

Constructing Porous Geometry

Gershon Elber

Dept. of Computer Science, Technion – IIT, Haifa 32000, Israel

gershon@cs.technion.ac.il

Abstract

This work introduces a modeling constructor for porous geometry via surface-trivariate compositions. By using 2-manifold (porous) surface tiles and paving them multiple times inside the domain of a 3-manifold trivariate function, precise yet general, porous and watertight, geometry might be constructed, via composition. The 2-manifold tiles are demonstrated to be either polygonal meshes or (a set of) B-spline surfaces whereas the 3-manifold trivariate is either a Bézier or a B-spline function.

After laying down the theoretical foundations, we demonstrate the power of this constructor over some models, only to present some 3D printed tangible examples.

1 Introduction and related work

The idea of freeform deformations (FFD) was introduced around thirty years ago [11] as a global deformation mapping, $\mathcal{T} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, and was based on trivariate tensor-product Bézier vector functions. Since then, a large body of work was presented on a variety of FFD techniques, including extensions to B-spline support [7] and the use of FFDs [3] as an animation tool. While in general, FFDs map a box-shaped domain into a deformed-box in Euclidean space, [2] introduces Extended FFDs to form a deformation that better resembles the shape of the input model. [2] 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 [8].

Other, more recent variations, even suggested variations of FFDs that remove some topological restrictions from the deformed object. [6] suggested teared surfaces that incorporated non-iso-parametric curves of C^{-1} discontinuity inside B-spline surfaces. Similarly, [10] suggested the exploitation of discontinuous FFDs to induce tears in the deformed models for animation and surgery incision simulations.

Interestingly enough and while the body of FFD work is significant, FFDs were never seen as modeling tools. Almost exclusively, FFDs were applied to an input model, resulting a modified model. In this work, we combine the general FFD's idea with a surface detailing technique, such as [5], into a modeling constructor of porous geometry. In [5], surface detailing geometry is encoded over 2-manifolds in 3D, with examples like scales and thorns.

The existing body of FFD work updates or augments an existing model as are surface detailing techniques. Herein, we propose a simple change of concept, and introduce a variation of FFD as a *modeling constructor*. The Trivariate function \mathcal{T} can be of any general shape. Techniques to build trivariate functions are known and almost any 2-manifold surface model can be made into a 3-manifold trivariate using constructors like ruled volumes or volumes of revolution. Then, the 3D domain of \mathcal{T} will be paved with simple porous elements, aka tiles, toward the construction of a periodically looking complex, porous model, in the general shape of \mathcal{T}

The rest of this work is organized as follows. Section 2 presents the different stages of this proposed FFD variant and the necessary tools. In Section 3, some examples and results are presented, only to conclude, in Section 4.

2 Algorithm

Let \mathcal{T} be a trivariate Bézier or B-spline 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 $B_{i_1, n_1}(u)$ are the Bézier or B-spline basis functions.

The mapping of a triangle through \mathcal{T} can be approximated by simply mapping its three vertices, V_k , $k = 0, 1, 2$, as $\mathcal{T}(V_k)$. Such a mapping is precise only at the vertices and some FFD related work aimed to control this error, possibly by dividing the triangles to smaller ones, adaptively. A surface tile that is a polygonal mesh can then be mapped one triangle at a time. Note that general n-gons, such as rectangles and pentagons, are not necessarily going to remain planar after the mapping of their vertices through \mathcal{T} and hence, typically, the polygons in the tile are refined to consists of triangles only.

Now consider a surface tile that is a Bézier or a B-spline surface $S(u, v) = (S_x(u, v), S_y(u, v), S_z(u, v))$:

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

One can map the control points, Q_{j_1, j_2} , of S through \mathcal{T} but this mapping of S , is only an approximation, much like the polygonal tile case, where only the vertices were mapped. Further, continuity will not be preserved and the geometry will not follow the curvature induced by \mathcal{T} (See also Figure 1 (c)). Alternatively, a precise mapping of S through \mathcal{T} (See also Figure 1 (d)) can be computed using composition [4]:

$$\begin{aligned} \mathcal{T}(S(u, v)) &= \mathcal{T}(S_x(u, v), S_y(u, v), S_z(u, v)) \\ &= \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} P_{i_1, i_2, i_3} B_{i_1, n_1}(S_x(u, v)) B_{i_2, n_2}(S_y(u, v)) B_{i_3, n_3}(S_z(u, v)), \end{aligned} \quad (3)$$

which amounts to the computation of products of terms in the form of $B_{i_1, n_1}(S_x(u, v))$. If B_{i_1, n_1} is a polynomial (Bézier) function and $Q_{j_1, j_2} = (q_{j_1, j_2}^x, q_{j_1, j_2}^y, q_{j_1, j_2}^z)$, then:

$$\begin{aligned} B_{i_1, n_1}(S_x(u, v)) &= \binom{n_1}{i_1} S_x(u, v)^{i_1} (1 - S_x(u, v))^{n_1 - i_1} \\ &= \binom{n_1}{i_1} \left(\sum_{j_1=0}^{m_1} \sum_{j_2=0}^{m_2} q_{j_1, j_2}^x B_{j_1, m_1}(u) B_{j_2, m_2}(v) \right)^{i_1} \\ &\quad \left(1 - \left(\sum_{j_1=0}^{m_1} \sum_{j_2=0}^{m_2} q_{j_1, j_2}^x B_{j_1, m_1}(u) B_{j_2, m_2}(v) \right) \right)^{n_1 - i_1}, \end{aligned} \quad (4)$$

or merely products and summations of Bézier or B-spline basis functions. See [1] for more on the computation of products and summations of splines.

Interesting enough, if \mathcal{T} is a B-spline trivariate, S cannot cross knot lines in \mathcal{T} or otherwise S must be divided along the knot lines of \mathcal{T} into smaller, not necessarily rectangular surface patches. Further, those new patches must be again divided into rectangular patches, a feasible process that is unfortunately far more difficult and that also affects the regularity and continuity of the tiles. Hence, herein we limit ourselves to tiles that cross no knot lines in \mathcal{T} .

Having the ability to compute $\mathcal{T}(S)$ for both polygonal and surface tiles, we consider periodic tiles in 3D that pave the domain of \mathcal{T} ($c_x \times c_y \times c_z$) times. See also Figure 1. A tile is considered periodic if

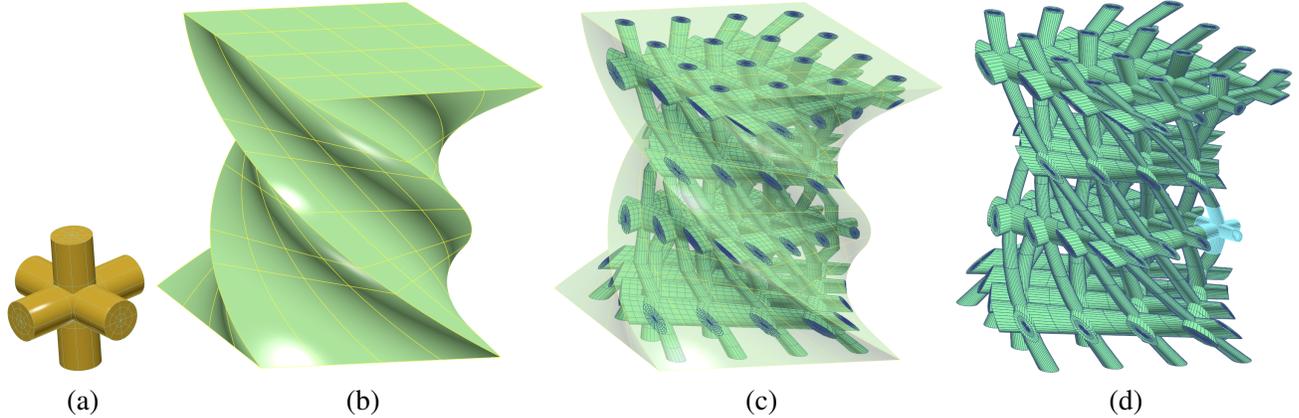


Figure 1 : A simple example of a tile S with three cylindrical surfaces in the shape of a 3D cross (a) paved $(4 \times 4 \times 4)$ times in the domain of the given trivariate \mathcal{T} (b). (c) presents the result when mapping just the control points of the surfaces in the cross resulting in a piecewise C^1 geometry (shown embedded in the transparent trivariate) whereas (d) shows the precise smooth composition result of $\mathcal{T}(S)$ (note one tile is highlighted in cyan).

the shapes of the openings (holes) in the tile on the $x_{min}, y_{min}, z_{min}$ boundaries match the shapes of the openings in the tile on the $x_{max}, y_{max}, z_{max}$ boundaries, respectively. If the given tile is either closed or open and periodic, a watertight model can be formed. If the matched openings offer a smooth transition or even higher continuity (and \mathcal{T} is sufficiently continuous), a smooth periodically constructed model will form. In the next section, the power of this modeling constructor is fully revealed and demonstrated.

3 Results and Examples

A modeling constructor based on FFD can be powerful. It enables the fabrication of geometry that is very difficult to construct in alternative ways. Figure 2 shows a polygonal tile of over 1000 polygons paved $(4 \times 4 \times 20)$ times in the domain of a B-spline trivariate in the shape of a duck. The duck was provided as a sweep surface and converted into a trivariate using volumetric Boolean sum over six faces, four around the sweep surface and two near the head and the tail. The domain of the same duck is paved, in Figure 3, with B-spline surfaces' tiles, only to be using precise surface-trivariate composition computation, following Equation (3). Six bilinear B-spline surfaces defined this hollowed tile. The result of the composition is shown in Figure 3 (b) whereas Figure 3 (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 Figure 3 (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 watertight but we still need to close boundary openings along the boundary of \mathcal{T} . This closure is simple to achieve - every tile that is a boundary tile in some direction, in the pavement of the trivariate's domain, will be clipped and sealed with the plane of that boundary, possibly using a Boolean set operation. Figure 3 (d) shows a watertight porous model that resulted from applying this boundary clipping operation to the model in Figure 3 (c).

In Figure 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 again used to convert the sweep surface to the trivariate that is shown in Figure 4 (b). The tile in Figure 4 (a) consists of four bicubic helical B-spline surfaces, constructed using algebraic sum [9] between

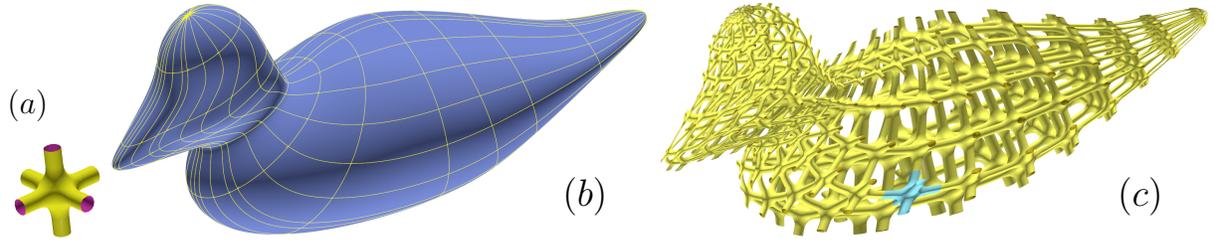


Figure 2: A polygonal 3D cross (a) is tiled ($4 \times 4 \times 20$) times in a B-spline trivariate of orders ($3 \times 3 \times 4$) and lengths ($6 \times 5 \times 17$) in the shape of a duck (b), resulting in (c) (note one tile is highlighted in cyan).

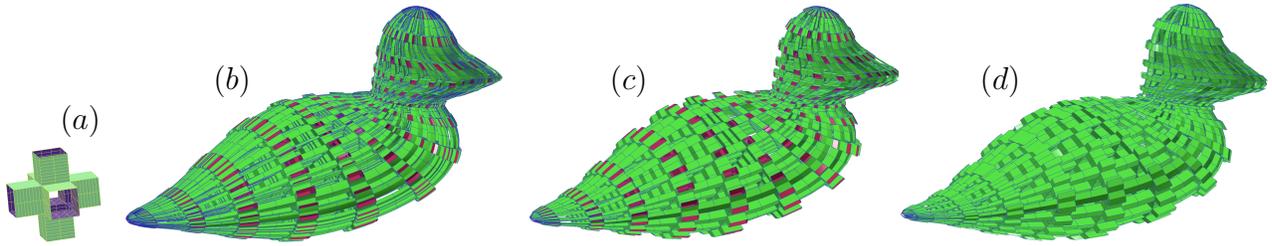


Figure 3: Six bilinear B-spline surfaces form the tile in (a) that paves the domain of a trivariate in the shape of a duck. (b) presents the smooth and precise surface-trivar composition. In (c), the tile is converted to polygons and vertices are mapped through the trivariate, resulting in C^1 discontinuities (note the belly area). Note the interior of the resulting surface is exposed, in magenta, in (b) and (c). (d) shows the result of the boundary clipping, creating a watertight model. See also Figure 6 (left).

a quarter of a helix 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 Figures 4 (c) and (d). Figures 4 (e) to (h) shows the same trivariate paved using different 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 a fairly complex tile is exploited. The tile, in Figure 5 (a), is paving the domain of a torus trivariate in Figure 5 (b) and Figure 5 (c), using two different pavement's resolutions.

Finally, and as a proof for the viability of the constructed models and their watertightness, Figure 6 presents two of the presented examples 3D-printed using additive manufacturing.

4 Conclusions

In this work, we have presented a modeling constructor for creating complex, porous, geometry, exploiting FFD techniques. If the input geometry is precise in the form of B-spline surfaces, the output will be precise as well, to within machine precision, also in the form of B-spline surface. However, the output B-spline will be of higher degrees due to the composition operations. For example, the bilinear B-spline surfaces in the tile of the duck example of Figure 3 (b) are mapped to new surfaces of orders (6×6) and mesh size of (21, 6) and the helical bicubic surfaces (of mesh size (10×9)) in the tiles of the knot example of Figure 4 are mapped to new surfaces of orders (28×28) and mesh size of (109, 163). While fairly high orders, all computations in the presented examples took from seconds to dozens of seconds to complete, in practice, and with no observed instability or inaccuracy.

In this work, we have presented constructors using tiles that are either polygonal or spline based. The

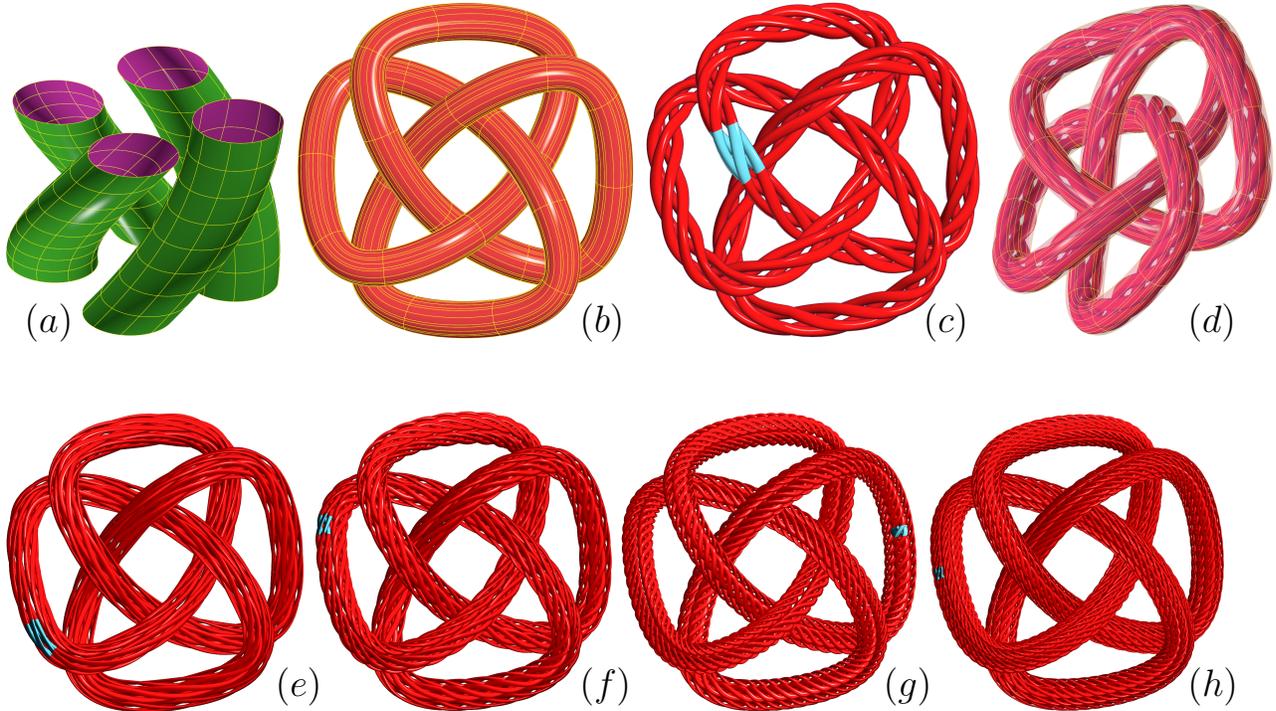


Figure 4: A tile consisting of four B-spline helical-looking surfaces (a) is paved ($1 \times 1 \times 47$) times in the domain of a B-spline trivariate of orders ($4 \times 4 \times 4$) and lengths ($4 \times 4 \times 50$) in the shape of a knot (b), resulting in (c) (note one tile is highlighted in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

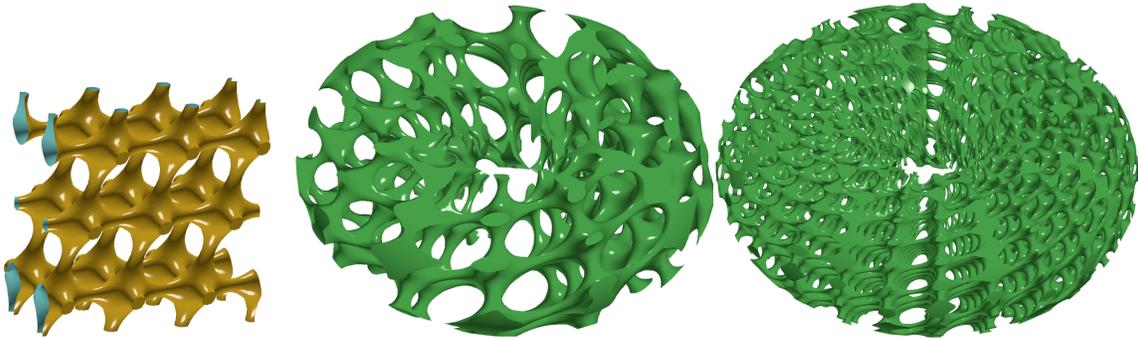


Figure 5: A fairly complex polygonal periodic tile consisting of around 20k polygons in (a) is paving the domain of a torus trivariate using two different resolutions in (b) and (c). See also Figure 6 (b).

same concepts can be employed for tiles that formed out of curves, having a set of curves as the result of a curve-trivariate compositions. Alternatively, the tiles can consist of trimmed surfaces, where only the tensor product B-spline surfaces undergo through the mapping \mathcal{T} and the trimming information is simply copied as is. Finally, the input tile can also be a trivariate, resulting in a porous trivariate model, using trivariate-trivariate composition.

The presented geometry construction scheme can be further refined and improved in other 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 and repeated for all tiles, or be global as a specification over the



Figure 6: A 3D printed porous duck (left) from Figure 3 and a porous torus (right) from Figure 5. Printing courtesy of Stratasys.

trivariate itself.

Herein, the same tile was used throughout the pavement of a trivariate, 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.

5 Acknowledgments

This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No.278/13).

References

- [1] COHEN, E., RIESENFELD, R., AND ELBER, G. *Geometric modeling with splines: an introduction*. 2001. AK Peters, Ltd., Massachusetts, USA, 2001.
- [2] 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* (Aug. 1990), vol. 24, ACM Press, pp. 187–196.
- [3] COQUILLART, S., AND JANCNE, P. Animated free-form deformation: an interactive animation technique. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques* (July 1991), vol. 25, ACM Press, pp. 23–26.
- [4] DEROSE, T. D., GOLDMAN, R. N., HAGEN, H., AND MANN, S. Functional composition algorithms via blossoming. *ACM Trans. Graph.* 12, 2 (Apr. 1993), 113–135.
- [5] ELBER, G. Geometric texture modeling. *IEEE CG&A* 25 (July-August 2005), 66–76.
- [6] ELLENS, M. S., AND COHEN, E. An approach to c^{-1} and c^0 feature lines. *Mathematical Methods for Curves and Surfaces* (1995), 121–132.
- [7] GRIESSMAIR, J., AND PURGATHOFER, W. Deformation of solids with trivariate b-splines. In *Eurographic 89* (1989), pp. 137–148.
- [8] MACCRACKEN, R., AND JOY, K. I. Free-form deformations with lattices of arbitrary topology. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996), ACM Press, pp. 181–188.
- [9] PIEGL, L., AND TILLER, W. *The nurbs book*. 1997.
- [10] SCHEIN, S., AND ELBER, G. Discontinuous free form deformations. In *Proceedings of Pacific Graphics 2004* (October 2004), pp. 227–236.
- [11] SEDERBERG, T. W., AND PARRY, S. R. Free-form deformation of solid geometric models. *Computer Graphics* 20 (Aug 1986), 151–160.