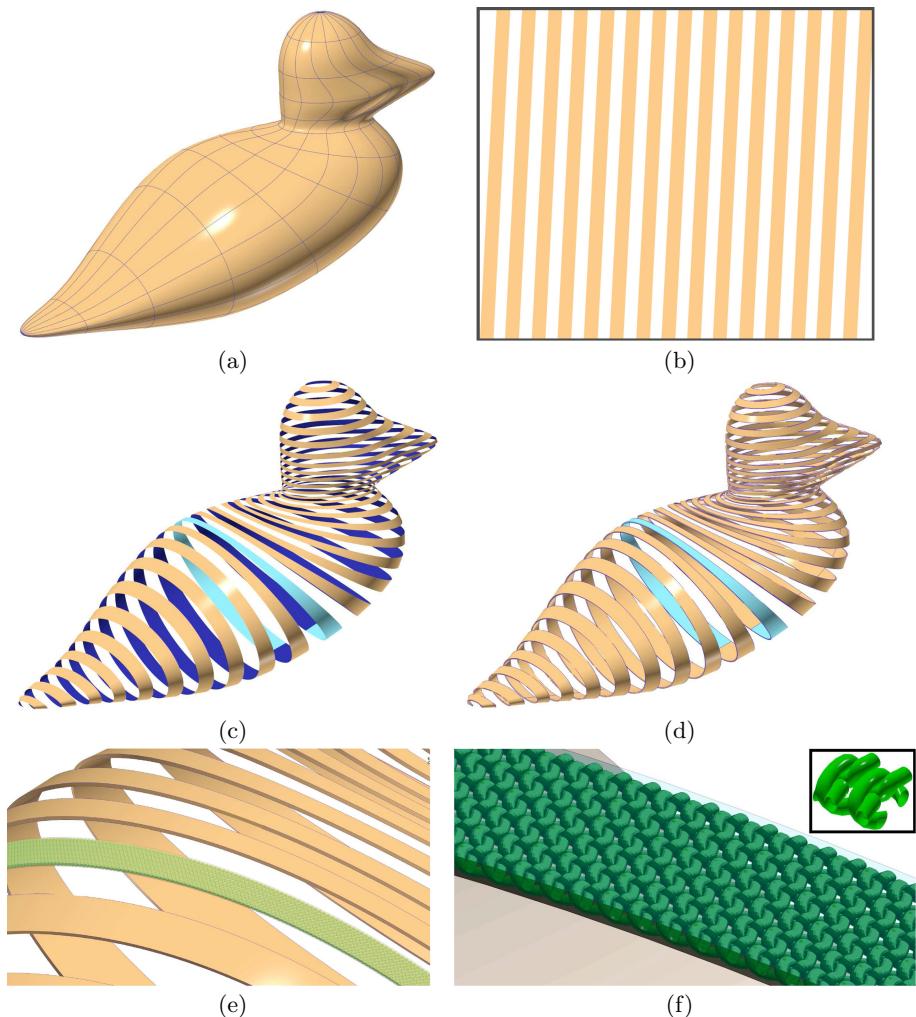


Fig. 8. A two level composition. The periodic strip in (b) is composed into the duck surface in (a) to yield (c). By offsetting the strip surface in (c) and ruling a volume between the strip and its offset, (d) is formed. The trivariate strip in (d) is made transparent (to enable the visibility of its interior) and then composed with a stitches like tile as a second composition, yielding (e) (and a zoom-in, in (f), with one tile enlarged), modeling a composite strip with detailed stitched fibers.

role in the placement of fibers, in the modeling of composites. Curves can be embedded in the domain of the mapping trivariate, \mathcal{T} , only to be mapped to Euclidean space via curve-trivariate composition. Because \mathcal{T} is unlikely to be an isometric mapping, one might be required to compensate for the distances between adjacent curves in the domain of \mathcal{T} so the mapped curves are more

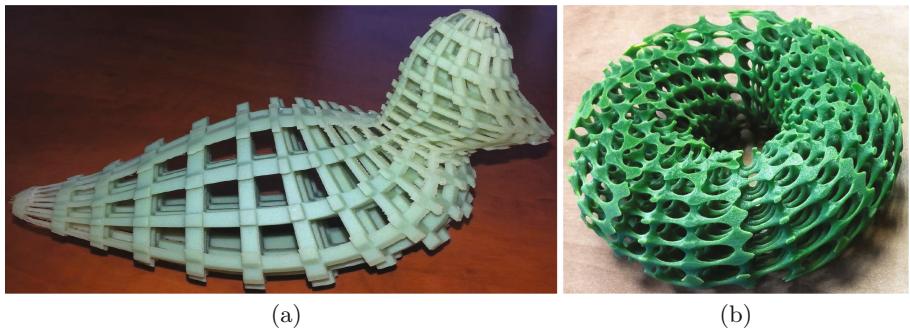


Fig. 9. A 3D printed porous duck (a) from Fig. 2 and a porous torus (b) from Fig. 5. Printing courtesy of Stratasys Israel.

equally spaced in Euclidean space. Figure 10 shows one such example, where helical curves are mapped through T to yield the Euclidean space curves. The result is a set of univariates, through which, one can, for example, sweep any cross section to precisely yield fibers of that cross section.

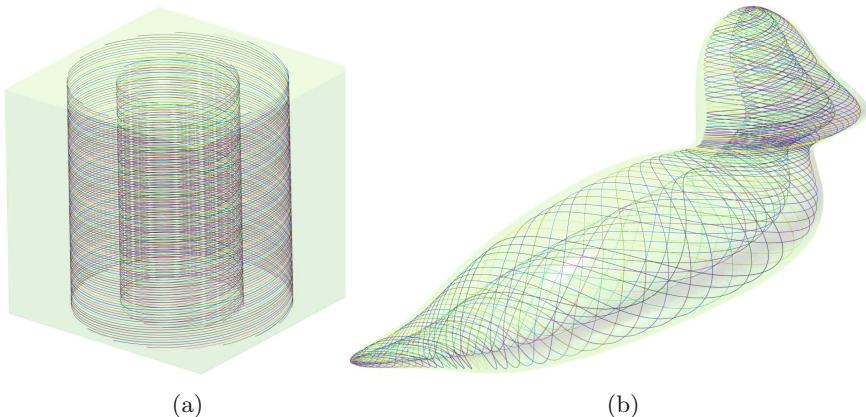


Fig. 10. Two sets of concentric helical curves are embedded in the domain of the trivariate (a) only to be curve-trivariate composed to yield the result shown in (b), in a trivariate duck model.

We have already shown that trivariate-trivariate composition is viable. One can also use trivariate-trivariate composition to construct volumetric micro-structures toward Isogeometric Analysis (IGA). Figure 11 shows two structures of trivariate B-spline functions, created using the presented trivariate-trivariate composition. Each trivariate in Fig. 11 is differently colored.

Finally, Fig. 12 shows a small-deformation linear elasticity analysis of an isotropic material (young modulus $E = 1$ and Poisson coefficient $\nu = 0.3$),

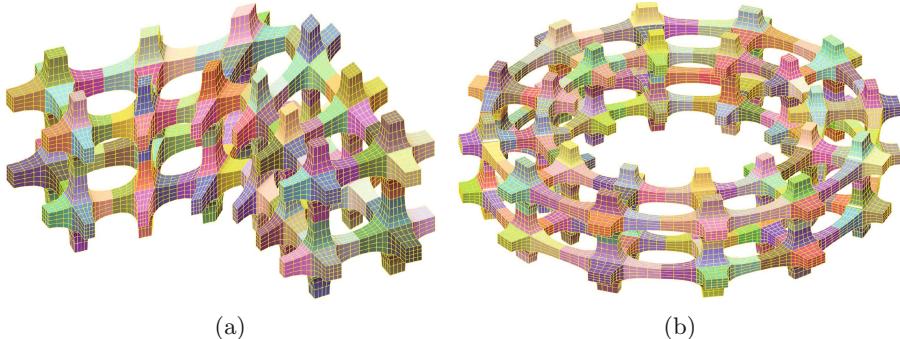


Fig. 11. Two examples of micro-structures consisting of trivariate B-spline functions (each trivariate in a different color) that were created using trivariate-trivariate composition, possibly toward Isogeometric Analysis (IGA). In (a), 168 trivariate Bézier elements of orders (4, 4, 7) are shown and in (b), 336 trivariate Bézier elements of orders (5, 5, 9) are presented. (Color figure online)

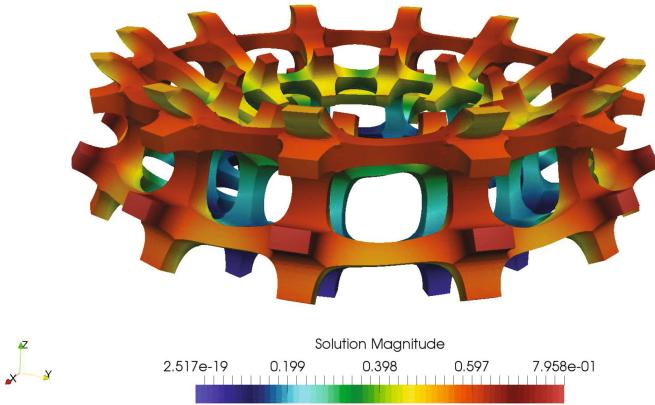


Fig. 12. The result of elasticity isogeometric analysis for the micro-structures from Fig. 11(b), using the igatools library [20].

using the igatools library [20], for the micro-structure in Fig. 11(b). The boundary conditions of the problem include a lower internal ring that is completely blocked and a Dirichlet boundary condition that is applied for the top faces of the upper external ring, that is moved vertically a quantity 0.5, but cannot move horizontally. This IGA problem was solved in less than two minutes, on a modern PC workstation.

4 Discussion, Limitations, and Future Work

The complexity of the result deserves some considerations. The composition operations in Fig. 1 employed a tile that consists of a cubic curve, a cubic by

quadratic surface, and a trivariate of degrees $(3, 3, 2)$. The mapping function was a trivariate Bézier of degrees $(1, 1, 3)$.

A direct mapping of control points will end up with the same degrees as the input. On the other hand, the smooth result, of Fig. 1(d), of the precise composition mapping, resulted in deformed curves of degree 15, deform surfaces of degrees $(15, 10)$ and deformed trivariates of degrees $(15, 15, 10)$.

Recall Eq. (1). Given a trivariate T of degrees (n_1, n_2, n_3) and a surface S of degrees (m_1, m_2) , the degrees of the composition surface $T(S)$ are

$$(n_1 m_1 + n_2 m_1 + n_3 m_1, n_1 m_2 + n_2 m_2 + n_3 m_2). \quad (7)$$

For the surface case of Fig. 1, Eq. (7) indeed yields the degrees of

$$(1 \times 3 + 1 \times 3 + 3 \times 3, 1 \times 2 + 1 \times 2 + 3 \times 2) = (15, 10).$$

Clearly, the degrees can be higher and as a second example, in Fig. 4, the tile consists of four surfaces of degrees (3×3) , and the trivariate knot is of degrees $(3 \times 3 \times 3)$. The composed surface elements are of degrees (27×27) (and lengths (109×163)). That said, and beyond the computation overhead that these high degrees poses, we have detected no instability difficulties for degrees below one hundred, including with rational forms. Rational input will yield rational output but in the same degrees as the polynomials case. Figure 13 exemplifies this observed stability. Given a unit cube polynomial trivariate, T , of a tri-order d and a rational cylinder surface of unit size, S , of bi-order d , we examined the error in the circularity of $T(S)$ (measured as the distance deviation from the axis of the cylinder) as a function of d . As can be seen in the figure, the error remains very small and its growth is approximately linear with the orders.

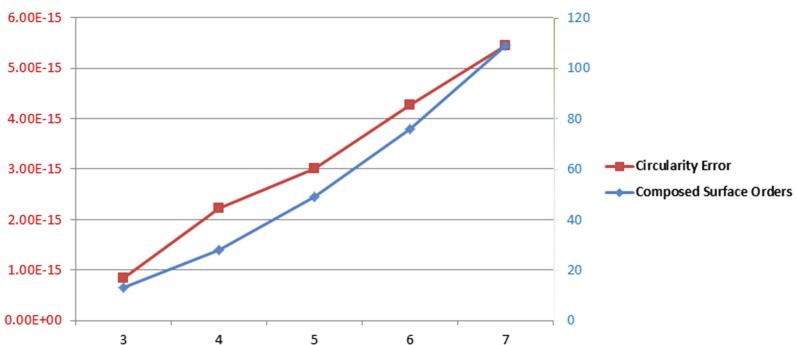


Fig. 13. Given a unit cube polynomial trivariate, T , of a tri-order d and a rational cylinder surface of unit size, S , of bi-order d , the error in the circularity of $T(S)$ is examined and shown in red as a function of d . Also shown is the bi-order of $T(S)$ in blue. (Color figure online)

Since the geometry that is synthesized in this work is typically deformed, the need for precise rational circular arcs is diminished. While we do support

compositions of rational forms, in all other presented examples in this work, polynomial forms were used.

All the micro-structure models presented in this work were created in seconds to minutes. Measured on an 2.8 GHz *i7* laptop running Windows 10, the example in Fig. 2 synthesized $256 = (4 \times 4 \times 16)$ tiles (each consisting of 6 bilinear surfaces) in a little over a minute. The example in Fig. 4(c) synthesized 47 tiles (each consisting of 4 helical surfaces) in around 20 s. Finally, the example in Fig. 11(b) synthesized 48 tiles (each consisting of 7 trivariate, 336 Bézier elements in all) in around 5 s.

Both inputs, the tile and/or the mapping trivariate, \mathcal{T} , can be singular, in which case the constructed micro-structure is likely to be singular as well. However, verifying the regularity of either the tile or \mathcal{T} is fairly simple by computing and bounding the magnitude of their Jacobian. For \mathcal{T} , this amounts to the (spline) product of:

$$|J| = \left\langle \frac{\partial \mathcal{T}}{\partial x} \times \frac{\partial \mathcal{T}}{\partial y}, \frac{\partial \mathcal{T}}{\partial z} \right\rangle,$$

and verifying that $|J|$ never vanish, for example, by verifying that all the coefficients of $|J|$ are of the same sign.

In all examples presented and due to \mathcal{T} , different local scale factors are applied to different tiles in the output geometry. This clear limitation stems from the fact that the deformation trivariate function is rarely isometric. One can only establish bounds on the different scale factors, by computing the field of the first fundamental form [2] of the deforming trivariate, as spline functions, and bound their range.

Because every tile is likely to be deformed a bit differently from other tiles, if some local properties are to be preserved, some extra measures must be taken into considerations. Consider the example in Fig. 8. While the original tile could have been synthesized using a highly accurate circular cross section, the mapped/deformed result in Fig. 8(f) is likely to violate that circularity. However, one can deform an axial curve of the tile's circular geometry, using curve-trivariate composition (i.e. Fig. 10), only to sweep a circular cross section through the deform axial curve, using a regular sweep surface operator. Here, the result will be stitches that are circular as precise as desired (and allowed by the sweep operation).

In this work, we uniformly paved tiles in the domain of the mapping FFD. However, and as already stated, especially for B-spline FFDs, this can result in unequally stretched tiles. While the notion of arc-length is difficult to extend from curves to surfaces and trivariate, one can still devise a scheme to try to equalize the stretch in the mapping \mathcal{T} , via a domain reparametrization, before the paving process takes place. Indeed, the internal parameterization of \mathcal{T} can have a grand affect on the distribution of the tiles in the structure and is a degree of freeform to further investigate and employ.

Having fairly complex tiles, paving and mapping them numerous times can require an intense amount of memory (and computing power). While unavoidable at times, one can consider a lazy synthesis of the geometry on the fly and

as needed in real time, possibly with the help of parallel computing. Further, one can take advantage of the inherent hierarchy and given local geometric operations (like slicing), converge rapidly and process/synthesize only these micro-structure's tiles that are active and affect the local geometry (intersect with the slicing plane). Similarly, if tiles are (trimmed) freeform shapes, their tessellation into polygons might also be done on the fly and by demand, again with the possible help of GPUs or parallel computing. This potential difficulty of large memory and computing needs is likely to play a role also in analysis of porous geometry, while part of the difficulties in the analysis might be alleviated via homogenization of the structure [3].

In all examples, by definition, the deforming function was a tensor product trivariate and the tiles were cuboids aligned along the main axes, in a cube-like topology. In [16], use of trimmed trivariate was already proposed and herein, in order to support \mathcal{T} mapping using trimmed trivariate, tiles that intersect with the trimming domain must be properly pruned or sealed. Further, considering an interior trimming boundary in some volumetric model, between two trimmed trivariate must also be addressed, following [16]. Conceptually, there might be no need to prune tiles that are completely inside the volumetric model, even when they cross interior trimming boundaries, while matching the boundaries of the tiles along these interior trimming boundaries will be required. Proper, whatever that means, treatment of tiles in such a trimmed volumetric environment is still an open question, including the proper management of continuity across interior trimming surfaces.

Non-tensor product FFDs were already proposed, for example, in [4], as Extended FFDs. Use of Extended FFDs or other mappings, instead of the tensor product trivariate employed herein, will allow one to support micro-structures that are not necessarily of cuboid topology. Further, the tiles themselves are not confined to cube-like topology, and any tile that periodically paves 3-space can be used. Examples include hexagonal prisms or tetrahedra tiles, possibly embedded in a large hexagonal prism or tetrahedra deformation function. Alternatives to tensor product trivariate should be explored as mapping function and those can include splines over general triangulations or box-splines. Moreover, any tiling of 3-space can be used and semi-regular tiling, where two (or more) differently shaped tiles are employed together, is another example.

The presented micro-structures' construction scheme can be further refined and improved in additional directions. Attributes like colors or texture can be mapped to the resulting geometry where the attributes' specifications can either be local, coming from the tile itself and repeated for all tiles, or be global as a specification over the mapping trivariate.

Herein, the same tile was used throughout the pavement of a deforming trivariate. Alternatively, one can select each tile out of a (predetermined or created on the fly) random (set of) tile, resulting in a randomly looking porous geometry. Further, if certain physical constraints apply, such as local stress fields, the synthesized tiles can obey such constraints and locally adapt their shape to optimally satisfy these constraints while preserving continuity condi-

tions between tiles. The preservation of continuity can be performed incrementally and on the fly, by using, for random tile at indices (ijk) , the boundary conditions of previously constructed random tiles $(i-1, j, k)$, $(i, j-1, k)$, and $(i, j, k-1)$, if any, along their shared boundaries.

5 Conclusions

In this work, we have presented a purely geometric modeling constructor for the synthesis of complex, porous or micro-structure objects, by drawing from FFD techniques, possibly recursively. We have presented constructors using tiles that are either polygonal or spline based, as curves or (trimmed) surfaces and (trimmed) trivariates. If the input geometry is precise in the form of B-spline curves, (trimmed) surfaces, or (trimmed) trivariates, the mapped output will be precise as well, to within machine precision, also in the same form of B-spline shapes, albeit typically of higher degrees. The successful utilization of the presented micro-structure modeling constructor in application areas such as bio-engineering, mission critical engineering, or alternatively jewelry design, a few fields we already foresee as viable, or other areas, is yet to be seen.

All the implementation, include source code, is available as part of the IRIT geometric modeling environment (<http://www.cs.technion.ac.il/~irit>). Further, most micro-structure examples presented in this work are available, as obj (for polygons) and iges (for surfaces) files, in <http://www.cs.technion.ac.il/~gershon/site/modeling.html>.

Acknowledgments. This research was supported in part by the ISRAEL SCIENCE FOUNDATION (grant No. 278/13). I also like to thank Boris van Sosin for his help in implementing the trivariate-trivariate composition operator.

The IGA of the model in Fig. 12 has been performed with the help of Pablo Antolin (EPFL Lausanne), Annalisa Buffa (EPFL Lausanne and IMATI-CNR Pavia), Massimiliano Martinelli (IMATI-CNR Pavia); Giancarlo Sangalli (University of Pavia and IMATI-CNR Pavia)

References

1. Armillotta, A., Pelzer, R.: Modeling of porous structures for rapid prototyping of tissue engineering scaffolds. *Int. J. Adv. Manuf. Technol.* **39**, 501–511 (2008)
2. Carmo, M.P.D.: Differential Geometry of Curves and Surfaces. Prentice-Hall, Englewood Cliffs (1976)
3. Chen, J., Shapiro, V.: Optimization of continuous heterogeneous models. In: Pasko, A., Adzhiev, V., Comninou, P. (eds.) Heterogeneous Objects Modelling and Applications. LNCS, vol. 4889, pp. 193–213. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-68443-5_8](https://doi.org/10.1007/978-3-540-68443-5_8)
4. Coquillart, S.: Extended free-form deformation: a sculpturing tool for 3D geometric modeling. In: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, vol. 24, pp. 187–196. ACM Press, August 1990

5. Coquillart, S., Jancne, P.: Animated free-form deformation: an interactive animation technique. In: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, vol. 25, pp. 23–26. ACM Press, July 1991
6. DeRose, T.D., Goldman, R.N., Hagen, H., Mann, S.: Functional composition algorithms via blossoming. *ACM Trans. Graph.* **12**(2), 113–135 (1993)
7. Elber, G.: Free form surface analysis using a hybrid of symbolic and numerical computation. Ph.D. thesis, University of Utah (1992)
8. Elber, G.: Geometric deformation-displacement maps. In: The Tenth Pacific Graphics, pp. 156–165, October 2002
9. Elber, G.: Geometric texture modeling. *IEEE Comput. Graph. Appl.* **25**(4), 66–76 (2005)
10. Elber, G.: Constructing porous geometry. In: FASE 2016, June 2016
11. Ellens, M.S., Cohen, E.: An approach to C^{-1} and C^0 feature lines. In: Mathematical Methods for Curves and Surfaces, pp. 121–132 (1995)
12. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design. Academic Press Professional, Boston (1993)
13. Farouki, R.T., Rajan, V.T.: Algorithms for polynomials in Bernstein form. *Comput. Aided Geom. Des.* **5**(1), 1–26 (1988)
14. Griebsmair, J., Purgathofer, W.: Deformation of solids with trivariate B-splines. *Eurographc* **89**, 137–148 (1989)
15. MacCracken, R., Joy, K.I.: Free-form deformations with lattices of arbitrary topology. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 181–188. ACM Press (1996)
16. Massarwi, F., Elber, G.: A B-spline based framework for volumetric object modeling. *Comput. Aided Des.* **78**, 36–47 (2016)
17. Morken, K.: Some identities for products and degree raising of splines. *Constr. Approx.* **7**(1), 195–208 (1991)
18. Paik, K.L.: Trivariate B-splines. MSc. Department of Computer Science, University of Utah (1992)
19. Pasko, A., Fryazinov, O., Vilbrandt, T., Fayolle, P.-A., Adzhiev, V.: Procedural function-based modelling of volumetric microstructures. *Graph. Models* **73**(5), 165–181 (2011)
20. Pauletti, M.S., Martinelli, M., Cavallini, N., Antolin, P.: Igatools: an isogeometric analysis library. *SIAM J. Sci. Comput.* **37**(4), 465–496 (2015)
21. Piegl, L., Tiller, W.: The NURBS Book. Springer, Heidelberg (1997)
22. Porumbescu, S.D., Budge, B., Feng, L., Joy, K.I.: Shell maps. *ACM Trans. Graph.* **24**(3), 626–633 (2005)
23. Schein, S., Elber, G.: Discontinuous free form deformations. In: Proceedings of Pacific Graphics 2004, pp. 227–236 (2004)
24. Schroeder, C., Regli, W.C., Shokoufandeh, A., Sun, W.: Computer-aided design of porous artifacts. *Comput. Aided Des.* **37**, 339–353 (2005)
25. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. *Comput. Graph.* **20**, 151–160 (1986)
26. Xiao, F., Yin, X.: Geometry models of porous media based on Voronoi tessellations and their porosity-permeability relations. *Comput. Math. Appl.* **72**, 328–348 (2016)