

HYPERSEEING

The Publication of the International Society of the Arts, Mathematics, and Architecture

SUMMER 2022
www.isama.org

The Proceedings of the SMI-SCULPT 2022

Turing Reaction-Diffusion Sculptures

*Special
Issue on
Shape
Modeling
International
2022*

27-29
June
2022
Online



HYPERSEEING
Special Issue on SMI-SCULPT 2022

Shape Modeling International 2022
**Shape Creation Using Layouts,
Programs, & Technology (SCULPT) Event**

*Twenty first Interdisciplinary Conference of the International
Society of the Arts, Mathematics, and Architecture*

Online Event
June 27-29, 2022

SMI Conference Chairs

Ying He (Nanyang Technological Institute, Singapore)
Stefanie Hahmann (Inria, France)
Yang Liu (Microsoft Research Asia, China)

SCULPT Paper Chairs

Oleg Fryazinov (Bournemouth University, UK)
Carlo H. Séquin (UC Berkeley, USA)

SCULPT Communication Chair

Ergun Akleman (Texas A&M University, USA)

Steering Committee: Ergun Akleman, Loic Barthe, Karina Rodriguez Echavarria, Oleg Fryazinov, Guiseppa Patane, Konrad Polthier, Carlo Sequin,

FASE Program Committee

Valery Adzhiev
Bournemouth University

Ergun Akleman
Texas A&M University

Loic Barthe
IRIT - Université de Toulouse

Michael Birsak
King Abdullah University of Science and
Technology

Daniela Cabiddu
CNR IMATI

Nathan Carr
Adobe Systems Inc.

Raphaëlle Chaine
LIRIS Université Lyon 1

Gershon Elber
Technion- Israel Institute of Technology

Bianca Falcidieno
Consiglio Nazionale delle Ricerche (CNR)

Oleg Fryazinov
Bournemouth University

Xin Li
Louisiana State University

James Mallos
Sculptor

Alexander Pasko
Bournemouth University, UK

Giuseppe Patanè
CNR-IMATI

Helmut Pottmann
Vienna University of Technology

Karina Rodriguez
University of Brighton

Henry Segerman
Oklahoma State University

Carlo H. Séquin
University of California, Berkeley

Shinjiro Sueda
Texas A&M University

Remco Veltkamp
Utrecht University

Charlie Wang
Delft University of Technology

SCULPT Supported by

Bournemouth University
Texas A&M University
UC Berkeley

CONTENTS

Proceedings of Fabrication and Sculpting Event

Author(s)	Title	Pages
REGULAR PAPERS		
Carlo H. Séquin	Partitioning and Assembly of Gosper Sculptures	13-28
Q. Youn Hong, Youngjin Park, and Gershon Elber	Conformal Parametric Microstructure Synthesis for Boundary Representations	29-45
Emma Fogarasi, and Oleg Fryazinov	3D Printing Hypercomplex Fractals	47-56
SHORT PAPERS		
Alyn Rockwood, and Irene Gironacci	Turing Reaction-Diffusion - a World in a Grain of Sand	59-64

Preface

History of Shape Creation Using Layouts, Programs, & Technology (SCULPT) Event

The SCULPT started as an experiment in expanding the scope of shape modeling international (SMI) conference in 2012. It was originally called Fabrication and Sculpting Event (FASE). We also had another FASE event in SMI'2013. There were very positive responses to the FASE papers and presentations in both 2012 and 2013. Although we skipped FASE in SMI'2014, based on the success of earlier events, we continued the FASE event from 2015 to 2021 as a part of SMI conference.

In 2013, Nat Friedman, the chair of the International Society of the Arts, Mathematics, and Architecture (ISAMA), asked me if we could organize the event as an annual ISAMA conference. I presented the idea to the SMI steering committee. The Committee unanimously agreed with the suggestion. As a result, this event can now be considered also as the Twentieth Interdisciplinary Conference of ISAMA.



The ISAMA conference has a rich history. The first Art and Mathematics Conference (AM 92) was organized by Nat Friedman at SUNY-Albany in June, 1992. This conference was followed by annual conferences AM93-AM97 at Albany and AM 98 at the University of California, Berkeley, co-organized with Carlo Séquin. ISAMA was founded by Nat Friedman in 1998 along with the ISAMA publication *Hyperseeing* co-founded with me in 2006. In addition, the Art/Math movement has taken off with the formation of many additional conferences and organizations. In particular, we mention the very successful Bridges conference organized by Reza Sarhangi in 1998 and the excellent Bridges Proceedings. The significance of the art/math movement is now recognized internationally and in particular by the extensive art/math exhibit at the annual Joint Mathematics Meeting of the American Mathematical Society and the Mathematical Association of America organized by Robert Fathauer.

In 2022, we decided to name the event as SCULPT to communicate the concept better. It is a short form of Shape Creation Using Layouts, Programs, & Technology (SCULPT). The main difference with other math/art conferences is that SCULPT focuses solely on physical realization of 3D shapes. We invite submissions mainly from practitioners such as sculptors and architects to describe their methods. We expect that such papers and the following discussions can provide new problems, issues, and questions for theoretical shape modeling research.

Ergun Akleman

Editor, *Hyperseeing*

Preface

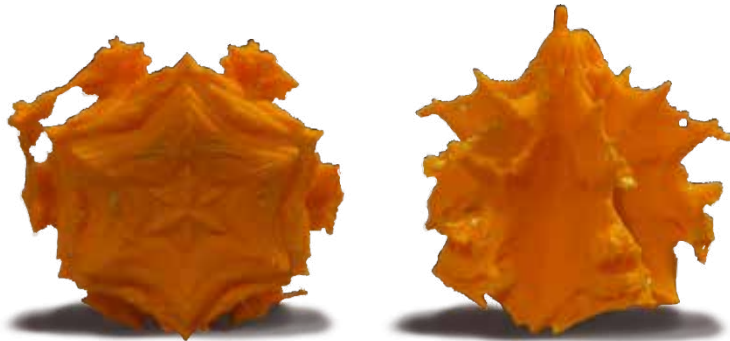
Shape Creation Using Layouts, Programs, & Technology (SCULPT) 2022

There are at least two aspects to shape modeling: theoretical and practical. The mathematical and theoretical aspects of shape modeling have traditionally been supported by the SMI conference. With the Fabrication and Sculpting Event (FASE) our goal is to include more hands-on, application-oriented ways by designers and sculptors who construct sophisticated real-world objects.

FASE has its own program committee, and the accepted papers are published in Hyperseeing. With FASE, we hope to attract practitioners who might usually be less inclined to write papers containing formal algorithms or mathematical proofs, but who nevertheless have important things to say that are of interest to the shape modeling community and who also might provide visually stimulating material.

For this year's Fabrication and Sculpting Event, we solicited papers that pose new questions and motivate further research in designing, fabrication and sculpting. Topics should be useful, for example, in the following areas: Fabrication of digital models, Advanced manufacturing techniques such as additive manufacturing, laser cutting or CNC milling, Interactive or procedural design of manufacturable shapes, Interconnections of complex modeling and fabrication processes, visually stimulating fabrication techniques or printed structures.

Thus, the scope of FASE is the intersection of shape modeling and fabrication methods/algorithms, and papers may focus on both the digital/theoretical and the physical domain or just one of these domains – as long as the connection to the other domain is clear. It is not a requirement that the techniques presented in the paper involve computation as such, but they need to have a clear algorithmic or mathematical element.



We received six submissions this year and three of them were accepted as twelve-page or longer full papers, and another one were accepted as short papers eight pages or shorter. The five accepted papers span a wide range of topics and views on the fabrication process of various artistically interesting artifacts. We wish to thank the authors and the reviewers for their participation in the SMI/ISAMA 2022 SCULPT Event. We hope that new ideas and partnerships will emerge from the FASE papers that can offer a glimpse into a much larger territory and the event can enrich interdisciplinary research in Shape Modeling. We hope that the attendees of SMI 2022 will enjoy this event of the conference.

Oleg Fryazinov, and Carlo Séquin

SCULPT Papers chairs

FULL PAPERS



Partitioning and Assembly of Gosper Sculptures

Carlo H. Séquin

University of California, Berkeley; sequin@berkeley.edu

Abstract

This paper addresses the technical challenges of physically realizing some tubular geometrical sculptures, using two examples from opposite ends of a spectrum of possible Gosper sculptures. A first example concerns a large-scale metal sculpture of a *Gosper-Ball* with only 38 tubular segments, but which still exhibits the characteristics of a 3D Gosper curve quite nicely. The challenge is how to accurately assemble the 38 tubular pieces while realizing the required angles at all 38 joints. The paper presents a judicious partitioning of the overall sculpture and some simple jigs that help to put the different sub-assemblies together.

The second example concerns 3D-printing of desk-top models of complex *Gosper-Globes* with up to 756 segments, where a variant of a 2D plane-filling Gosper curve is constrained to a spherical shell. The lengthy tubular loop representing the Gosper curve requires a lot of support material during the additive fabrication phase. This support material is difficult to remove from the inside of the spherical shell. Partitioning the shell into a few smaller domains reduces this problem without unduly complicating the assembly of the overall sculpture.

Introduction and Previous Work

Taking a self-similar, 2D plane-filling curve, such as the Hilbert curve [10] or the Gosper curve (Fig.1a) into the third dimension offers interesting mathematical design challenges and can result in attractive geometrical sculptures [3]. For the last two years, I have been specifically interested in transforming variants of the Gosper curve [9][12] into attractive 3D geometrical sculptures [4] (Fig.1c). Making a 3D space-filling Gosper-like curve (Fig.1b) has its own geometrical and algorithmic challenges, which have been discussed elsewhere [8][4][5]. Physically realizing such sculptures, either in the form of a 3D-print with several hundred cylindrical segments, or as a human-scale sculpture made from metal tube segments, poses additional challenges. Some of these problems can be addressed by judiciously partitioning the overall sculpture into a few sub-assemblies that are relatively easy to fabricate, and which then are not too difficult to compose into the final sculpture. In this study, I use two examples from opposite ends of a spectrum of possible Gosper sculptures to discuss these fabrication challenges – in accordance with the stated interests of the SCULPT track.

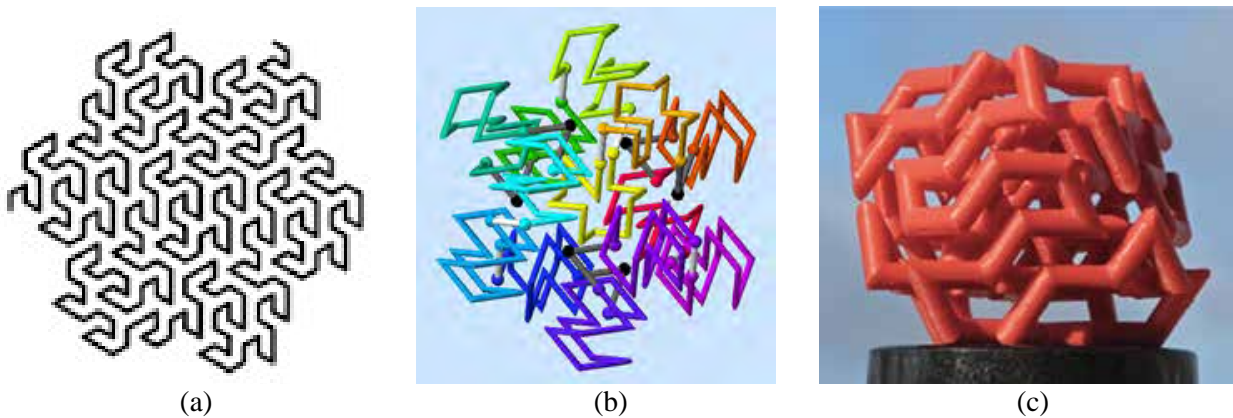


Figure 1: (a) 2D Gosper curve; (b) a 3D Gosper curve; (c) a small Gosper sculpture.

The first example concerns a *Gosper-Ball* of low complexity with only 38 segments made from metal tubes. Here the main problem is how to join the pre-cut metal tube segments while accurately realizing the required angles at all the joints, so that the overall sculpture will close nicely into a symmetrical loop. The first step is to pick the right sub-structures that can be assembled easily with good accuracy because they contain several co-planar segments. These sub-assemblies are then joined with simple, helpful jigs that assist in implementing proper angles and proper spacings.

The second example concerns 3D-printing of *Gosper-Globes*, where a variant of a 2D plane-filling Gosper curve is constrained to a spherical shell. I discuss making desk-top models of complex *Gosper-Globes* with up to 756 segments on low-end 3D-printers. The thin, lengthy tubular loop needs a lot of support material during the additive fabrication phase, which then is difficult to remove from the inside of the spherical shell. Partitioning the shell into a few smaller domains reduces this problem without unduly complicating the assembly of the overall sculpture.

Metal Sculpture: *Gosper-Onion_38*

Gosper-Balls fill a roughly spherical domain with a 3D polyline that tries to capture the basic characteristics of a 2D Gosper curve. In one approach, the Gosper strand visits the atomic sites of a face-centered cubic (FCC) lattice by making equal-size, nearest-neighbor steps. The FCC lattice is the lattice that allows densest sphere-packing. Figure 1c shows such a *Gosper-Ball* covering 140 FCC lattice sites.

Chris Ohler, who has created several abstract geometrical sculptures in metal [2], is interested in building a Gosper sculpture from metal tubing. When planning to make a first Gosper sculpture from individual metal tube segments, it is prudent to keep it rather simple. This raises the question: What is the minimal number of tube segments that can still present the characteristics of a 3D Gosper curve? One promising approach is to place some typical, hexagon-shaped “*Gosper-Claws*” in the outermost, peripheral layer of the sculpture, where these features can easily be seen, and to construct a small inner core that holds these outer elements in place. I call this the *Gosper-Onion* approach [4]. Working on the FCC lattice, I start with a core of six vertices on the corners of an octahedron (Fig.2a). I then surround this core with a second shell based on a truncated octahedron. This second shell has 24 atomic sites to define its convex hull and eight more atoms in the centers of the eight hexagonal faces (Fig.2b). This *Onion-Shell* approach can be continued. The third shell would have a total of 102 atomic sites (Fig.2c).

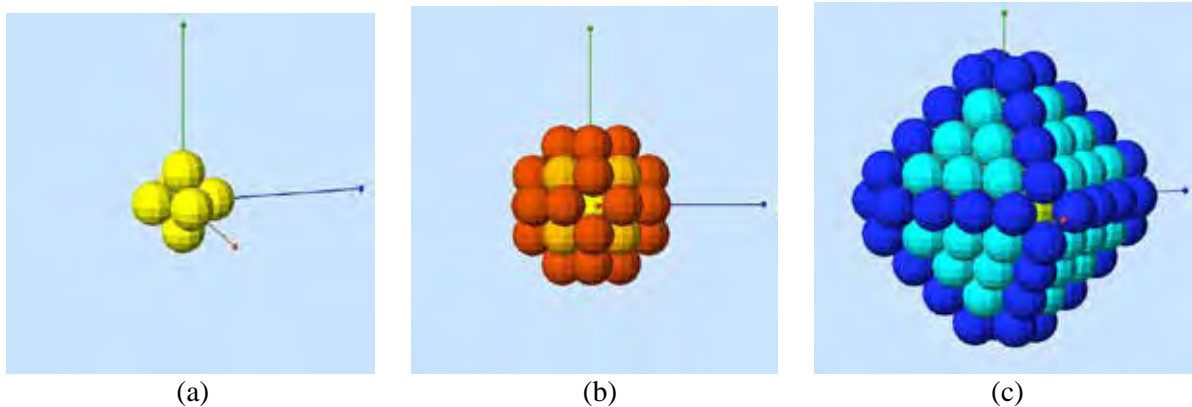


Figure 2: (a) 6 atoms in shell 1; (b) $24+8=32$ atoms in shell 2; (c) $6+48+48=102$ atoms in shell 3.

Various Design Options

The 38 sites on the FCC lattice can be connected in many different ways. To make a simple but convincing representation of a 3D Gosper curve, a *Level-2 Gosper-Onion* seems particularly well suited.

The seven sites in the hexagonal faces of shell 2 are perfectly suited to accommodate one typical Gosper-Claw. In a first approach, I place two such claws on adjacent faces and combine them into a “Double-Claw” (Fig.3). I can do this on two pairs of faces (yellow and orange), and this naturally splits the outer shell into two equal halves. The remaining sites in the other four hexagonal faces are used to connect the two outer half-shells to the inner octahedral sites. I found two different ways of doing this. In *Model A* (Fig.3a), the two Double-Claws are cross-connected with two inner struts (green and blue) that each use three vertices of the central octahedron. Each of the two internal connecting branches passes through two opposite vertices of the central octahedron and exhibits a right-angle bend in between. This structure exhibits 4-fold D_2 -symmetry (with 3 mutually perpendicular C_2 rotation axes). Some purists may object to the 90° bend. In a second approach (*Model B*), the two connector branches each use three vertices associated with a single triangular face of the central octahedron. This structure employs only 60° and 120° bends, like the 2D Gosper curve, but it exhibits only 2-fold C_2 -symmetry (Fig.3b).

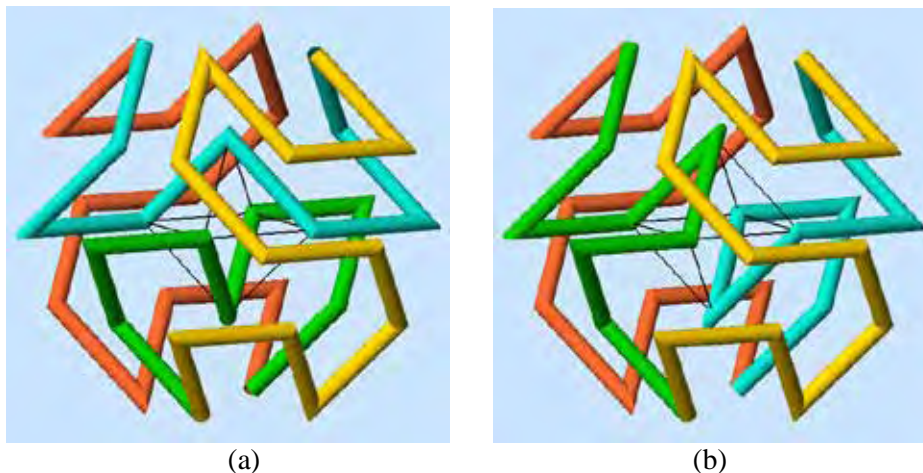


Figure 3: *Gosper-Onion_38* paths: (a) *Model A* with D_2 symmetry, (b) *Model B* with C_2 symmetry.

As an alternative solution, the four hexagon-shaped Gosper-Claws can be placed on four non-adjacent hexagonal faces of the truncated octahedron – instead of forming two Double-Claws on two pairs of adjacent faces. This approach offers additional possibilities, since the four claws can be rotated or reflected individually, whereas in the construction of the Double-Claws, the orientation of the individual claws is prescribed. On the other hand, it is quite a challenging puzzle to find good ways to connect the four individual claws into a single loop, while avoiding, as much as possible, bending angles of 90° or 0° , and while maintaining at least C_2 -symmetry.

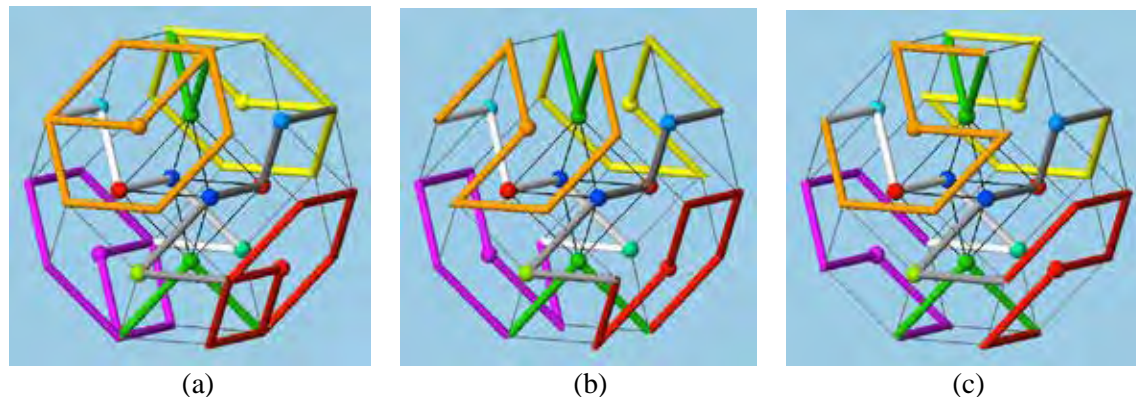


Figure 4: (a) *Claw faces* shown as hexagonal rings; (b) rings replaced with claws to form *Model C*; (c) rings replaced with reversed claws to form *Model D*.

To tackle this design challenge, I present the four hex-faces that were supposed to carry the four individual claws as simple hex-rings (Fig.4a). I can then focus on finding four connector paths that connect the four claws into a single loop. This requires that the two connecting points to each of the four hex-rings are placed 120° apart with respect to the center of the hex-faces. The individual claws can then be rotated to have their two endpoints (which lie 120° apart) line up with the two path connection points.

After some amount of searching through possible solutions, I found a nice configuration that exhibits C_2 -symmetry around the (vertical) z-axis (Fig.4b). This design also keeps the four hex-faces that do not carry an individual Gosper-Claw relatively “flat”, without the pointy spikes produced by 120° -bends that jut out in the centers of these faces, as is the case in both *Model A* and *Model B*. This model still has two of the square faces of the cuboctahedron spanned by four spiky 120° -bends, which may serve as the “legs” of an actual sculpture – to give it a rather “airy” look (Figs.19b,c). However, this model does have six bends of 90° . More recently I found that by reversing the orientation of the four Gosper-Claws I could get rid of four of the 90° bends (Fig.4c). By mixing and matching the orientation of the four Gosper-Claws, there are other interesting possibilities to create different internal connector paths.

Fabrication Issues

Chris Ohler [2], has committed himself to construct such a low-complexity Gosper sculpture based on *Model A*, using metal tubing with a diameter of 133mm. He has worked out the relevant cutting angles for all 38 tubular segments. For each tube segment, one needs to know the mitre angle at which the ends of the tube must be cut, as well as the torsion angle between the two cuts. For a poly-line path that lies in a plane, these torsion angles are 0° as long as the bending continues in the same CW- or CCW-sense, and they are 180° wherever there is an inflection point. For an arbitrary polyline in 3D space, many possible torsion angles may occur. It is interesting to note that for these Gosper curves that connect nearest neighbours on the FCC lattice, only four non-planar torsion angles appear: 54.74° , 70.53° , 109.47° , and 125.26° . Figure 5a shows Ohler’s list of cutting angles α and torsion angles β for one quarter of *Model A* discussed above. The remainder of the model follows the same list – or it’s reversal – because of the symmetry of the sculpture. Figures 5b and 5c show some of the pre-cut metal tubes and the formation of a properly mitred turning angle of 120° .

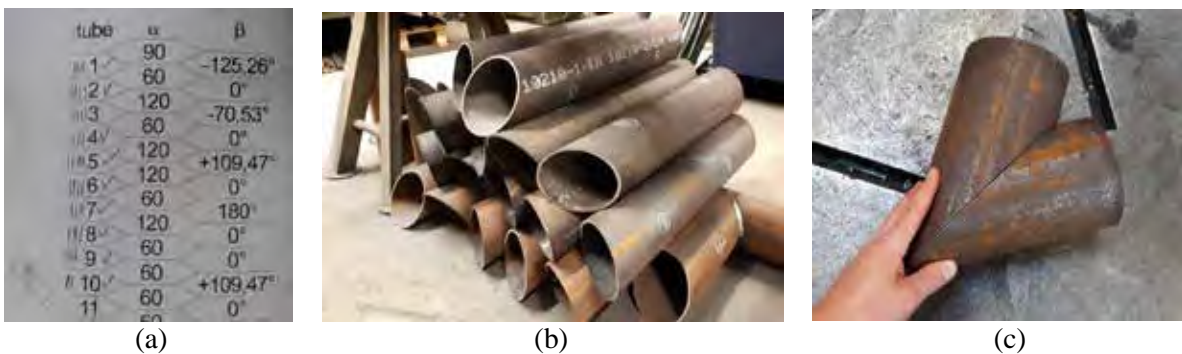


Figure 5: (a) Cutting angles α and torsion angles β for one quarter of all the tube segments. (b) Pre-cut metal tube segments. (c) A mitered 120° tubular junction. (Images by C. Ohler).

Assembly Issues

The next issues to be addressed are: In what sequence should the individual tube segments be assembled? And: What kinds of jigs are required to hold individual tube segments and partial subassemblies in exactly the right positions, so that in the end the Gosper-loop closes nicely on itself? It would not be accurate enough to just match the elliptical cross sections of two consecutive tube segments to obtain a clean realization of the whole sculpture. It is advantageous to have additional help from some suitably designed jigs to hold small pre-assembled chains of a few consecutive tubular segment in their proper positions. In the following I present a proposed assembly scheme, verified on a small 3D-print model.

In each Gosper-Claw there are six consecutive segments that lie in the same common plane. It makes a lot of sense to preassemble those tube-chains, since this is mostly a 2D layout and we need not worry about obtaining the proper torsion angles. It may also be practical to add one or two additional tubular segments at the ends of such a Gosper-Claw. In the following tests and evaluations of the proposed assembly sequences, these parts with five to eight tube segments were 3D-printed as cohesive units. The large, planar regions of the Gosper-Claws then permit the insertion of some judiciously chosen “spacer slabs,” which allow additional parts to be added in suitably offset, parallel neighbor-planes.

An Initial Assembly Step

Figure 6 shows the six tubular chains that form the sculptural sub-assemblies of *Model A*: two internal cross-connector branches (Fig.6a), two complete Gosper-Claws (Fig.6b), and two partial Gosper-Claws that constitute the remainder of a Double-Claw (Fig.6c).

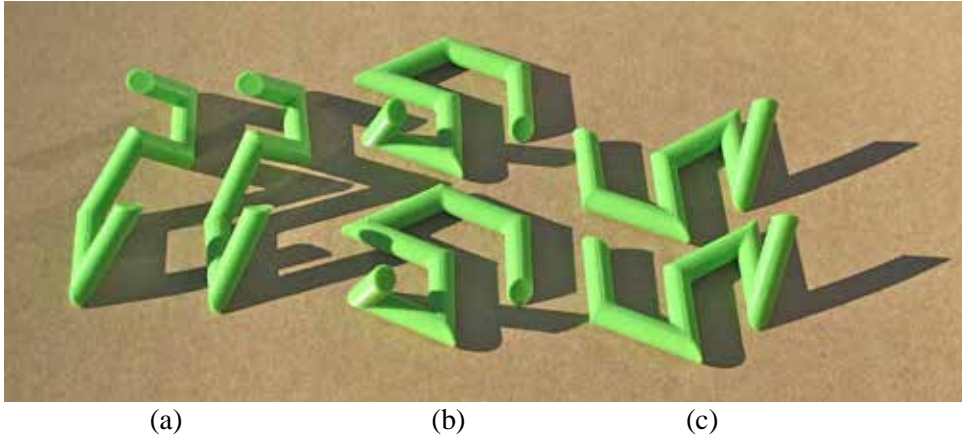


Figure 6: *Sub-assembly tube chains: (a) internal connectors, (b,c) peripheral Gosper claws.*

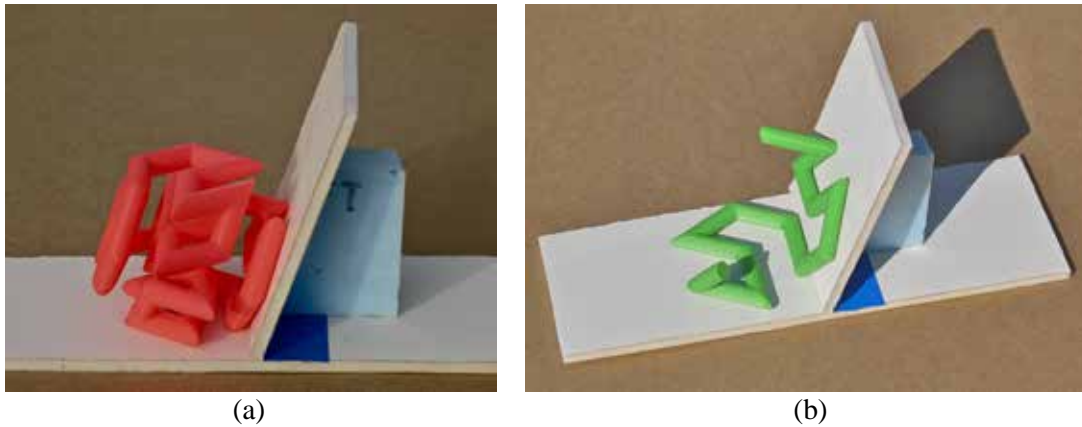


Figure 7: *(a) Jig with proper dihedral angle. (b) Two simple claws fused into a Double-Claw.*

The two Gosper-Claws making up a Double-Claw occupy two adjacent faces of the outer *Onion* shell and form a dihedral angle of 109.47° between them. A jig with a wall properly slanted at this angle (Fig.7a) makes it easy to put two such parts together (Fig.7b) to form one of the two Double-Claws (Fig.8a,b). This is also the first step for assembling *Model B*.

Next, I needed to find a suitable jig for lining up the two internal connectors. The two different sculpture models, *Model A* and *Model B* (Fig.3a,b), require slightly different approaches.

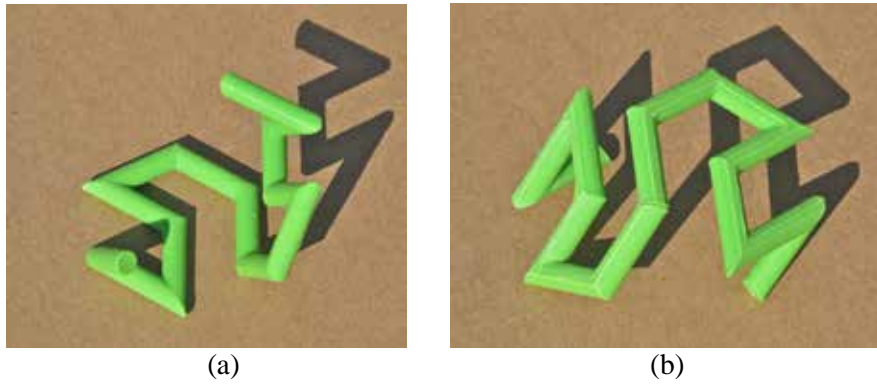


Figure 8: *The resulting peripheral Double-Claws: (a) inside view, (b) outside view.*

Assembling Model A

In *Model A* (Fig.3a) the internal connectors form some “crooked” struts that cross each other in the central octahedron. Figure 9 shows how the two inner struts relate to one another. Pairs of some tubular segments in one strut happen to be collinear with the corresponding segments in the other strut. Furthermore, the 3-segment, planar end-chains in each strut are parallel to the corresponding end-chains in the other strut. Thus, two *spacer-slabs* of proper thickness can be used to keep the two struts in proper position with respect to one another. The thickness of this spacer-slab is calculated as the basic path-segment separation, ss , minus the given tube diameter, td . In 2D, the distance between nearest parallel path segments is: $ul * \cos(30^\circ) = 0.866 ul$, where ul is the unit-length step of the Gosper path. In the 3D sculpture, the critical separation of a whole group of coplanar segments from another group in an adjacent parallel plane is further reduced by: $\sin(109.47^\circ) = 0.9428$, (the dihedral angle of the octahedron). Thus:

$$ss = 0.8165 ul, \text{ and thus the spacer-slab thickness, } st, \text{ should be: } st = 0.8165 ul - td.$$

With the two spacer slabs in place (Fig.9a), the collinearity mentioned above can be checked and fine-adjusted, and the two crossing struts can be bundled into a single rigid assembly (Fig.9b).

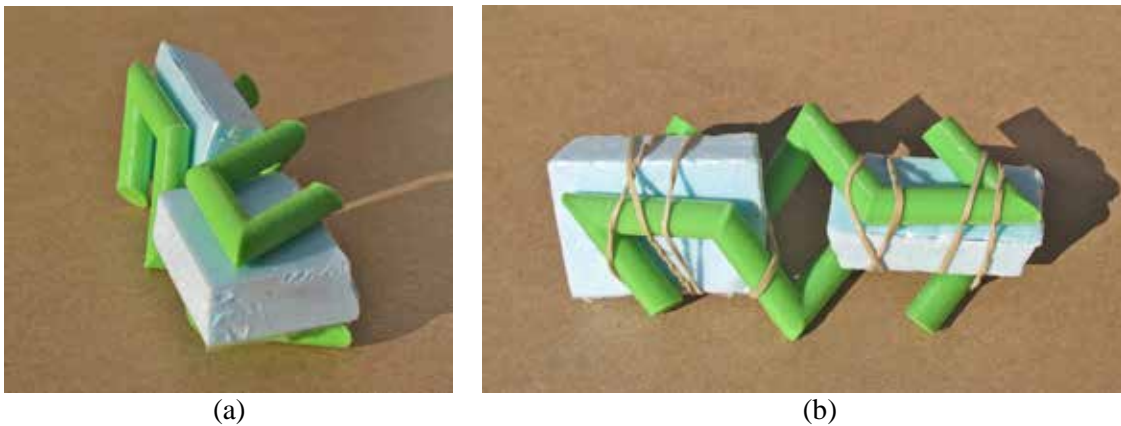


Figure 9: *(a) Alignment of the two inner cross-connectors: (a) end-view, (b) side-view.*

The same thickness spacer-slab is also used to define the proper distancing of that central bundle from the outer claws. Thus, a spacer-slab can be placed inside one of the Double-Claws (Fig.10a) and the central bundle is then placed on top of this slab (Fig.10b). This brings two connection points into close proximity and allows them to be fused with the proper orientation and alignment (see red arrows in Figures 10b and 10c).

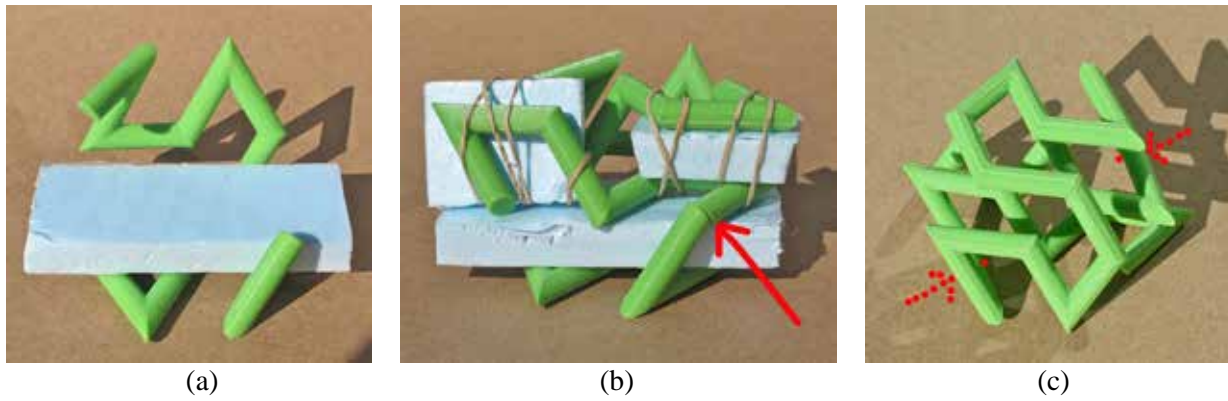


Figure 10: *Inserting cross connectors: (a) spacer-slab in a Double-Claw; (b) connector bundle placed; (c) the result with two connections made (shown upside down).*

Because of the symmetry of this sculpture, the same type of spacer-slab also controls the spacing of the central cross-connector bundle to the second Double-Claw. A spacer-slab is placed onto the central bundle, and the second Double-Claw is then draped across this assembly (Fig.11a). This would allow me to make the final two connections to this upper Double-Claw. However, in reality, it is better to just make the connections between the bottom Double-Claw and the connector bundle, and then remove the spacer slab. This results in Figure 10c. This structure, turned upside down, can then be placed on a second configuration that looks like Figure 10a, and the final two connections can be made (Fig.11b). Eventually the last spacer slab is also removed, revealing the completed *Model A* (Fig.12).

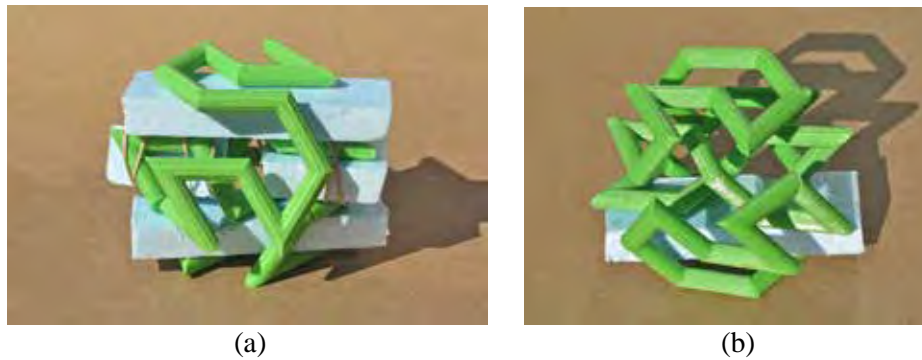


Figure 11: *(a) All four parts properly spaced. (b) Pre-connected part (Fig.10c) on top of spacer slab.*

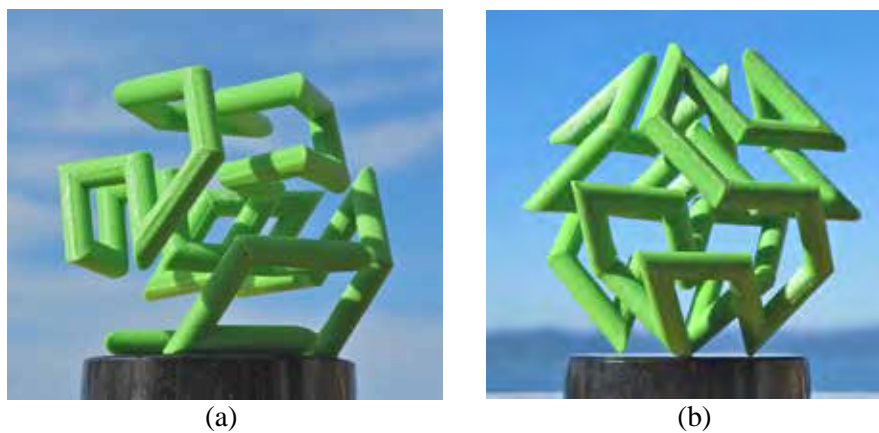


Figure 12: *Resulting sculptural maquette: (a) resting on one Gosper-Claw, (b) standing on four “legs.”*

Assembling Model B

Figure 13 shows the six sub-assemblies for *Model B*. The first step is the same as for *Model A*: The two Double-Claws are formed by re-using the jig shown in Figure 7. In *Model B* the internal connectors are more compact and do not cross each other. Figures 13b,c show that in this model, too, there are open slots that can accommodate appropriate spacer-slabs.

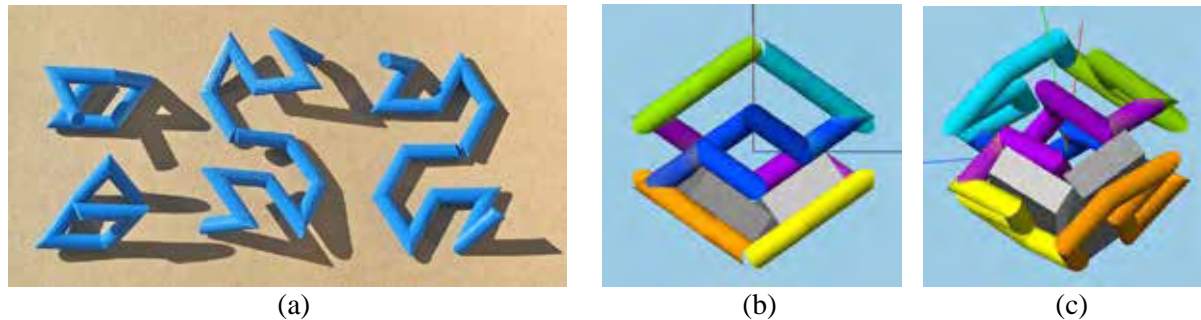


Figure 13: (a) Sub-assemblies for Model B. (b,c) Open slots for spacer-slabs in Model B.

First, the two connector parts are aligned and fused against the bottom Double-Claw. To do this, I form a configuration with one spacer-slab placed inside one Double-Claw, as seen in Figure 10a. On this slab I place one of the connector pieces and fuse it with one end to the underlying Double-Claw (yellow arrow in Fig.14a). Then the spacer slab is removed and re-inserted flush against the other half of the same Double-Claw. In this configuration, I add the second connector unit and fuse it to the other end of the Double-Claw (yellow arrow in Fig.14b). Next, I form a new cradle (like Fig.10a) with the 2nd Double-Claw. Into this cradle, I drop the assembly described above and fuse both ends of the Double-Claw to the two connector units (Fig.15a). Removal of the spacer slab reveals the completed *Model B* (Fig.15b).

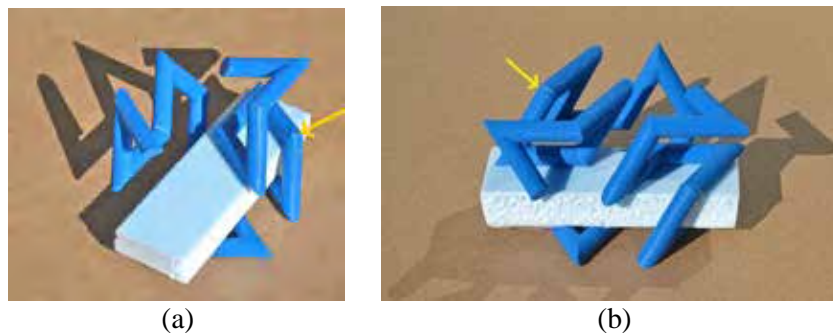


Figure 14: (a) Placing and fusing a first connector; (b) switching spacer slab to place 2nd connector.

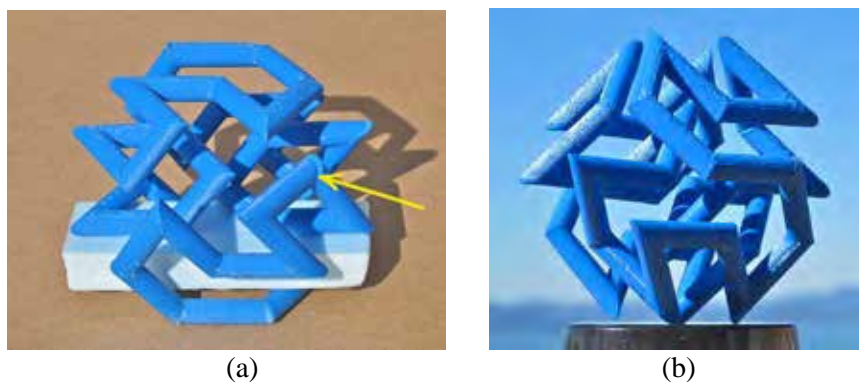


Figure 15: (a) One Double-Claw plus two connectors placed on 2nd Double-Claw plus spacer-slab. (b) Resulting sculptural maquette, Model B, standing on four “legs.”

Assembling Model C

Model C can also be partitioned into parts that start with a planar, 6-segment Gosper-Claw. However, in this model there are no clearly identifiable internal connector parts. Instead, the Gosper-Claws are enhanced in two different ways. In the *simpler part* (Fig.16a), one extra segment is added to each end of the Gosper-Claw. In the more *complex part* (Fig.16b), one of the ends of the Gosper-Claw carries a chain of *four* tubular segments that form a “zig-zag” path that lies in a plane parallel to the Gosper-Claw (Fig.16b). Figure 16 shows the four 3D-printed subunits, composed of 8 and 11 segments, respectively.

For a metal tube construction, the 11-segment part may be further partitioned into another *simple* part with 8 segments plus a separate planar branch of three segments, which can be added in a plane offset from the Gosper-Claw by using one of the spacer slabs described below.

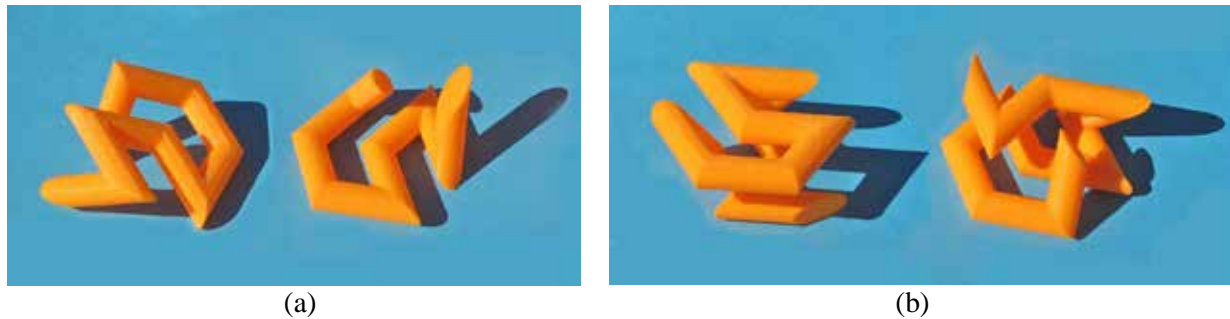


Figure 16: *Model C:* (a) the two simple parts, (b) the two complex parts. – (different views).

To assemble these parts, spacer-slabs again come in handy! Figures 17a,b show that there are spacer-slabs that fit snugly into both the simple and the complex part. Note that the orientation of these slabs is perpendicular to the dominant coordinate axes of the octahedron, and the thickness of these spacer-slabs thus is only: $ss = 0.7071 * ul - td$. Pairs of identical parts with their inserted slabs then fit together nicely, so that the two slabs become parallel to each other. Between them, they squeeze the zig-zag branch that needs to be joined, and they guarantee that this branch remains planar (Fig.17c,d).

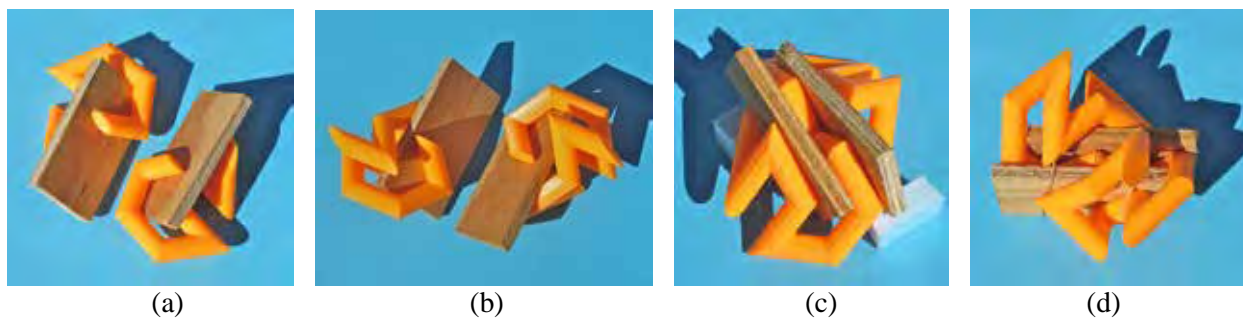


Figure 17: *Spacer-slabs:* (a) in the simple part; (b) in the complex part; (c) joining two simple parts; (d) joining two complex parts.

Figures 18a,b show the resulting joined pairs of identical parts. To assemble these into the final model, two spacer-slabs are used simultaneously (Fig.18c). These slabs are again parallel to the claw-faces, and thus their thickness is again: $st = 0.8165 ul - td$, as discussed above for *Model A*. After removal of the spacer-slabs, the sculpture *Model C* emerges (Fig.19).

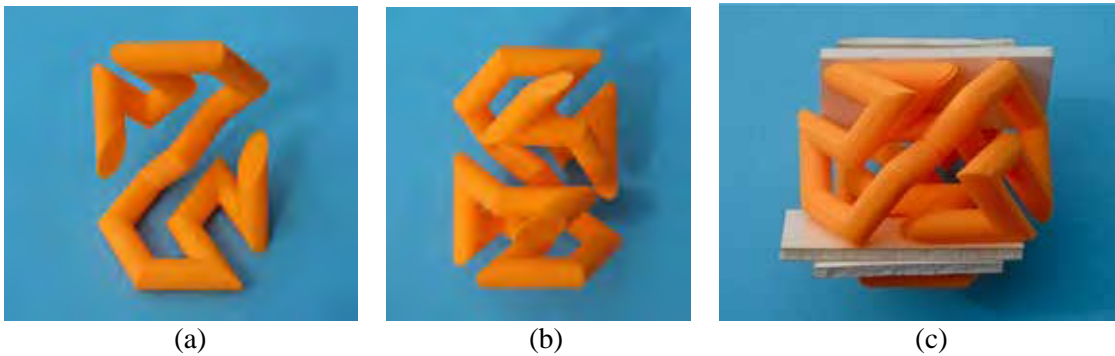


Figure 18: The two glued double-parts: (a) simple; (b) complex; (c) aligned with proper spacer-slabs.

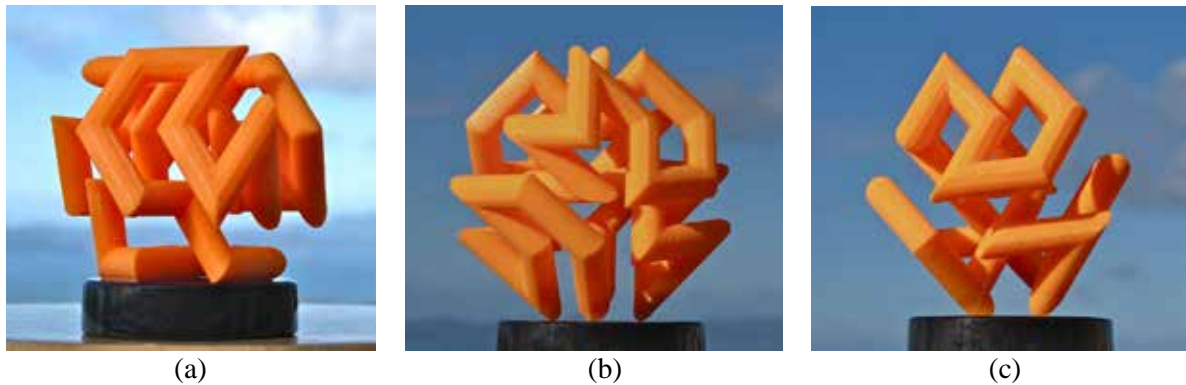


Figure 19: Final sculpture Model C: (a) placed on one Gosper-Claw, (b,c) standing on four “legs.”

Similar sub-assemblies and compositions with the help of corresponding spacer-slabs also apply in the construction of *Model D* – and probably for most other variants of this *Gosper-Onion_38* with four separate Gosper-Claws in different orientations and different rotations.

Scaling Issues

In our SLIDE [6] or JPCAD [1] programs for designing and visualizing these sculptural models, I can readily adjust the tube diameter used in the sweep along the Gosper polyline to find the best level of “transparency” into the inner parts of the sculpture. I like a value that leaves a little more than a tube diameter of empty space between adjacent parallel tube segments, say, a tube diameter of: $td = 0.4 ul$.

When considering a physical sculpture made from metal tube segments, the diameter of the tubes used is typically given, and thus the unit step length, ul , that results in the desired separation between nearest parallel tube segments would then be calculated as: $ul = 2.5 td$. Thus, if the sculpture is constructed from tubes with a diameter of 133mm, the Gosper polyline would be scaled so that the unit-step-length, ul , is about 33cm. The overall *Gosper-Onion_38* sculpture would then have a diameter of about: $3 ul + td$, or a little more than 1 meter.

3D-Printed Gosper-Globes

In my *Gosper-Globes* the Gosper strand is confined to lie in a thin spherical shell. The sites where the strand makes a turn lie on a sphere. But these sites do not form a completely regular array, and the step-length between nearest neighbors will vary somewhat. In the examples discussed in this section, I approximate the sphere with a pentakis-dodecahedron (Fig.20a) [11]. I then cover each of the 60 triangles with an array of subdivision sites ranging from a subdivision level of 3 (Fig.20c) for the smaller globe (Fig.24) to a subdivision level of 5 in the final sculpture (Fig.28).

Stability Issues

To stay true to the 2D Gosper curve, a Gosper sculpture should consist of just one (possibly closed-loop) strand. However, if the Gosper strand has more than a few hundred segments and no intermediate support, it would result in a rather flimsy sculpture, where parts of it would sag under the influence of gravity. In order to make more complex *Gosper-Globes*, I allow the Gosper strand to assume a more complex graph structure. In the following, this graph has the topology of the wire-frame of a cube. Thus, there will be eight 3-way branching points. The “edges” between two nearest branching points then “fractalized,” i.e., deformed into Gosper-like polylines that together will cover the surface of a sphere as uniformly as possible.

To define the best set of “lattice” points on the sphere surface, I want to start with as “round” a polyhedron as possible, – not with a cube. Good candidates are the dodecahedron, or better yet, the pentakis-dodecahedron (Fig.20a). The 60 triangles can then readily yield an array of subdivision points. When these points are projected onto the circum-sphere, they yield a nice pattern of almost uniformly spaced points. It is also easy to integrate the desired wire-frame cube with this dodecahedral geometry. Each pentagonal region carries one of the 12 cube edges, and these edges connect two non-adjacent vertices in each pentagon (Fig.20b). These 12 wire segments are then “fractalized” to cover the whole pentagonal region. In Figure 20c, one “fractalized cube edge” runs from vertex A to vertex C.

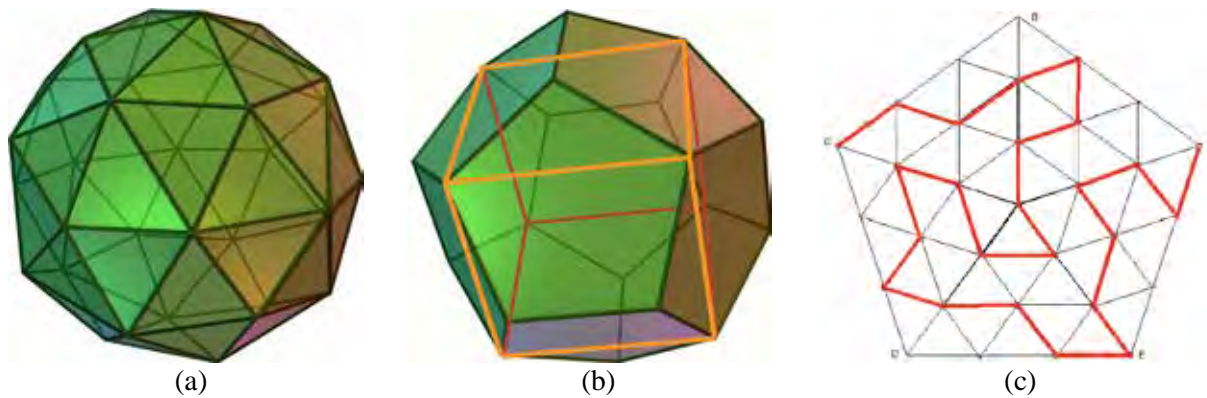


Figure 20: (a) *Pentakis-dodecahedron.* (b) *Cube-frame embedded in a dodecahedron.* (c) *Level-3 fractalized cube edge to cover a pentagonal region.*

Fabrication Issues

Complex spherical balls or shells (Figs. 1c, 22a) formed by an intricate web of tubular pathways, such as 3D polylines following a Hilbert curve [10] or mimicking a Gosper curve [9][12], are difficult to print on most 3D printers that do not provide an all-around, support structure as is the case in SLS (Selective Laser Sintering) or in Z-Corp printers [13].

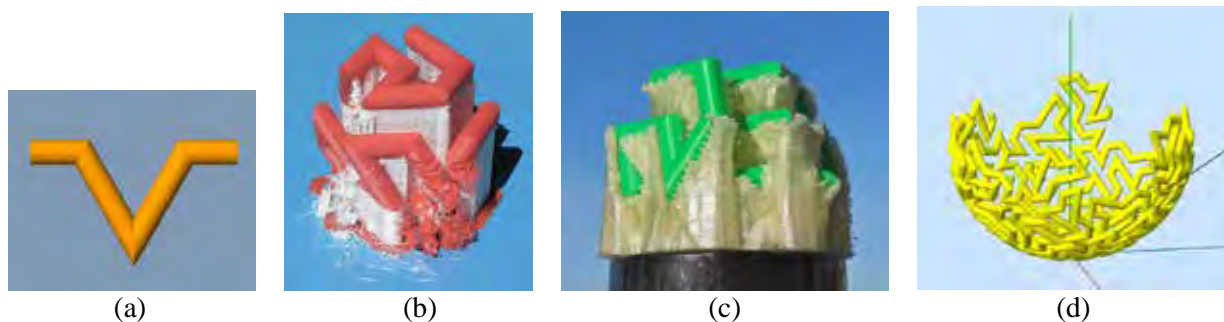


Figure 21: (a) *Down-spike in the Gosper curve,* (b) *collapsed because of insufficient support structure,* (c) *fabricated with a copious amount of support material.* (d) *Half of a spherical Gosper-Globe.*

One reason is that it is difficult to remove the support material from the inside regions of the sphere. A second difficulty, more specific to the Gosper sculptures that I am currently interested in, is related to any downward-pointing spikes (Fig.21a). These spiky points may not get an adequate support structure (Fig.21b) that would allow them to be anchored in a sturdy enough way, so that they can then support any lengthy, complex tubular pathways that connect to this down-spike [4]. This problem can be mitigated by using a very generous amount of scaffolding material to support every model facet with a downward-pointing face normal (Fig.21c). However, this creates an enormous amount of scaffolding, which may fuse into a rather solid block that is near-impossible to remove from the inner parts of the whole structure – unless the support material is soluble.

One idea to work around this problem is to split the spherical shell into two hemispheres and print each in the orientation of a hemispherical bowl (Fig.21d). This places the support material on the outside of the bowl and instantiates it radially only as far until the slope of the bowl becomes steep enough so that no further support is required. However, for a hemispherical shell, the amount of support material is still relatively voluminous, and down-spikes in the upper, steeper parts of the bowl surface are still a problem.

The situation improves as we split the sphere surface into more, and correspondingly “flatter” parts. A natural next step is to split the sphere into *four* quadrants – or rather “*tetrants*” – since I plan to use a partitioning based on tetrahedral symmetry (Fig.22b). Now the total volume of support material is much smaller, and there are no longer any truly “down-pointing” spikes. Such a partitioning of the sphere also allows me to construct maquettes that are somewhat larger than the build volume of the available printer.

Gosper-Globe_276

As a first demonstration, I am constructing a spherical shell, which I call *Gosper-Globe_276* (Fig.22a), on an Ultimaker printer [7] with PLA build-material and also with PLA support-material. In order to make this approach work, I want all tubular elements appearing in each sphere-tetrant to be connected to each other, forming a reasonably rigid part that can be used as a building block for the whole sphere shell. Thus, rather than designing a single-loop Gosper path that winds its way serially through all four tetrant domains, I have introduced some branching points. The resulting tubular structure in the spherical shell now has the same connectivity as the wire-frame of a cube: Twelve zig-zag-y paths connect eight 3-way branch points. One of these valence-3 branching points can be found in the center of each of the four tetrant domains; the other four junctions appear at three symmetrically spread-out locations on the rims of the tetrant domains.

It was more difficult than anticipated to find a reasonable Gosper path to replace all twelve cube edges, so that three of them lie entirely in each tetrant part. To facilitate the assembly of the four tetrant parts, I combined the geometries of the three partial 3-way branch points lying on the periphery of the tetrant part into one fully modeled 3-way junction. I then placed just one of those onto the border of each tetrant part, – shown in contrasting colors in Figure 22b. Straight, cylindrical tube connections then accommodate the two incoming paths from the other two tetrant parts that connect to this branch point.

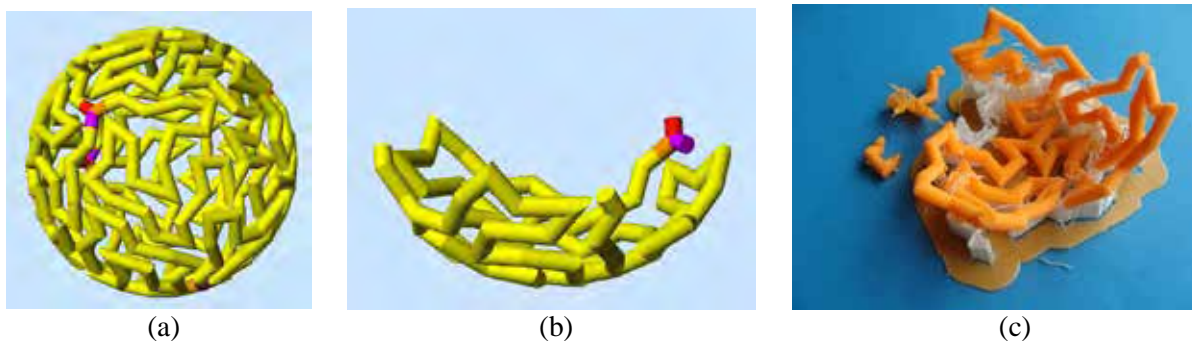


Figure 22: (a) CAD model of *Gosper-Globe_276*; (b) one “tetrant” of it. (c) A first, unsuccessful print.

In my first attempt, I used the default settings for the Ultimaker printer as supplied by the Cloud 3DPrinterOS service [14] offered through the Jacobs Institute for Design Innovation at Berkeley. The resulting print had an incomplete support structure that caused some breaks in the 3D tubular pathways (Fig.22c). In my second attempt, I used the *expert* mode in the 3DPrinterOS. I lowered the support overhang angle from 60° to 40° , and I increased the support density from 15% to 20%. This resulted in a good support structure and nice, sturdy parts. Figure 23a shows one tetrant part as it came off the Ultimaker printer. The reduced overhang angle was sufficient to firmly support the many tubular segments that then serve as anchors for additional tubular branches, which may take off at steeper angles and which need no further support. The support was easy to break away, and the “spidery” parts (Figs.23b,c) were rigid enough to keep their shapes when handled for clean-up and for assembling the complete spherical shell.

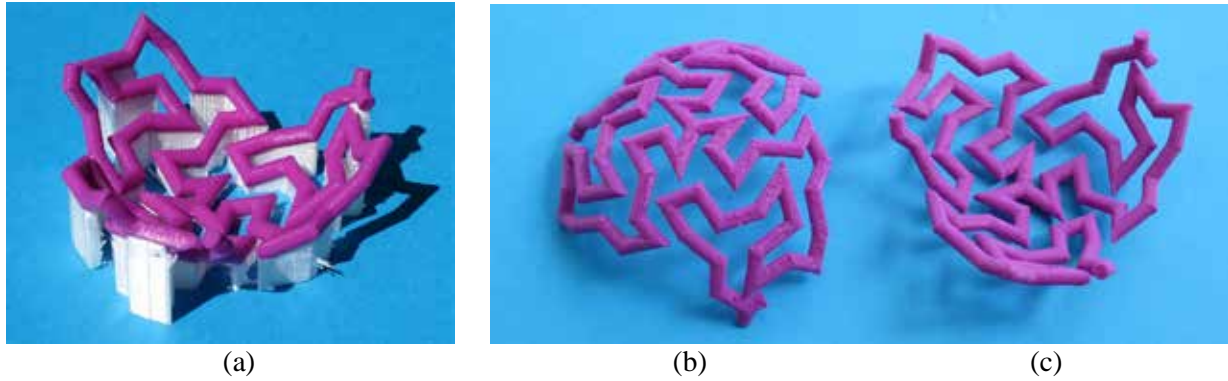


Figure 23: (a) Tetrant part as it came off the Ultimaker printer. (b, c) Outside and inside view of the tetrant part after its (white) support structure had been removed.

The first two tetrants connect to each other with just two tubular joins. This leaves the angle of rotation around the line through these two connection points rather poorly defined. To make sure that the two parts join at the right angle so that they nicely adhere to the sphere of the final sculpture, I formed a bowl-shaped supporting jig that kept the parts tilted at the right angles. Joining a third tetrant part is less ambiguous, since it joins the previous two parts at *three* tubular join-points (Fig.24a). Adding the fourth tetrant by fusing another three joints then completes *Gosper-Globe_276* (Fig.24b).

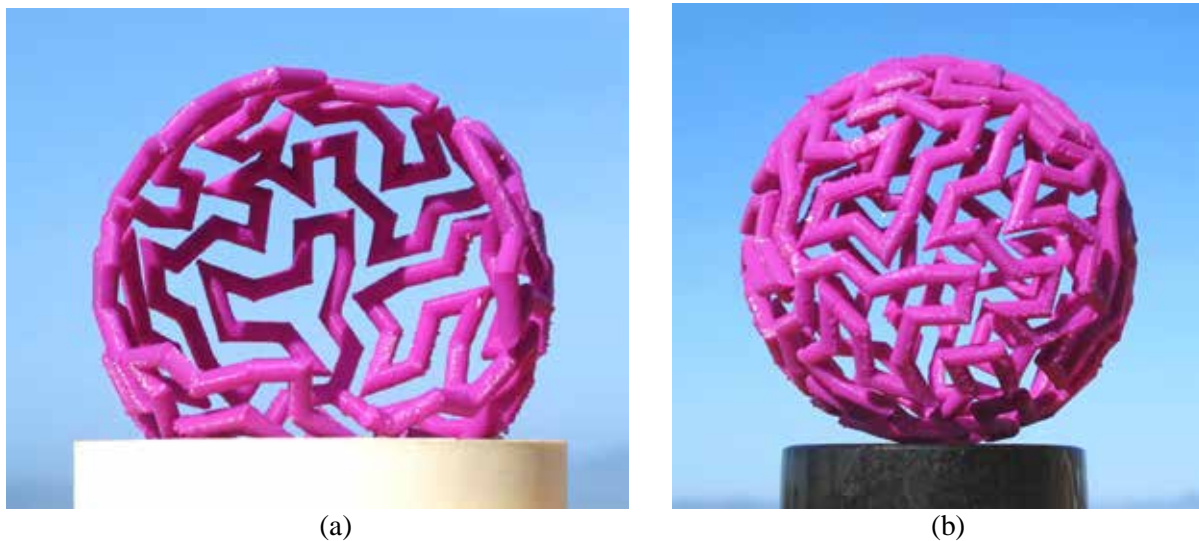


Figure 24: (a) Three tetrant parts assembled into a partial spherical shell. (b) The complete spherical shell of *Gosper-Globe_276*.

Gosper-Globe_756

The construction of the purple *Gosper-Globe_276* gave me the confidence that this approach is indeed practical. So, I embarked on building a globe of much higher complexity and of a size that exceeds the build volume of the Ultimaker printers to which I had access through the Cloud 3DprintOS.

The 3D-printed parts from which the *Gosper-Globe_276* was assembled have three zig-zag paths running from a central 3-way branch point to the three branch points on the periphery of the tetrant in 23 steps on a set of vertices that correspond to a level-3 subdivision of the 60 triangles of the pentakis-dodecahedron [11]. In the new *Gosper-Globe_756*, I used a level-5 subdivision of these triangles, which results in the three zig-zag paths in each tetrant having 63 steps. Figure 25 shows one of those parts as it comes off the printer, and later, after about half the support material has been removed (Figs.25b,c).

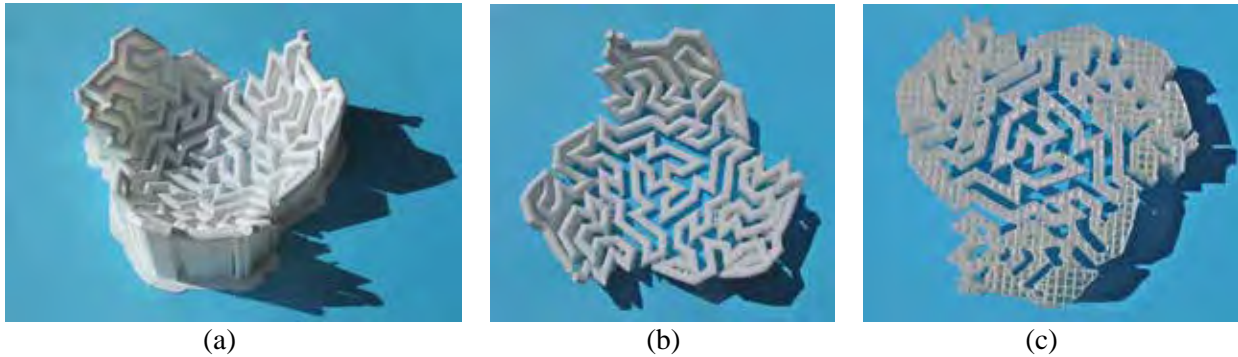


Figure 25: *One tetrant of the Gosper-Globe_756: (a) as it comes off the Ultimaker 3D printer, (b) after some support removal, (c) seen from the support side.*

At first, I was worried that each tetrant part would be rather “flimsy” and would deform seriously under gravity once it was freed from its support structure. Therefore, I kept much of the support in place until I had properly joined two such tetrants, and I planned to remove the remaining support structure after having joined all four parts.

The first two tetrant parts connect in only two of the 3-way branch points. To make sure that they get joined at the right angle, I constructed a jig that provides two mounting planes with the dihedral angle of the tetrahedron (70.5 degrees) between them (Fig.26a). Figure 26b shows the resulting assembly of the first two tetrants joined in this jig.



Figure 26: *(a) Jig to hold two tetrants at the proper angle between them. (b) The resulting fused tetrant pair.*

At this point I started the tricky task of removing some support material from this joined pair without stressing or breaking the actual Gosper path. It turned out that handling this assembly was awkward, and soon one of the glued joints broke. This encouraged me to remove all the support from one of the tetrants to see how “flimsy” the unsupported tetrant part really was. Fortunately, it turned out that it was stable enough, so that I no longer had to worry whether I could properly join unsupported tetrant parts (Fig.27a).

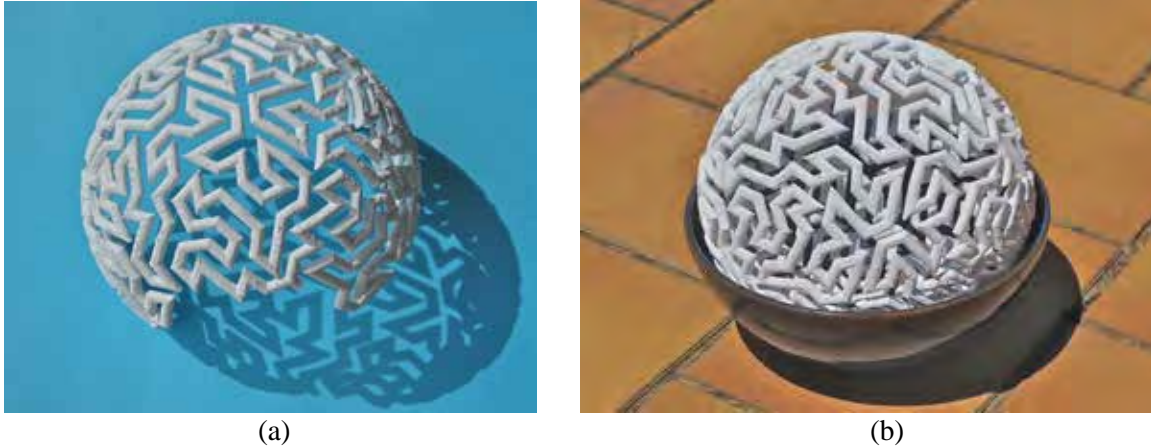


Figure 27: (a) Two tetrant parts fused together. (b) Combining the two pairs in a suitable bowl.

The tetrant parts #3 and #4 were individually freed of all support material and then joined into a pair-constellation like Figure 27a, using again the jig shown in Figure 26a. Joining the two pairs into the complete sphere turned out to be more difficult. The two pairs were indeed rather “flimsy” or “squishy” components! I joined them by placing both parts with the appropriate rough alignment between them into a small salad bowl of the right size (Fig.27b) and then lining up and gluing together a first joint between the two assemblies (using Duco Cement). After that joint had hardened sufficiently, I then lined up and glued a second joint. I repeated this process for joints #3 and #4. With all joints properly fused, the completed globe was firm enough to hold its shape under the influence of gravity, even when just placed on a flat surface or onto a small supporting ring (Fig.28a). A nicer way to show off the assembled *Gosper-Globe_756* is to support it with a small transparent cylinder or a small glass bowl (Fig.28b).

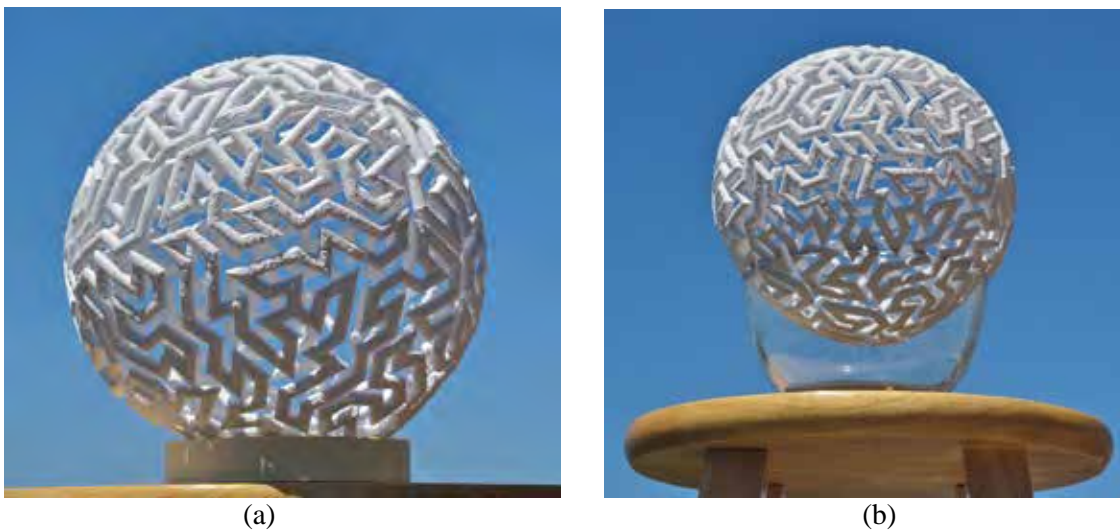


Figure 28: The final *Gosper-Globe_756*: (a) Side view, (b) View from below.

Conclusions

There are many different kinds of Gosper-Sculptures, and there is no standard procedure to design and fabricate such sculptures. Every type of geometry has different design challenges, and the size and type of realization of such a sculpture presents additional difficulties. But in all cases where the complete model cannot be built as a single coherent object on a suitable 3D-printer, judiciously partitioning the overall sculpture into a few sub-assemblies is an important step. Smaller sub-assemblies may be easier to build, but then the composition of these smaller parts into the final sculpture becomes a bigger problem. Complicated scaffolding may be required for the assembly of many small parts into an accurately aligned and balanced sculpture.

For the simple *Gosper-Onion_38* metal sculpture, the described, promising approach was driven by looking for sub-assemblies that have large co-planar regions, for which it was easy to accurately implement the proper torsion angles; these sub-assemblies can then be assembled with the help of a few simple jigs. For the complex *Gosper-Globes*, I was aiming to have as few partitions as possible and to make the largest parts that could be 3D-printed without a lot of difficult-to-remove support material. Different Gosper-Sculpture types and different construction materials will require a re-evaluation of these tradeoffs.

Acknowledgements

I would like to thank the reviewers for their constructive comments as well as good suggestions for future work. One such suggestion was to break *Gosper-Onion_38* into 38 individual parts with individually keyed junctions between the tubular segments to turn this into a 3D snap-together puzzle.

References

- [1] T Chen, "JPCAD." Tech-Report (EECS-2021-125) May 14, 2021. – <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-125.html>
- [2] C. Ohler, "posted artwork." – <https://www.instagram.com/ohleris/?hl=en>
- [3] C. H. Séquin, "Hilbert Cube 512." – https://people.eecs.berkeley.edu/~sequin/PAPERS/SIGGR06_HilbertCube.pdf
- [4] C. H. Séquin, "3D Gosper Sculptures." SPM - SMI_FASE, Hyperseeing, Oct. 2021. – <http://people.tamu.edu/~ergun/hyperseeing/2021/FASE/sequin2021.pdf>
- [5] C. H. Séquin, "Gosper Sculptures Revisited." Bridges, accepted for publication March 2022. – https://people.eecs.berkeley.edu/~sequin/PAPERS/2022_Bridges_Gosper.pfd
- [6] J. Smith, "SLIDE design environment." (2003). – <https://people.eecs.berkeley.edu/~ug/slide/>
- [7] Ultimaker, "Reliable 3D printers." – <https://ultimaker.com/3d-printers>
- [8] J. Ventrella, "Is there a 3D Gosper curve?" – <https://spacefillingcurves.wordpress.com/2016/09/27/is-there-a-3d-gosper-curve/>
- [9] Wikipedia, "Gosper curve." – https://en.wikipedia.org/wiki/Gosper_curve
- [10] Wikipedia, "Hilbert curve." – https://en.wikipedia.org/wiki/Hilbert_curve
- [11] Wikipedia, "Pentakis dodecahedron." – https://en.wikipedia.org/wiki/Pentakis_dodecahedron
- [12] Wolfram MatWorld, "Peano-Gosper Curve." – <https://mathworld.wolfram.com/Peano-GosperCurve.html>
- [13] Z-Corp 3D-printers, "Zcorp 3D Printer Review." – <https://www.3dbeginners.com/zcorp-3d-printer-review/>
- [14] 3DPrinterOS, "Premium 3D Printer Management." – <https://www.3dprinteros.com/>

Conformal Parametric Microstructure Synthesis for Boundary Representations

Q Youn Hong^{1*} and Youngjin Park² and Gershon Elber¹

¹Computer Science Dept., Technion, Israel Institute of Technology, Haifa, Israel

²Dept. of Computer Engineering, Dong-A University, Busan, Korea

Abstract

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

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

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

1 Introduction

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

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

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

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

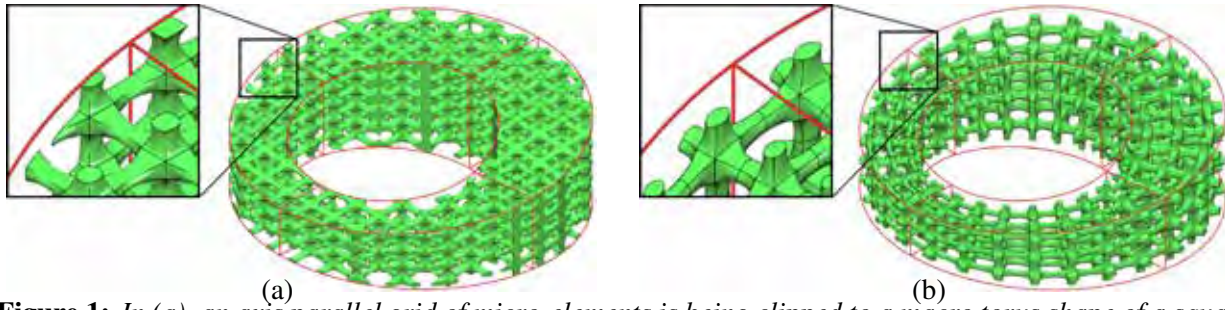


Figure 1: In (a), an axis parallel grid of micro-elements is being clipped to a macro torus shape of a square cross section of varying size. In (b), the micro-elements follow the 'flow' of the same macro torus shape, while no tile is being clipped.

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

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

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

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

2 Previous work

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

2.1 Microstructure Construction

Populating the interior of a volumetric object with microstructures (or a set of micro-elements) has become a common practice in the modeling of 3D geometric object, while exploiting additive manufacturing (AM). One

can control, with flexibility, the geometric properties of the structure while allowing the use of heterogeneous, possibly graded, materials. Microstructures have been adopted in the design of artificial bones, medical applications, or industrial designs such as airplane wings, heat exchanger, and solid rocket fuel [2]. Further details regarding the methodologies and challenges in microstructure-based modeling have compiled in several survey papers, including [23, 18, 15].

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

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

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

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

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

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

Some research have proposed adaptive methods to construct a microstructure. They adaptively manipulate either the macro shape or a coarse microstructure to yield a microstructure that is conformal to the boundary of the macro shape, as much as possible. For instance, Leblanc et al. [14] model a volumetric object using blocks, which are subdivided hierarchically to fit to the given macro shape. Adaptive Voids [17] fill the interior of the mesh adaptively, with tetrahedra, by tessellating the dual mesh of the boundary mesh

and placing the tetrahedra hierarchically. The construction of the microstructure is started from the boundary shape, and some margins around the boundary mesh are filled with solid materials, while the interior of the shape is modeled with an adaptive size of porous cellular structures. Sitharam et al. [22] fill the boundary mesh with the microstructure of corner-sharing tetrahedra (CoSTs). Starting from the coarse CoSTs, the structure is refined with smaller tetrahedra when further details of the boundary shape must be maintained. Kambampati et al. [12] adopt function-based representation (F-rep) to represent the microstructure in the macro object, which is adaptively voxelized with different level of sparsity. A level set method is used to identify active voxels that are part of the shape. The F-rep based microstructures are filled in the macro shape and combined with the macro level surface, just as Pasko et al. [20] do.

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

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

2.2 Cage-based Deformation

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

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

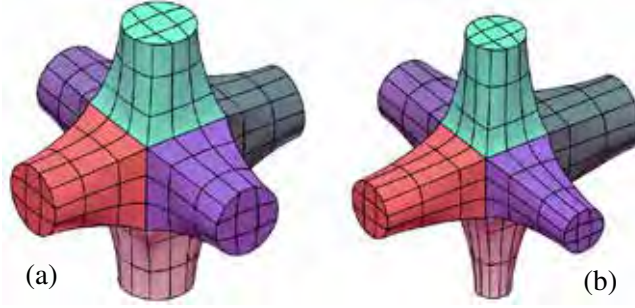


Figure 2: *The parametric tile used in this work, consisting of seven trivariates, all differently colored. In (a), the tile is uniform in all direction. However, the tile, being parametric, can, for instance, have varying arm thicknesses, as can be seen in (b).*

3 The algorithm

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

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

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

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

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

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

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

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

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

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

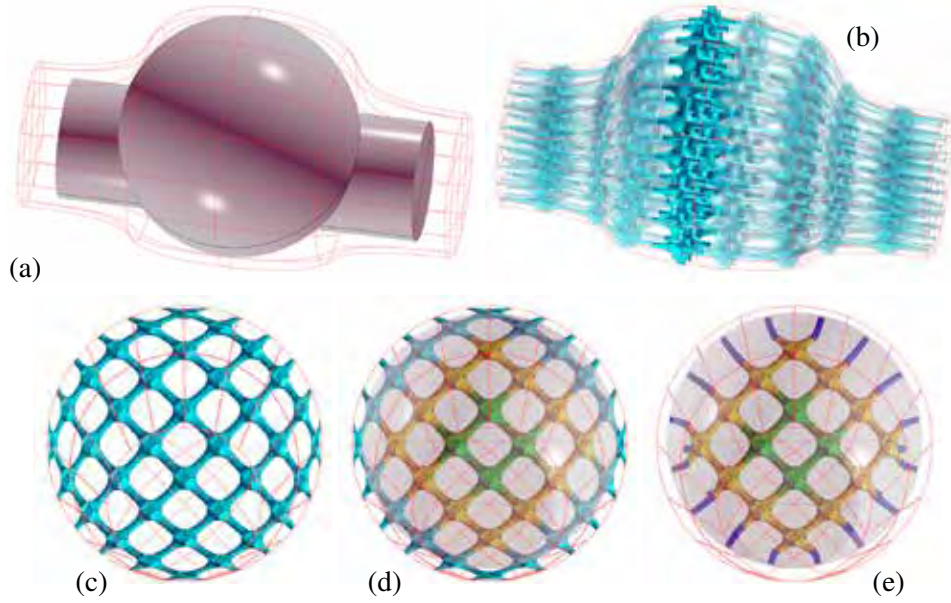


Figure 3: An example of microstructures tiled in a B-rep trimmed surfaces model. In (a), A B-rep model \mathcal{M} is drawn in gray, with a cage trivariate \mathcal{T} drawn in a red wireframe. In (b), microstructure tiles in $\mathcal{MS}_{\mathcal{T}}$ are drawn in cyan. In (c), we show one layer of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

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

Input:

\mathcal{M} : a B-rep model;

$\mathcal{T}(u, v, w)$: a trivariate volumetric cage, containing \mathcal{M} ;

\mathbf{t} : a parametric tile to populate \mathcal{T} (and \mathcal{M}) with;

k, m, n : controls over tiling density of \mathbf{t} in \mathcal{T} , in uvw ;

α : a shape control over the created bridging tiles;

Output:

$\mathcal{MS}_{\mathcal{M}}$: a microstructure tiling of \mathcal{M} , using tile \mathbf{t} , and following the 'flow' of \mathcal{T} ;

Algorithm:

- 1: $\mathcal{MS}_{\mathcal{T}} :=$ a microstructure tiling of \mathcal{T} , using tile \mathbf{t} and controls n, m, k , following [8];
 - 2: $\mathcal{MS}_{\mathcal{F}} := \mathbf{FilterTilesToM}(\mathcal{M}, \mathcal{MS}_{\mathcal{T}})$; // Alg. 2.
 - 3: $\mathcal{MS}_{\mathcal{B}} := \mathbf{BridgeTilesToM}(\mathcal{M}, \mathcal{MS}_{\mathcal{T}}, \mathcal{MS}_{\mathcal{F}}, \alpha)$; // Alg. 3.
 - 4: $\mathcal{MS}_{\mathcal{M}} := \mathcal{MS}_{\mathcal{F}} \cup \mathcal{MS}_{\mathcal{B}}$;
 - 5: **return** $\mathcal{MS}_{\mathcal{M}}$;
-

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

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

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

Input:

\mathcal{M} : a B-rep model;

$\mathcal{MS}_{\mathcal{T}}$: a microstructure formed using a trivariate that fully contains \mathcal{M} ;

Output:

$\mathcal{MS}_{\mathcal{F}}$: a filtered microstructure of $\mathcal{MS}_{\mathcal{T}}$, with tiles completely inside \mathcal{M} ;

Algorithm:

```

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

```

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

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

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

Input:

\mathcal{M} : a B-rep model;

$\mathcal{MS}_{\mathcal{T}}$: a microstructure of volume \mathcal{T} fully containing \mathcal{M} ;

$\mathcal{MS}_{\mathcal{F}}$: a filtered microstructure of $\mathcal{MS}_{\mathcal{T}}$, completely inside \mathcal{M} ;

α : a shape control over the created bridging tiles;

Output:

$\mathcal{MS}_{\mathcal{B}}$: a set of bridging tiles from tiles in $\mathcal{MS}_{\mathcal{F}}$, to $\partial\mathcal{M}$;

Algorithm:

```

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

```

Finally, Algorithm 4 presents one approach to creating the bridging tiles. Bridging tiles join the outermost faces of the filtered tiles in $\mathcal{MS}_{\mathcal{F}}$ to the boundary of \mathcal{M} and enable to maintain the conformality

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

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

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

Input:

\mathcal{M} : a B-rep model;

\mathbf{t}_i : a tile fully inside \mathcal{M} ;

\mathbf{t}_{nbr} : a tile adjacent to \mathbf{t}_i , and intersecting/out-of \mathcal{M} ;

α : a shape control over the created bridging tile;

Output:

\mathbf{b} : a bridging tile from \mathbf{t}_i , from its adjacent face with tile \mathbf{t}_{nbr} , to $\partial\mathcal{M}$;

Algorithm:

- 1: $F :=$ shared face of \mathbf{t}_i and \mathbf{t}_{nbr} ;
 - 2: $p_F :=$ center location of F ;
 - 3: $\vec{n}_F :=$ outgoing unit normal of F at p_F ;
 - 4: $p_{\mathcal{M}} :=$ closest point on $\partial\mathcal{M}$ to p_F , calculated via B-rep **PointProjection**;
 - 5: $d := dist(p_F, p_{\mathcal{M}})$;
 - 6: $q := p_F + \alpha d \vec{n}_F$;
 - 7: $r :=$ closest point on $\partial\mathcal{M}$ to q , calculated via B-rep **PointProjection**;
 - 8: $C :=$ quadratic Bézier curve, using p_F , q , and r as its control points.
 - 9: $\mathbf{b} :=$ sweep volume of surface F along curve C ;
 - 10: **return** \mathbf{b} ;
-

4 Results

Unless otherwise stated, tiles used in this section are trivariates, typically tricubic. Clearly, tiles consisting of surfaces will be faster to process and more so for curves, etc. We will demonstrate that, in this section as well. We start, in Figure 4, with a simple trimmed surfaces based B-rep model that is a union of a sphere and a cylinder. The original cage trivariates and the original set of micro-elements $\mathcal{MS}_{\mathcal{T}}$, are presented in Figure 4 (i), while the final set of conforming micro-element tiles is presented in (ii) to (iv). The filtered (interior to \mathcal{M}) tiles, $\mathcal{MS}_{\mathcal{F}}$, are drawn in green and yellow, with the yellow tiles are those from which bridging

tiles to $\partial\mathcal{M}$ are formed. Finally, the bridging tiles themselves, $\mathcal{MS}_{\mathcal{B}}$, are painted in blue. A low-resolution (recall k, m, n in Algorithm 1) tiling is presented in (a) whereas (b) presents a higher resolution version of (a) in all three axis, using the same cage trivariate. Finally, in (c), a high resolution tiling is shown, with a different, radial, cage trivariate function, \mathcal{T} .

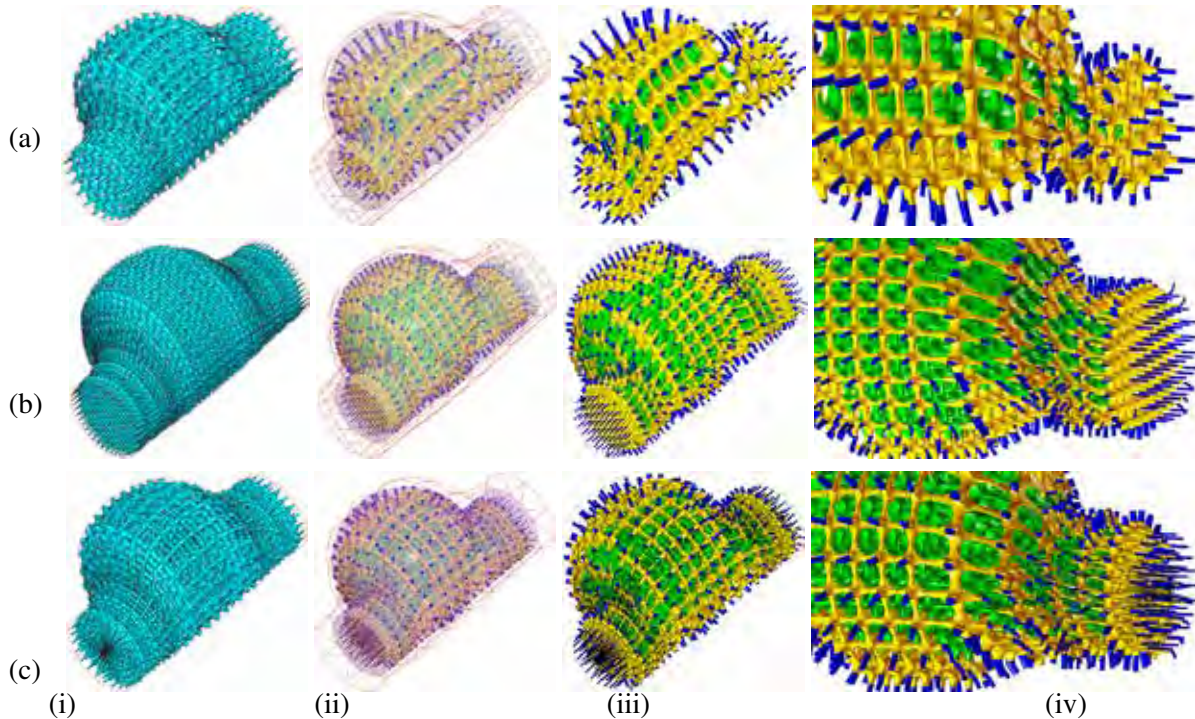


Figure 4: Three examples of microstructures, tiled in a B-rep trimmed surfaces model. (a) shows a fairly low resolution microstructure whereas (b) presents twice higher the resolution compared to (a), in all three (u, v, w) axes. In (c), a high resolution version, similar to (b), is presented but with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

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

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

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

²see <http://graphics.stanford.edu/data/3Dscanrep>

³downloaded from <https://www.thingiverse.com/thing:3731/files>

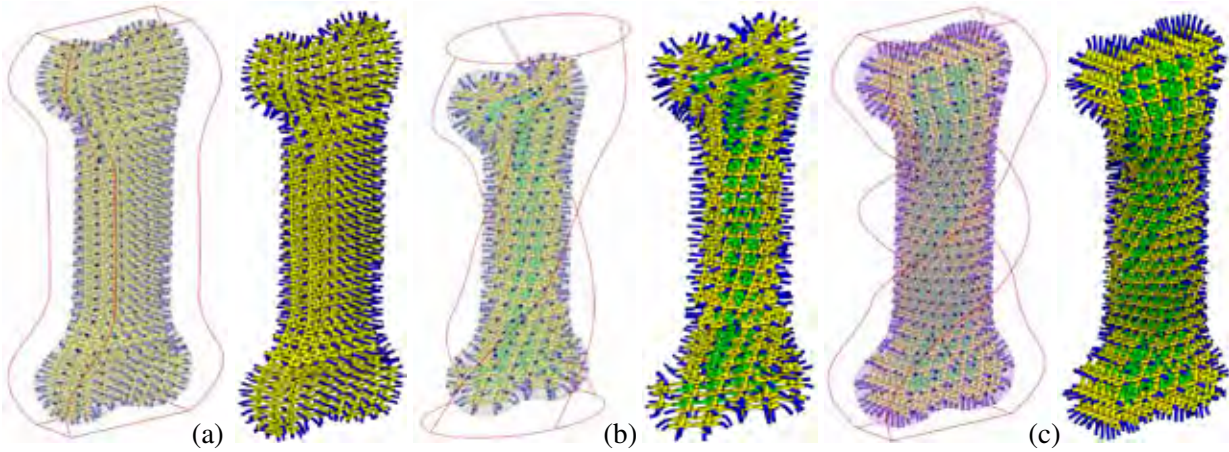


Figure 5: A polygonal B-rep model M of a bone (4252 polygons) tiled with microstructures. Three different cages are shown, from (a) to (c), each of which with two images - the cage in red wireframe and the bone model transparent on the left and the final microstructure MS_M on the right. The filtered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

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

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

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

5 Extensions

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

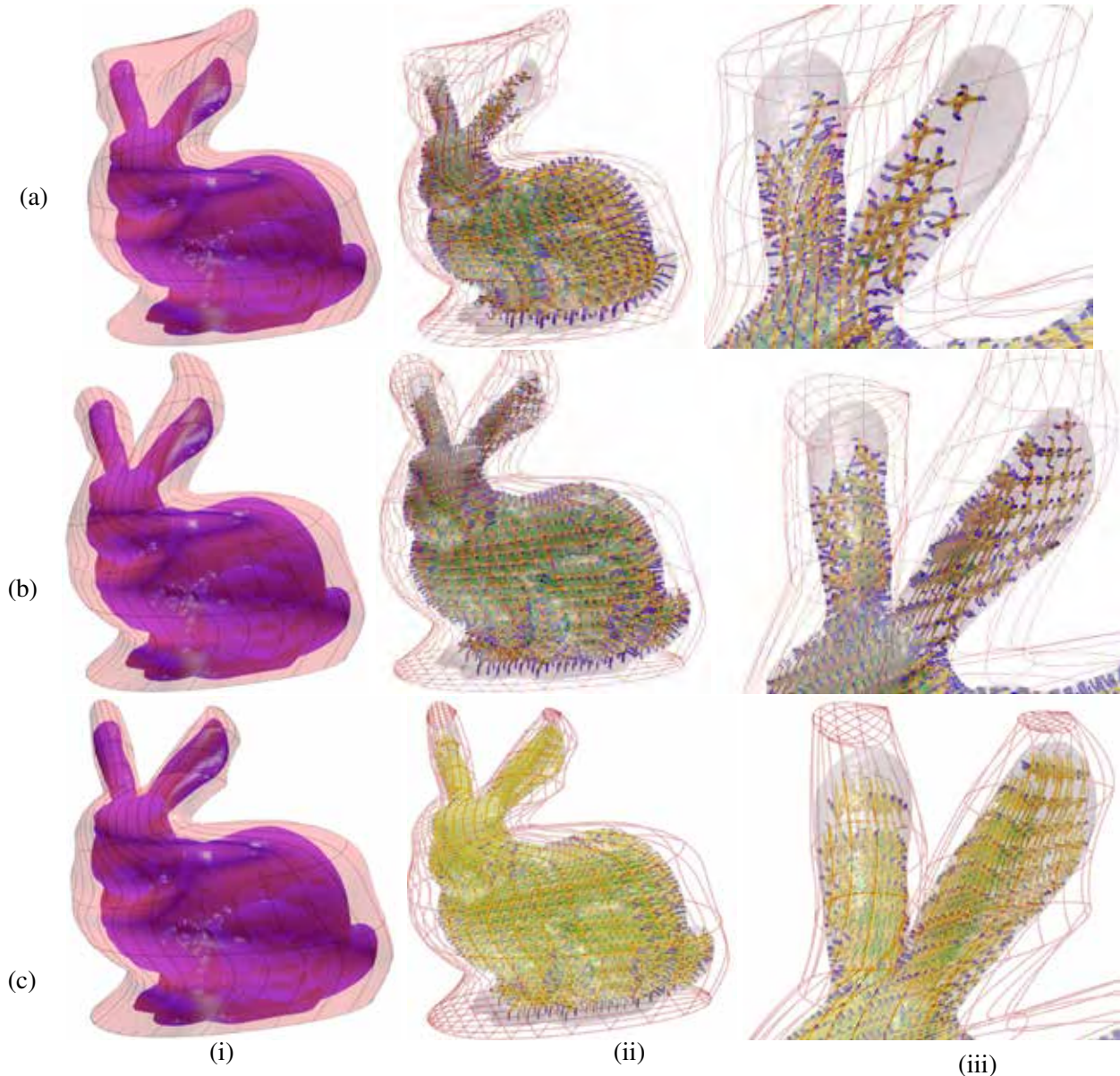


Figure 6: Three examples of microstructures tiled in the polygonal B-rep model of the Stanford Bunny (66848 polygons). (a) to (c) show three different trivariate cages, with (c) being the tightest. The cages are depicted transparent in (i) and using wire-frames in (ii) and (iii). Note the trivariate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

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

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

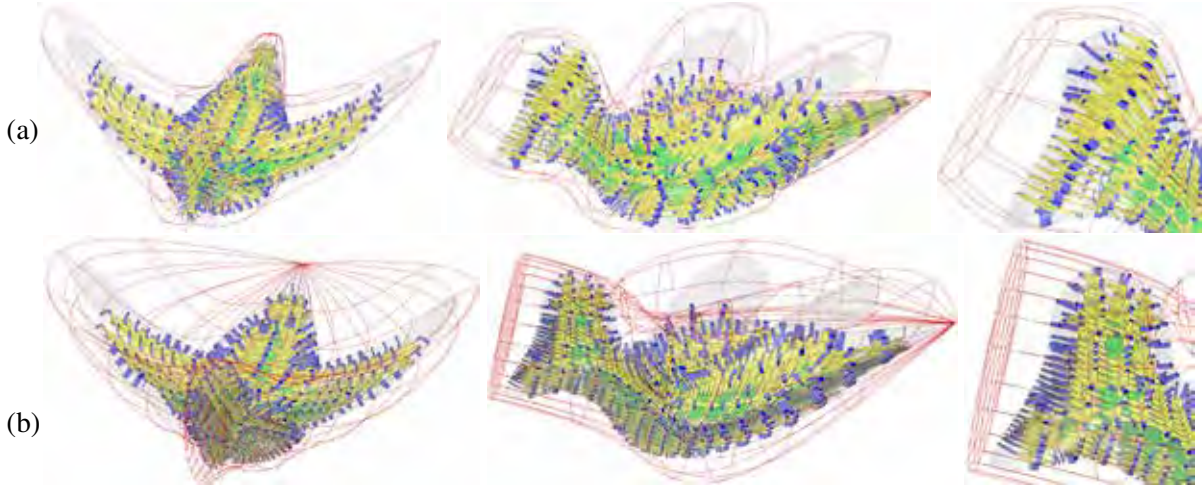


Figure 7: A trimmed surfaces based B-rep model of a duck tiled with microstructures. Two different cages are shown, in (a) and (b), each of which with three images. The cage trivariate is shown in a red wireframe, the B-rep model is presented transparent, and the final microstructure in yellow, green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

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

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

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

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

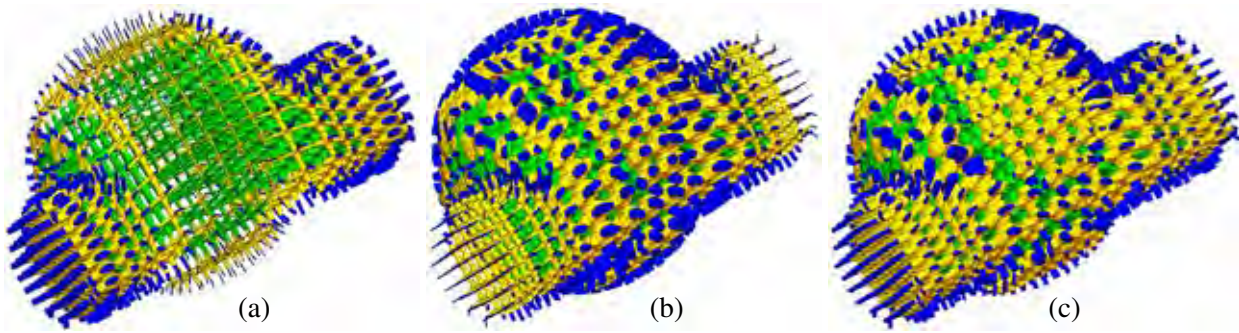


Figure 8: Several examples of variable thickness tiles in a B-rep model are shown. In (a), the tiles are thinned at the center, in the central sphere zone, in (b), the tiles are thinned toward the bases of the cylinder, and in (c), the thicknesses are modulated to follow some sine functions. Compare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

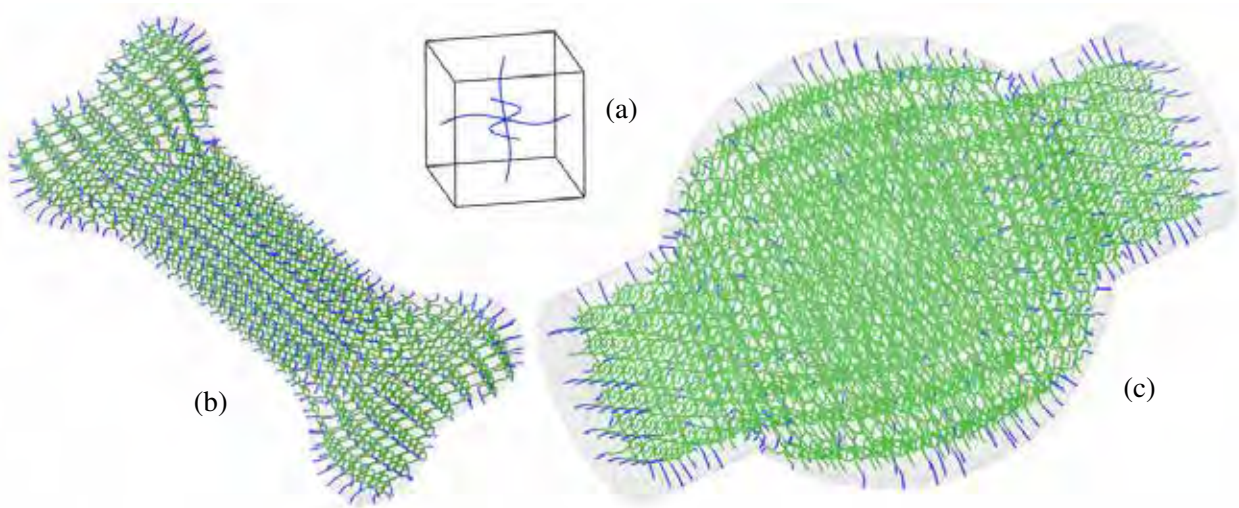


Figure 9: The tile in (a), consisting of three periodic freeform curves in a unit cube (cube is not part of the tile), is used to tile the B-rep polygonal bone model from Figure 5, in (b), and the B-rep trimmed surfaces based model, similar to the model in Figure 4, in (c). Interior tiles are drawn in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

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

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

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

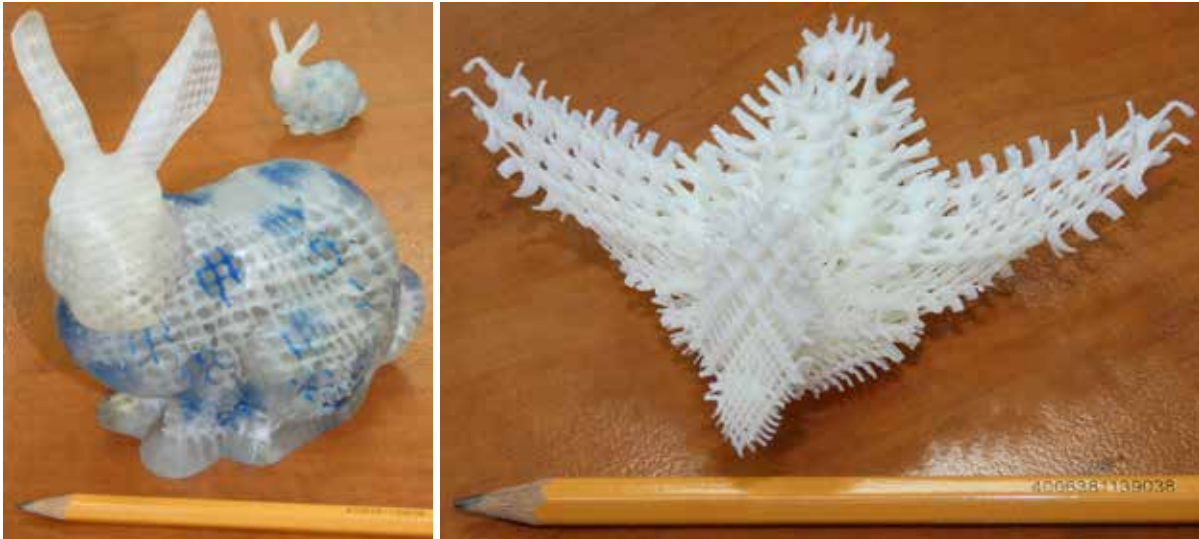


Figure 10: 3D printed versions of the Bunny (in two sizes) from Figure 6 and the duck from Figure 7. The Bunny was printed in a translucent body while for the duck, only the microstructure has been 3D printed. Printed on a J55 printer of Stratasys.

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

6 Conclusions and Future Work

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

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

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

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

discontinuity in the bridge.

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

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

Acknowledgements

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

References

- [1] T. Akenine-Moller, E. Haines, and N. Hoffman. *Real-time Rendering*. AK Peters/crc Press, 2019.
- [2] P. Antolin, A. Buffa, E. Cohen, J. F. Dannenhoffer, G. Elber, S. Elgeti, R. Haines, and R. Riesenfeld. “Optimizing Micro-Tiles in Micro-Structures as a Design Paradigm.” *Computer-Aided Design*, vol. 115, 2019, pp. 23–33. <https://www.sciencedirect.com/science/article/pii/S0010448519301939>.
- [3] A. Aremu, J. Brennan-Craddock, A. Panesar, I. Ashcroft, R. Hague, R. Wildman, and C. Tuck. “A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing.” *Additive Manufacturing*, vol. 13, 2017, pp. 1–13. <https://www.sciencedirect.com/science/article/pii/S2214860416302810>.
- [4] R. Arora, A. Jacobson, T. Langlois, Y. Huang, C. Mueller, W. Matusik, A. Shamir, K. Singh, and D. Levin. “Volumetric Michell Trusses for Parametric Design & Fabrication.” *Proceedings of the ACM Symposium on Computational Fabrication*. June 2019. pp. 1–13.
- [5] M. P. Bendsoe and O. Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer, 2004.
- [6] L. Chen, J. Huang, H. Sun, and H. Bao. “Technical Section: Cage-Based Deformation Transfer.” *Comput. Graph.*, vol. 34, no. 2, Apr 2010, pp. 107–118.
- [7] F. Conde-Rodríguez, Á.-L. García-Fernández, and J.-C. Torres. “Modeling the Internal Architecture of Composites.” *Computer-Aided Design*, vol. 129, 2020, p. 102930. <https://www.sciencedirect.com/science/article/pii/S0010448520301238>.
- [8] G. Elber. “Precise construction of micro-structures and porous geometry via functional composition.” *Proceedings of the 9th International Conference on Mathematical Methods for Curves and Surfaces*. 2016. pp. 108–125.
- [9] Q. Y. Hong and G. Elber. “Conformal Microstructure Synthesis in Trimmed Trivariate Based V-Reps.” *Computer-Aided Design*, vol. 140, 2021, p. 103085. <https://www.sciencedirect.com/science/article/pii/S0010448521000968>.
- [10] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. “Harmonic Coordinates for Character Articulation.” *ACM Trans. Graph.*, vol. 26, no. 3, Jul 2007, pp. 7–es.
- [11] T. Ju, S. Schaefer, and J. Warren. “Mean Value Coordinates for Closed Triangular Meshes.” *ACM Trans. Graph.*, vol. 24, no. 3, Jul 2005, pp. 561–566.

- [12] S. Kambampati, C. Jauregui, K. Museth, and H. A. Kim. “Geometry Design Using Function Representation on a Sparse Hierarchical Data Structure.” *Computer-Aided Design*, vol. 133, 2021, p. 102989. <https://www.sciencedirect.com/science/article/pii/S0010448520301822>.
- [13] F. Klein. “A New Approach to Point Membership Classification in B-rep Solids.” *the 13th IMA International Conference on Mathematics of Surfaces XIII*. 2009. pp. 235–250.
- [14] L. Leblanc, J. Houle, and P. Poulin. “Modeling with blocks.” *The Visual Computer*, vol. 27, no. 6, Apr 2011, p. 555. <https://doi.org/10.1007/s00371-011-0589-4>.
- [15] Y. Liu, G. Zheng, N. Letov, and Y. F. Zhao. “A Survey of Modeling and Optimization Methods for Multi-Scale Heterogeneous Lattice Structures.” *The International Journal of Advanced Manufacturing Technology*, vol. 143, no. 4, 2021, p. 040803. <https://doi.org/10.1115/1.4047917>.
- [16] F. Massarwi, J. Machchhar, P. Antolin, and G. Elber. “Hierarchical, random and bifurcation tiling with heterogeneity in micro-structures construction via functional composition.” *Computer Aided Design*, vol. 102, 2018, pp. 148–159.
- [17] A. Medeiros e Sá, V. M. Mello, K. Rodriguez Echavarría, and D. Covill. “Adaptive voids.” *The Visual Computer*, vol. 31, no. 6, Jun 2015, pp. 799–808.
- [18] A. Nazir, K. M. Abate, A. Kumar, and J.-Y. Jeng. “A state-of-the-art review on types, design, optimization, and additive manufacturing of cellular structures.” *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 9, 2019, pp. 3489–3510. <https://doi.org/10.1007/s00170-019-04085-3>.
- [19] J. Nieto and T. Susin. *Cage Based Deformations: A Survey*. Springer, 01 2013. vol. 7. pp. 75–99.
- [20] A. Pasko, O. Fryazinov, T. Vilbrandt, P.-A. Fayolle, and V. Adzhiev. “Procedural function-based modelling of volumetric microstructures.” *Graphical Models*, vol. 73, no. 5, 2011, pp. 165 – 181.
- [21] T. W. Sederberg and S. R. Parry. “Free-Form Deformation of Solid Geometric Models.” vol. 20, no. 4, aug 1986, p. 151–160.
- [22] M. Sitharam, J. Youngquist, M. Nolan, and J. Peters. “Corner-sharing tetrahedra for modeling micro-structure.” *Computer-Aided Design*, vol. 114, 2019, pp. 164–178. <https://www.sciencedirect.com/science/article/pii/S0010448519301848>.
- [23] F. Tamburrino, S. Graziosi, and M. Bordegoni. “The Design Process of Additively Manufactured Mesoscale Lattice Structures: A Review.” *Journal of Computing and Information Science in Engineering*, vol. 18, no. 4, 2018, p. 040801. <https://doi.org/10.1115/1.4040131>.
- [24] Y. Tang, G. Dong, and Y. Zhao. “A hybrid geometric modeling method for lattice structures fabricated by additive manufacturing.” *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 9, 06 2019, pp. 4011–4030.
- [25] J.-M. Thiery, P. Memari, and T. Boubekeur. “Mean Value Coordinates for Quad Cages in 3D.” *ACM Trans. Graph.*, vol. 37, no. 6, dec 2018.
- [26] B. van Sosin, D. Rodin, H. Sliusarenko, M. Bartoň, and G. Elber. “The Construction of Conforming-to-shape Truss Lattice Structures via 3D Sphere Packing.” *Computer-Aided Design*, vol. 132, 2021, p. 102962. <https://www.sciencedirect.com/science/article/pii/S001044852030155X>.
- [27] H. Wang, Y. He, X. Li, X. Gu, and H. Qin. “Polycube splines.” *Proceedings of the 2007 ACM symposium on Solid and physical modeling*. June 2007. pp. 241–251.
- [28] J. Wu, W. Wang, and X. Gao. “Design and Optimization of Conforming Lattice Structures.” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, 2021, pp. 43–56.
- [29] B. Zhu, M. Skouras, D. Chen, and W. Matusik. “Two-Scale Topology Optimization with Microstructures.” *ACM Trans. Graph.*, vol. 36, no. 4, 2017. <https://doi.org/10.1145/3072959.3095815>.

A. B-rep queries

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

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

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

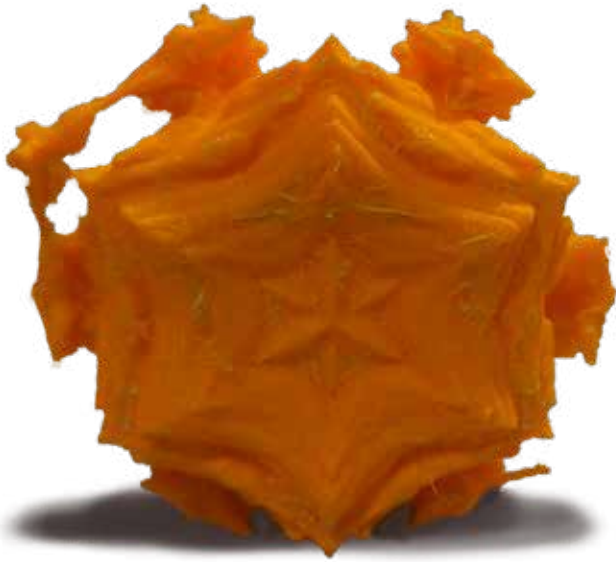
The final **BrepIntersection** test can be resolved by using Boolean operation computation where we only seek to find if the given two B-reps intersect or not.

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

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

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

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



3D Printing Hypercomplex Fractals

Emma Fogarasi and Oleg Fryazinov

The National Centre for Computer Animation, Bournemouth University, UK

Abstract

This paper describes a method to efficiently define and 3D print unique kaleidoscopic hypercomplex fractals. The proposed method allows the user to easily edit fractals in three ways: changing the power the fractal is raised to, changing the number of sides of the kaleidoscope, and the angle and position of the fractal, which creates different shapes due to the kaleidoscope. In the example of the Mandelbulb and the inverted Mandelbulb fractals, where the shape of fractals is manipulated through the “kaleidoscopic effect”, we create 3D fractal sculptures that can be realised as physical objects by using 3D printing. The flexible parametrisation of the fractal definition implemented as the Houdini tool allows the definition of many unique shapes of kaleidoscopic hypercomplex fractals that can be digitally fabricated with a little pre-processing. The paper results can be used for creating fractal-based sculptures as a part of 3D fractal art.

Introduction

Although the term “fractal” was introduced in the 20th century, self-similarities in geometry were well-known long before. The mathematics of fractals is not being considered mainstream; however, fractals consistently are used in art. Artists can use their infinite recursion to create beautiful, artistic forms that have made a basis for “fractal art”, which describes primarily computer-generated art that uses fractals in its imagery. Recently the hypercomplex fractals started to appear in feature animation films and other visual media. It is worth mentioning that fractal art is predominantly 2D, as the algorithms that create fractals in 3D are hard to manipulate.

This paper explores hypercomplex fractals and presents the method to create fractal sculptures with unique shapes. By extending the definition of the 3D Mandelbulb algorithm together with the subsequent self-defined kaleidoscopic effect, we allow defining many various fractal sculptures with allowing for artistic control at the same time. The implementation of the algorithm as a Houdini tool allows the user to create a unique shape that can be converted into a format suitable for 3D printing. We have 3D printed several models to demonstrate the applicability of the algorithm. The presented pipeline allows the hypercomplex fractals to be available as an artistic tool to be used for other artists working with fractal geometry.

Background and Related Works

Fractals are infinitely complex patterns that are self-similar on different scales. This recursiveness appears in nature, such as mountains, coastlines, clouds[1], and even blood vessels. Mathematical definitions for different fractals can vary. This paper focuses on fractals that use complex and hypercomplex numbers in their definition, specifically the 3D Mandelbulb created from the 2D Mandelbrot set.

Mathematics of fractals. The Mandelbrot set is made by iterating a simple recursive rule using complex numbers where the pattern itself appears on the complex plane. It was first discovered by Robert W. Brooks who drew it in 1978 for a study of Kleinian Groups [14]. The mathematical definition of the fractal set was done in 1980 by Benoit B. Mandelbrot. A Mandelbrot set is defined by the equation $z_{n+1} = z_n^2 + c$, where z and c are complex numbers [6], and the point is said to belong to the set if $|z| < 2$ for all iterations n . While the Mandelbrot set separates the set of complex numbers into two subsets, later it was noted that the fractal nature could be better perceived if the fractal visualisation is augmented with colours. For example, the colours of the points are determined by the number of iterations required to reach $r_{max} = 2$ where $r = |z|$. In

Figure one, the red points are the points that require 0 iterations, pink points require 1 iteration, violet points require 2 and blue points require 3 or more.

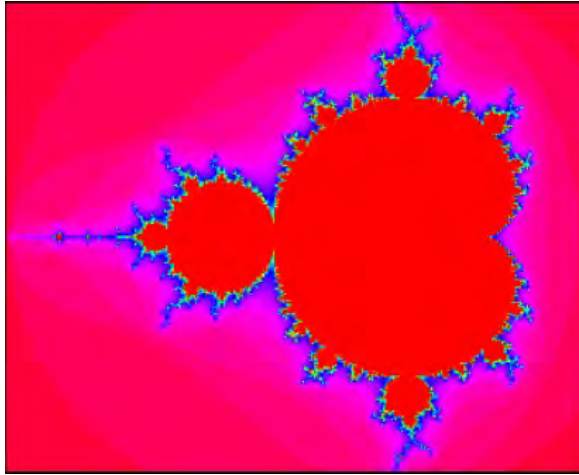


Figure 1: *The Mandelbrot set*

There are different ways to extend the Mandelbrot set from 2D into 3D. One of the ways of doing that is to rotate the set around its axis. The most well-known “3D Mandelbrot” is the Mandelbulb, a 3D shape discovered by Daniel White [7] where the extension from 2D into 3D is done by converting Cartesian coordinates to spherical coordinates, and then converting them back. The algorithm uses the definition similar to the Mandelbrot set, $z_{n+1} = z_n^2 + c$ however z and c here are hypercomplex (triplex) numbers representing 3D Cartesian coordinates. In its definition the exponential term is defined as

$$\langle x, y, z \rangle^n = r^n \langle \cos n\phi \sin n\theta, \sin n\phi \sin n\theta, \cos n\theta \rangle \quad (1)$$

Where $r = \sqrt{x^2 + y^2 + z^2}$, n is the order and angles are defined as $\theta = \text{atan2}\left(\frac{z}{\sqrt{x^2 + y^2}}\right)$ and $\phi = \text{atan2}\left(\frac{y}{x}\right)$

While the Mandelbulb is a set of 3D points that define an object with a finite volume, it has an infinite surface area due to its infinite recursion. In this paper, the Mandelbulbs are frequently referred to as Hypercomplex fractals. This is because while the Mandelbrot set uses complex numbers, Mandelbulb uses Hypercomplex numbers which are similar to complex numbers but extended into 3D [13].

Fractal art. Fractals were used in architecture such as Gothic Cathedrals [15] and art long before the term “fractal” was introduced. Fractal art, however, is predominantly 2D with only several exceptions such as fractal terrains [1] and relief carvings [3]. The main reason to stick to 2D fractals for artistic purposes is that the algorithms that create fractals in 3D are hard to manipulate.

In recent years, hypercomplex fractals have become an integral part of visual effects. They have been used in films such as the Guardian of the Galaxy Vol 2 and Annihilation. In Big Hero 6 [4], a variation of the Mandelbulb algorithm was used with parameters that allowed the VFX team to easily create a large variety of 3 dimensional forms. Similarly, our Houdini tool takes variations of Mandelbulbs with editable parameters which then can be applied for 3D printing.

“Fractal Effervescence” by David April made in 2006 is an art piece that creates fractal patterns from mathematical formulas to create a visually impactful image. The artwork was created by fusing three image files which were made with the software Apophysis developed by Mark Townsend. The three files had different types of transformations which created this image. [11]

Johan Andersson used 3D fractals to create Surreal Fractal Jewelry and Accessories soon after Daniel White discovered the Mandelbulb in 2009 [9]. One of his works involves creating fractal art by 3D printing Surreal Chess pieces. [12] His work is based on hypercomplex fractals, converting mathematical fractal algorithms to create 3D sculptures.

Houdini. Houdini is a software that uses a procedural system to create artwork. It has a node-based workflow which is ideal for visual effects as it allows their users to create dynamic simulations, and it also allows the user to build custom nodes with VEX, a language based on the C language. It is suitable for our work as it is aimed for visual effects and its custom nodes allow us to quickly visualise mathematical algorithms. [18]

Method and implementation

As mentioned previously, one of the main goals of this research is to create 3D fractal art with the visual being the main component, at the same time allowing for efficient visualization of the resulting fractal shapes. Since a Mandelbulb’s surface area is infinite, we approximate the shape of the hypercomplex fractal by representing the object with a high-resolution volumetric representation. The first step, however, is to allow for modifications in the equations that would result in different shapes, at the same time by keeping the visual appearance.

Despite some flexibility in the definition of the Mandelbulb, where the variations can be achieved by modifications of the order n (see equation 1), it is still limited in resulting shapes. To increase the variations, we have modified the formulation of the Mandelbulb (the equation 1) in the scope of our experiments.

First modification to obtain so-called “inverted” Mandelbulb was by using $\phi = \text{atan2}\left(\frac{-y}{x}\right)$. The resulting shape becomes drastically different from the original Mandelbulb, as we show in the Figure 2 (a) and (b).

To create an even larger variety of Mandelbulbs without inventing a new formula, we manipulated fractals by experimenting with the conversion of the coordinates from Cartesian to spherical and vice versa. In particular, good results can be obtained by changing trigonometric functions used for conversion from different coordinate systems and application of different coordinate system conversion in line with experiments by Paul Nylander in [13].

In figure 2 we present some shapes that we obtained with these experiments. In figure 2c we use applied different trigonometric functions to the inverted Mandelbulb to get $\langle \cos n\phi \sin n\theta, \sin n\phi \tan n\theta, \cos n\theta \rangle$ (compare with the equation 1). In the figure 2d we use further inversion of the coordinates by having $\theta = \text{atan2}\left(\frac{z}{\sqrt{|x^2 - y^2|}}\right)$. Similarly, in the figure 2e we used $\phi = (2 \sin \frac{z}{r})^{-1}$, the goal was to see how the Mandelbulb changes if the coordinates are converted differently before they are rotated. Finally, in 2f we applied different trigonometric functions as for 2c, but to the original Mandelbulb.

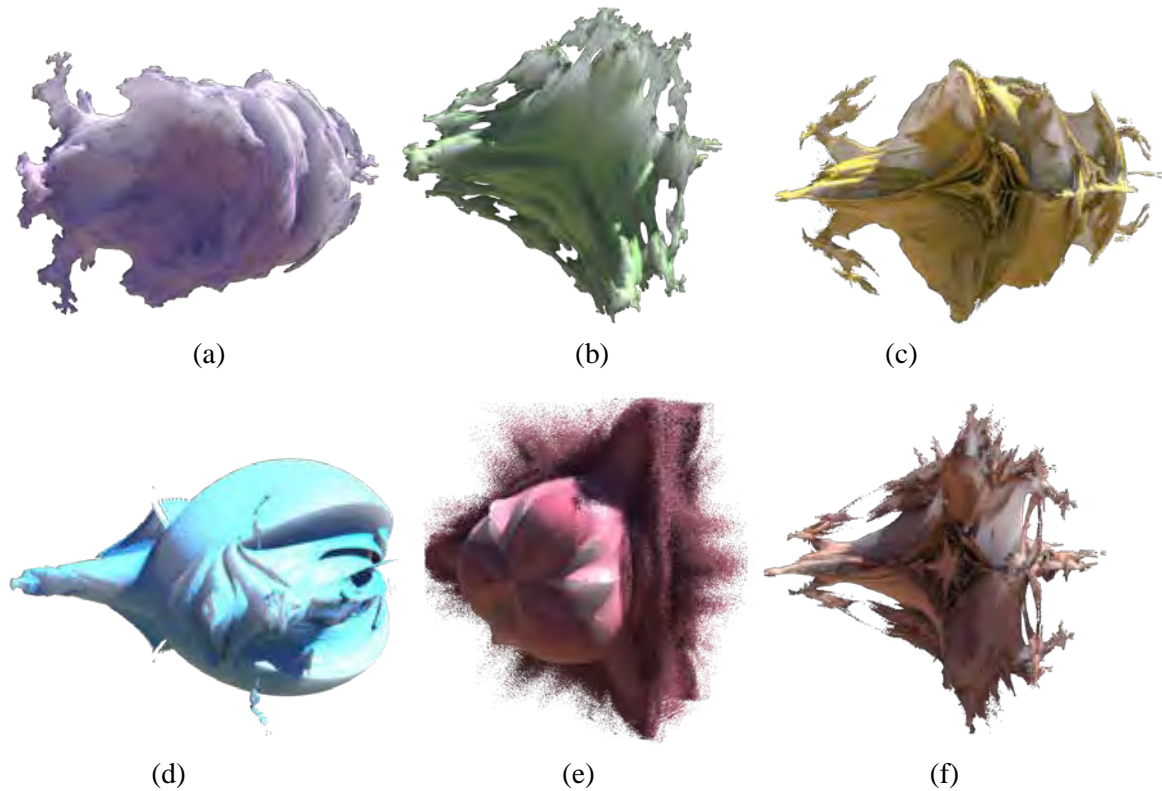


Figure 2: *Hypercomplex fractals of order 2, rendered in Houdini with subsurface scattering: (a) Original Mandelbulb, (b) Inverted Mandelbulb, (c,d) Inverted Mandelbulb with variations, (e) Mandelbulb with adjusted conversion before rotation, (f) Mandelbulb with adjusted conversion after rotation.*

To allow for larger variation in resulting shapes, we added a self-defined kaleidoscopic effect [9]. As noted above, hypercomplex fractals, on the surface, look infinitely complex. However, they are made up of repeating patterns even though they are not easily recognizable to the human eyes. We combined this with easily identifiable patterns, kaleidoscopes, to see the effect it would create. A part of the fractal is taken by using the clip node in Houdini, and mirrored by several sides through a copy node that the user defines. This takes a part of the existing Mandelbulb, mirrors it such that the edges of the parts align perfectly, and then these parts are repeated a given number of times. Due to this method, the kaleidoscopic Mandelbulb can only have an even number of sides, otherwise, one part of the Mandelbulb will not have a mirrored version and thus not connect to the rest of the mesh properly. This creates shapes that are more easily recognizable as symmetrical, which is a good design choice for 3D printing these sculptures as it is easier to balance them. The kaleidoscopic Mandelbulbs can also be edited by simply angling the Mandelbulbs to the user's liking to choose which part of the Mandelbulb will be mirrored. This allows us to create a lot of different kaleidoscopic Mandelbulbs that vary in shape. This created some dramatic changes between the fractals even though they all were produced from the same equation, as seen in Figure 3.

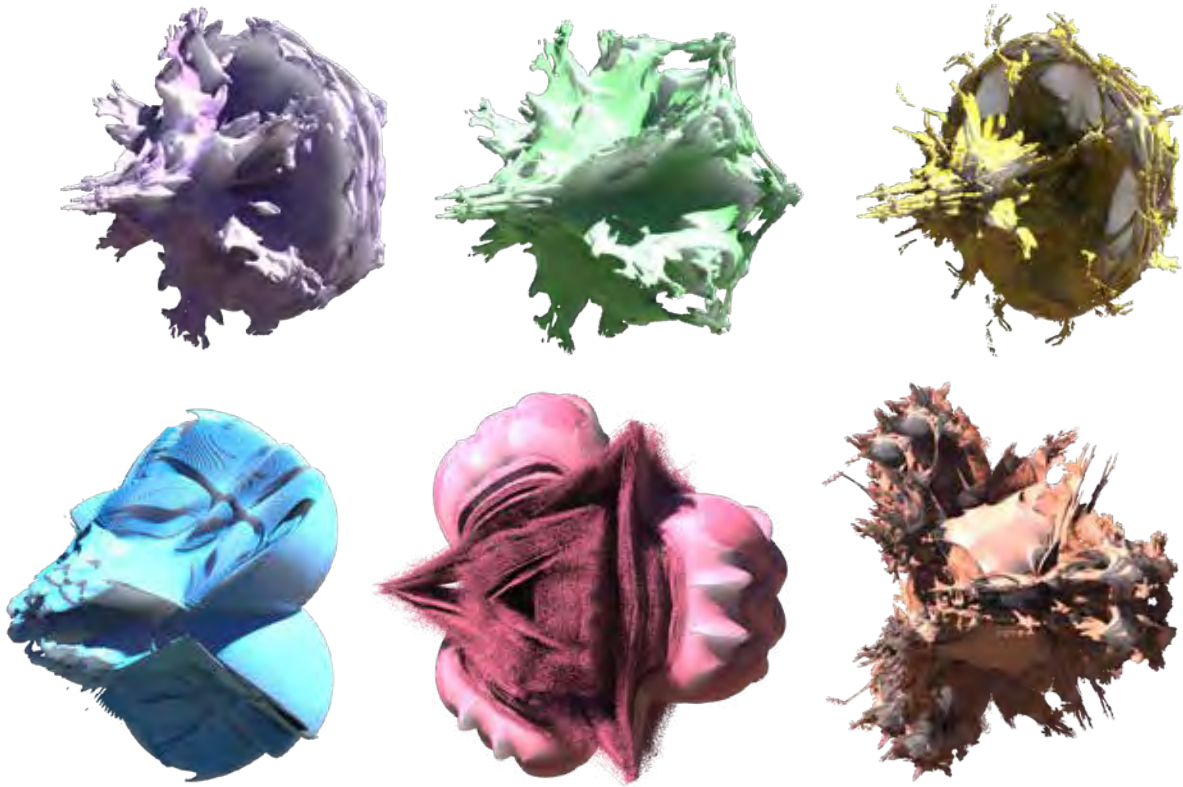


Figure 3: *Mandelbulbs from Figure 2 with the Kaleidoscopic effect applied to them.*

Figure 3 shows the previous Mandelbulbs shown in Figure 2 after applying the kaleidoscopic effect. The Mandelbulbs were purposefully angled in a way where the kaleidoscopic effect creates a visually pleasing shape. With this modification, we can create many hypercomplex fractal shapes by slightly changing the parameters. However, it is important to keep in mind that these shapes are created for 3D printing, therefore there are some limitations in the parameters. It is recommended to keep the power of the Mandelbulbs to a low number as, in this case, the resulting fractal shape has a smoother surface which is more suitable for 3D printing. It is also recommended to keep the number of sides for the kaleidoscopic effect even to avoid visible holes in the geometry and making the resulting object not suitable for 3D printing.

Houdini was used to create these fractals as it allows for quick visualization. The Mandelbulb fractal formula mentioned earlier on by Daniel White and Paul Nylander was converted into VEX code (a Houdini scripting language based on the C language) by referencing Inigo Quilez's algorithm as well [11] using a volume wrangle, which is a low-level node that allows the user to modify voxel values using code. The Vex code takes the Cartesian coordinates (x, y, z) of the points in the volume and converts them into spherical coordinates (r, θ, ϕ) . These points were then scaled and rotated and raised to the user defined power. They were then converted back to Cartesian coordinates and iterated. This node allows the user to edit the power the Mandelbulb is raised to as well as the iteration.

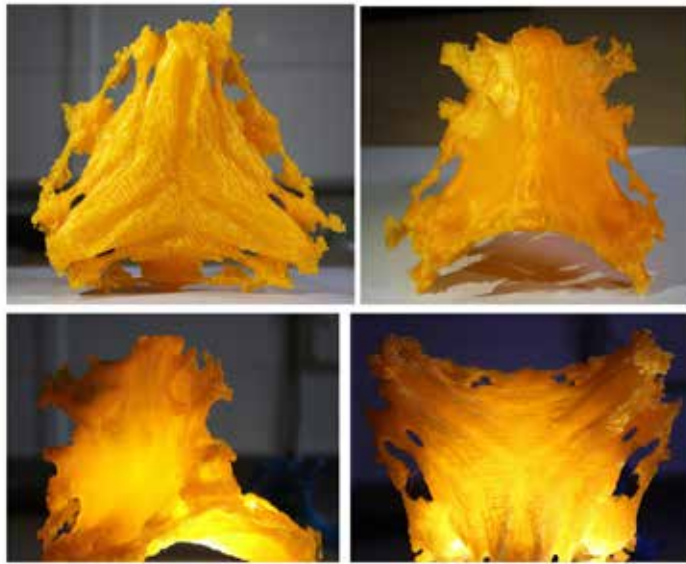


Figure 4: *Inverted 3D Mandelbulb printed in large scale with Ultimaker 2+. The support structure was removed manually.*

Results and analysis

Some of the models presented in the paper were 3D printed. We were using desktop machines of the Ultimaker and Prusa family to fabricate these shapes to demonstrate that the resulting models can be fabricated on a very low budget in a relatively short amount of time.

The original and inverted Mandelbulbs were printed on Ultimaker 3 with a soluble support structure to allow for fine details to be printed and preserved after the support structure was removed. The choice of the sculptures to print was defined by choosing the Mandelbulbs that have a mesh that has no areas that are floating and the visual impact the Mandelbulb has as a 3D printed sculpture. Both sculptures were printed quite small, on average they are 10 centimetres long. The size was chosen to be as such to keep the fine details and at the same time to make the printing process take not too long. These sculptures took about 8 hours to print. One of the objects was also printed in a larger size on single-material Ultimaker 2+ with a support structure that had to be removed manually. Although the single-material printing time on a larger scale was similar to multi-material printing, the post-processing time was annoyingly long, and some details were still lost because the support structure was impossible to remove without affecting the fine details. The resulting sculptures are shown in figures 4, 5 and 6.

The kaleidoscopic Mandelbulb was printed on Prusa MK3S/MMU2S with soluble support. It was printed quite small as well to decrease the duration of printing. Unfortunately, that was proven not the best decision as the resulting print has some very fragile parts, and the model is easier to break.

The availability of the hardware for 3D printing was somehow limited and we could not explore alternative ways to fabricate fractal shapes such as using SLS technology [16] or stereolithography (SLA) using resins [17]. This would be a better choice for these sculptures as it is capable of producing high quality prints. This could also allow for translucent materials which would increase the visual aesthetics of the final sculpture and could be used for lighting displays.

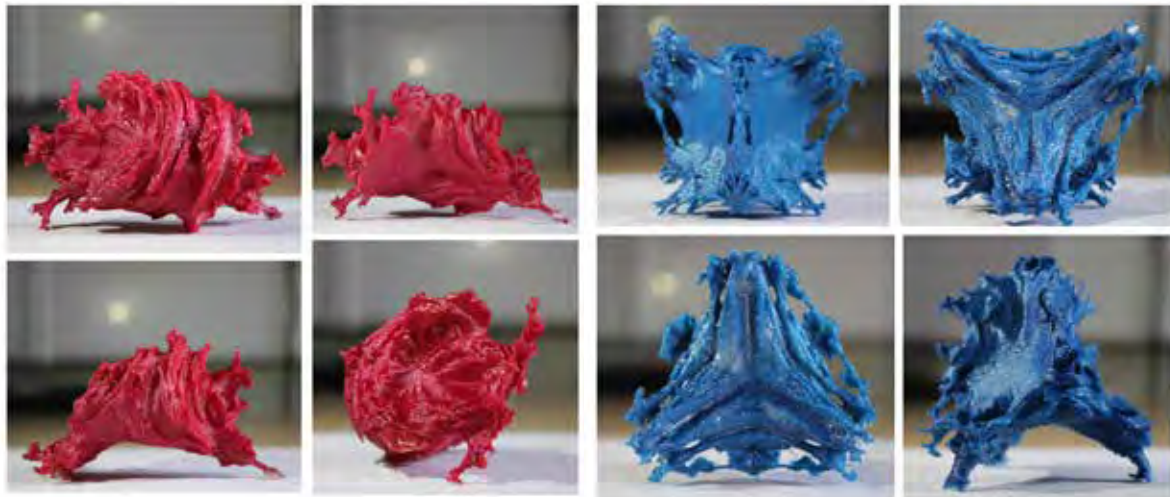


Figure 5: *Mandelbulb and Inverted Mandelbulb 3D printed on Ultimaker 3 with soluble support structure*

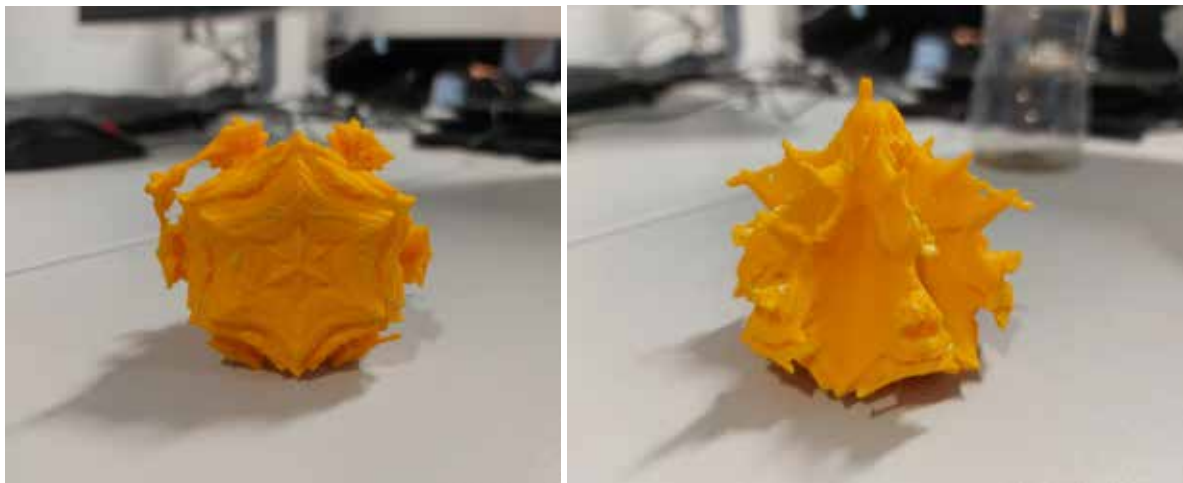


Figure 6: *Kaleidoscopic Mandelbulb printed on Prusa with a layer of PVA between the sculpture and the support.*

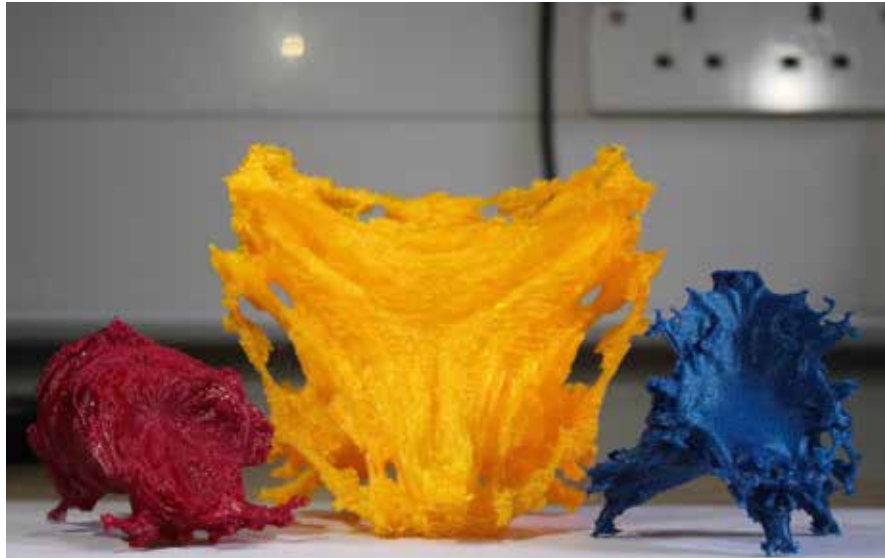


Figure 7: *The final 3D printed Mandelbulbs.*

Conclusions and discussion

Fractal art is an exciting area at the intersection of mathematics and digital art, as it allows to create infinitely complex shapes with a very simple formulation. In this work, we have shown how easy modifications of the Mandelbulb 3D fractal can result in many fractal shapes that can be converted into volume object and subsequently realised as a physical model by using 3D printing. We could achieve that by modifying the formula as well as by application of the kaleidoscopic effect implemented in Houdini software.

While the results are aesthetically pleasing and can serve as the first step into fractal art, we feel that there are some opportunities to develop the idea further and find applications in art. One of the applications of fractals in modern design is fractal jewellery. However, some modifications in shape are required to avoid very thin elements in fractal geometry as well as to ensure the printability of the shape. Automatic detection of non-printable features and correction of the volume object might be a potential future research direction. Another important aspect to consider is the artistic control of fractal shapes. For example, it would be interesting to explore how the artist can make modifications in fractal space and when the shape of the fractal is defined. We can expect that, in a nutshell, it can be done by manual sculpting over the shape of the fractal, but at the same time, an investigation into the semi-automatic methods of doing so can also be a direction for future research. To make the final sculpture more aesthetically pleasing, hand-painting or spray-painting the Mandelbulbs could achieve a more polished look.

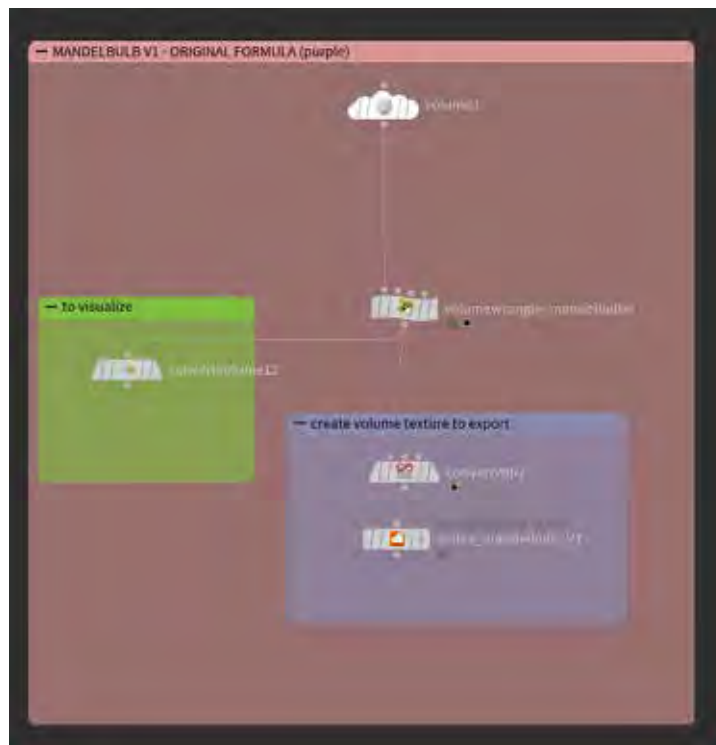
References

- [1] F. K. Musgrave and B. B. Mandelbrot, "The art of fractal landscapes," in IBM Journal of Research and Development, vol. 35, no. 4, pp. 535-540. 1991.
- [2] R.F. Voss (1988). Fractals in nature: From characterization to simulation. In: Peitgen, HO., Saupe, D. (eds) The Science of Fractal Images. Springer, New York, NY.
- [3] J. Milner. "Fractal sculpture gallery".
http://www.jimmilnersculpture.co.uk/fractal_stone_sculpture_gallery.htm.
- [4] D. Hutchins, O. Riley, J. Erickson, A. Stomakhin, R. Habel, M. Kaschalk. "Big Hero 6: Into the Portal." *ACM SIGGRAPH 2015 Talks*, New York, NY, USA, 2015. doi:10.1145/2775280.2792521
- [5] J. Ruis. "Fractal Trigeometry". <https://www.fractal.org/Formula-Mandelbulb.pdf>

- [6] B.B. Mandelbrot. FRACTAL ASPECTS OF THE ITERATION OF $z \rightarrow \Lambda z(1-z)$ FOR COMPLEX Λ AND z . Annals of the New York Academy of Sciences, 357: 249-259. 1980
- [7] D. White. "The Unravelling of the Real 3D Mandelbrot Fractal".
<https://www.skytopia.com/project/fractal/2mandelbulb.html>
- [8] J. Andersson. Surreal Fractal Jewelry and Accessories,
<https://www.shapeways.com/shops/mandelpieces>
- [9] I. Peterson. Fragments of Infinity: A Kaleidoscope of Math and Art. Wiley. 2001
- [10] I. Quilez "Mandelbulb". <https://iquilezles.org/articles/mandelbulb/>
- [11] American Mathematical Society. "Fractal Art: Beauty and Mathematics".
<http://www.ams.org/publicoutreach/math-imagery/mandelbrot>
- [12] B. O'Neal. "Fractal Artist Shows Stunning Strategy in Creating 3D Printe Surreal Chess Set".
<https://3dprint.com/40665/3d-printed-surreal-chess-set/>
- [13] P.Nylander. "Hypercomplex fractals." <http://www.bugman123.com/Hypercomplex/>
- [14] R. Brooks, P. Matelski, "The dynamics of 2-generator subgroups of $PSL(2,C)$. Irwin Kra. 1981
- [15] A. Sampler, B. Herreta, "The Fractal Pattern of the French Gothic Catherdrals", pp. 251-271. 2014
- [16] Xometry. "The Best Cheap 3D Printing Technologies". 2020
- [17] Xometry, "3D Technologies Comparison: SLA vs. FDM". 2021
- [18] SideFX.

Appendix

Node setup in Houdini:



VEX code for original Mandelbulb formula by Daniel White in Volume Wrangle:

```
VEXexpression
vector p = v@P;          //vector positions
float r = 0.0;          // radius
float power = chf("power"); // Power (N)
int maxiter = chi("Max_Iterations");

int num;

for (num = 0; num < maxiter; num++)
{
    //using Inigo Quilez' algorithm but converted to fit my formula

    //converting Daniel White and Paul Nylander's formula for a 3D Mandelbulb
    r = length(p); // sqrt(p.x*p.x, p.y*p.y, p.z*p.z)
    if (r>2) break;

    //convert to polar coordinates
    float theta = atan2(sqrt(p.x*p.x + p.y*p.y), p.z);
    float phi = atan2(p.y, p.x);

    //scale and rotate the point
    r = pow(r, power);
    theta *= power;
    phi *= power;

    //convert back to cartesian coordinates
    p = r*set(sin(theta)*cos(phi), sin(phi)*sin(theta), cos(theta));
    p += v@P;
}

if(num > 6)
    @density = 100000.0;
```


SHORT PAPERS



Turing Reaction-Diffusion -- a World in a Grain of Sand

Alyn Rockwood¹ and Irene Gironacci²

¹Boulder Graphics LLC, Boulder, CO; alynrock@gmail.com

²Swinburne University, Melbourne, Australia; igironacci@swin.edu.au

Abstract

Turing reaction-diffusion forms the basis for a multi-media, multi-technology approach to capture the mathematical/biological essence of morphogenesis in sculpture. The 2D Kassam algorithm for solving the differential equations defining reaction-diffusion is generalized to 3D. A carefully chosen threshold then generates a compelling isosurface that is polygonalized for 3D printing. Other steps in the process are: 1) Using Fourier filtering to eliminate distracting and poorly reproducible high frequency components, 2) Capping the isosurfaces within the physical printing range while maintaining a solid manifold object for manufacture, 3) Printing the polygon model in plastic and removing support struts, 4) Handcrafting extrapolations to the caps with modeling clay, while preserving tangent plane continuity to the original 3D print, and 5) Finishing aesthetic details by sanding and painting. The resulting sculpture possesses a pleasing visual flow, geometric consistency, tactile appeal, and a “nooks and crannies” element of discovery. On a deeper level, it evokes awe for Turing’s proposed mechanism of biological development and diversity.



Figure 1: The TuringB reaction-diffusion sculpture: a) view of 2D cuts, and b) full 3D view.

“To see a World in a Grain of Sand
And a Heaven in a Wildflower
Hold Infinity in the palm of your hand
And Eternity in an hour.”

-William Blake, *Auguries of Innocence*

Turing's Other Thesis

The classical (Church) *Turing thesis* asserts that a *Universal Turing Machine* can perform any calculation that is “real-world” computable. It is one of the most famous concepts in computer science. Many years after this first thesis, Alan Turing posited another, entirely different thesis of note; namely, that morphogenesis was driven by chemical reaction-diffusion. Morphogenesis is the process by which biological cells differentiate during development. In the final publication of his amazingly versatile career [3], Turing asserted that the concentration of chemical reagents drove embryonic cells to develop into more specific varieties, and that this concentration of chemicals resulted from the process known as reaction-diffusion (RD).

As with his first thesis, the thesis on morphogenesis has never been proven. Over the past 70 years, however, increasing anecdotal and scientific studies have buttressed its likely validity. By the early nineties, computers had advanced to the point where some impressive RD patterns could be produced by the computer graphics community. These were 2D, often mapped to 3D surfaces with compelling effects as in Fig. 2a [4]. Simulations have also used the theory to create synthetic digits and limbs [5]. Recent discoveries have shown that epigenetics, i.e. how and when DNA expresses itself, depends on concentrations of the methyl (CH_3) groups in the cells (methylation), though it is not clear that this concentration results from RD [6].

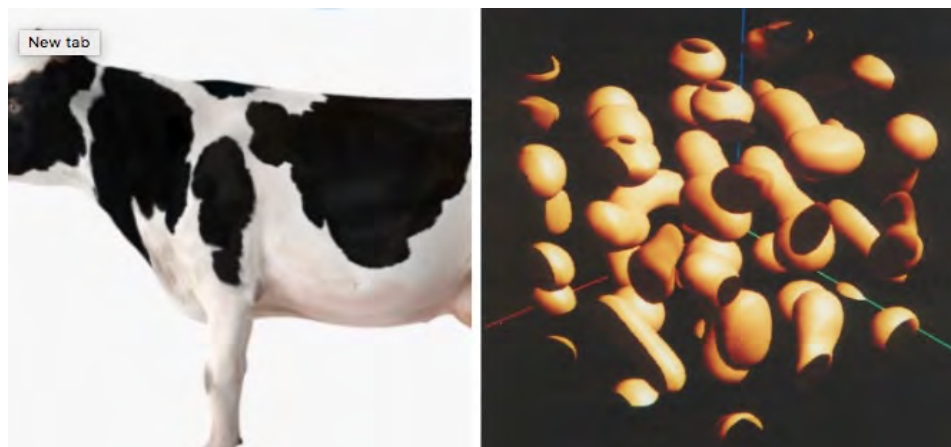


Figure 2: a) 2D RD texture mapped on a 3D surface. b) a 3D RD

In exploring RD patterns we were often struck by the intriguing appeal of the shapes that result, e.g., Fig. 1. In spite of the infinite variety of patterns possible, there was always a consistent esthetic and pleasing flow throughout the pattern. We attempt to capture that essence in sculptures. We created two objects for this paper, one to demonstrate the construction process, called TuringA and another to show the finished sculpture, called TuringB. We describe our approach in the next sections, including a brief overview of the necessary mathematical theory and 3D printing techniques, visualization considerations, and hand-crafted finishing.

Reaction-Diffusion in 3D

Chemical RD is governed by mathematical equations that determine how a mixture of two reagents change concentrations as they 1) react and 2) diffuse within each other. (The computation is a classic cellular automata, finite difference technique). Solutions at a given point in time for high resolutions can be very slow to compute. Kassam [2] improved computational speed by transforming the 2D problem to the Fourier domain and solving it in terms of frequencies, then transforming back to the spatial domain. We took their method and generalized it to 3D. (RD itself was first generalized to 3D images in [1], Fig. 2b). It is straightforward in theory to lift the RD equations to 3D, but as the noted American philosopher, Yogi Berra, once said, “In theory, theory and practice are the same. In practice, they are not.” Implementation of Kassam in 3D required new data structures, convergence tolerances, starting parameters, time step sizes, and many other algorithmic details which required a great deal of thoughtful development, and trial and error. Lomas [7] examines many of these issues.

Besides the propitious speed increases, which varied from 10 to 100 fold, Kassam’s use of the Fourier domain allowed fine surface details to be filtered out with no extra cost. This was essential for 3D printing since small details could spoil the print, leading to unusable results such as dangling threads and tears in the surface. It was a nice piece of serendipity. It didn’t change much in the overall appearance of the sculpture; it simply eliminated some sharp points and small, isolated “islands.”

RD evolves over time. In the beginning of the simulation, the minute variations in concentration showed up as an extremely complex object. By the end, the concentrations had often localized into a few prosaic regions. Somewhere between the extremes one finds the more interesting states that we used. Finding the right moment in time to sample RD was one of the aesthetic tasks.

In theory RD is computed as if the chemicals were in an infinite vat, since boundaries complicate the solution. In practice, of course, 3D printing necessitates boundaries. We implemented our theory with given boundary slices in MATLAB. Besides the necessary mathematical functions to do the RD/Fourier/numerical computations, MATLAB had functions to automatically output polygons which approximated the isosurface and could be used for display.

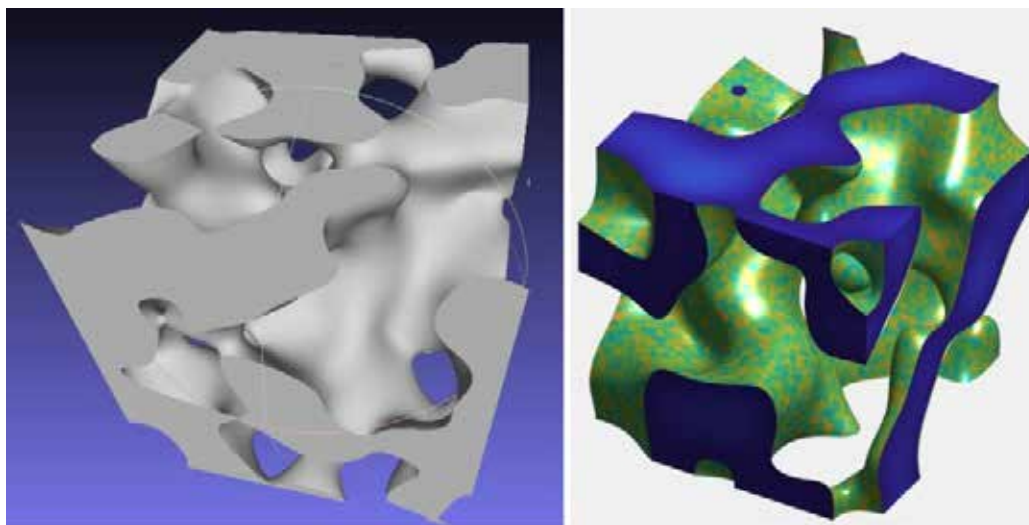


Figure 3: Different views of TuringB model a) Gouraud shaded, b) texture mapped and color coded.

Fig. 3 shows the polygon model with boundaries as output by MATLAB. We found the flat print boundaries themselves to have a charm of their own, and we colored them blue (Fig. 3b). Two-D intersection surfaces are common in morphogenesis e.g. animal hide patterns such as the Holstein cows (Fig. 2a). Computation on a 3D torus is a future option to avoid boundaries altogether.

Visualizing the 3D solution

The concentration of reagents in a solution is a function of the three dimensions of volume, like temperature in a room, or humidity in the atmosphere. Visualizing 3D functions has been an important topic for scientists and computer graphics researchers for decades. Two prominent methods are to visualize it as if the concentrations were a colored gas with opacity, which is higher in regions where the gas is more concentrated, the so-called *volume rendering*. The second is to view contours, or isosurfaces, of the function where all values are equal concentrations, e.g. using a technique known as marching cubes. Fig. 2b shows a volume rendered RD function from [1]. Whereas Fig. 3 is an isosurface rendering of our RD function. Iso-surface rendering is typically achieved by creating small approximating polygons on the surface. Such techniques are mature and readily available, as in MATLAB. The iso-surface rendering offers a convenient segue to 3D printing. The same MATLAB polygons used for rendering can also be used to generate the 3D prints. This is the pathway we took.

As an interesting aside, the polygonalized isosurface offers some attractive rendering options per se. Figure 3b was texture mapped and then lit with a light source. In Fig. 4 the surface was colored based on its curvature. Negative Gaussian curvature is warm colored while positive curvature is cool. The visualization shows that the surface is mostly saddle-like and also that it minimizes the variation of curvature. We speculate that the minimization is one reason the object feels like it has an overall consistency; one senses the harmony.

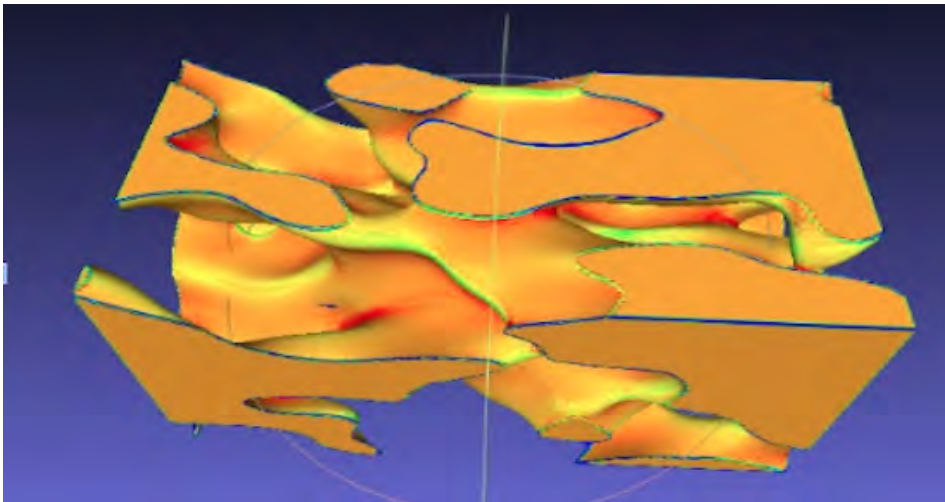


Figure 4: TuringB stretched and rendered according to Gaussian curvature.

The Devil's in the 3D Printing!

Although MATLAB ostensibly gave us the needed polygons for rendering as well as for 3D printing “automatically”, as anyone familiar with 3D printing knows, the work is far from complete. We ran the polygon model through the modeling software Blender to check for back facing polys, small gaps,

non-manifold and folded over polys etc. It's an exercise that all 3D makerspace acolytes know well. Fortunately, Blender has an array of tools to handle these issues. Additionally, we had to find isolated "islands" and eliminate them manually, as well as size the object and check that the surface could be properly supported with struts without generating too many of them. Those very hard plastic supports have to be removed eventually, and sanded clean. It can be a tedious job given the many intricacies of our Turing sculptures, so the fewer the better. We discovered the size of the prints affects not just the print time (144 hours for a one foot squared object), but more importantly, the likelihood that something will fail and the print won't complete. 3D printers still have substantial room to improve on the engineering front. We managed to complete one 8" cubed piece in only 44 hours.

We used a Lulzbot TAZ 6 3D printer. Watching the sculptures grow slowly from their base mesmerized, and made it all worthwhile. Even cutting out the struts with a Swiss army knife while only partially avoiding being nicked by their sharp ends created satisfaction as the shapes of the objects began to emerge. The white plastic in Fig. 5a is the 3D print for TuringA; the terracotta part is the clay..

Modeling Clay and Final Touches

We had two guiding principles for rounding off the flat sides of a 3D print: 1) The clay extension should maintain tangency to the curved surfaces of the 3D print, and 2) The surfaces should stay consistent with the aesthetic feel of the 3D print, mainly by minimizing the maximum curvature, and repeated eyeballing.

The modeling became a purely artistic effort. It used an air-dried terracotta clay, which had all the tactile/visual joys of pottery or any other sculpture form. Often the 3D print would strongly suggest the shape, other times the clay itself would reveal interesting directions. Sometimes the clay was chiseled off, thrown into the wastebasket and restarted. Fig 5b shows TuringA after sanding, but before painting.

In TuringB we chose to leave three sides flat. As mentioned above, the 2D RD of the boundaries was also interesting. In 3D the flow and turbulence of the object suggested a liquid metal. After some experimentation we decided that a silver tone seemed to belong. The light metal brought out the highlights, and effectively showed off the internal contours. On the other hand, the terracotta of TuringA suggests an organic finish, perhaps a flesh-toned theme? Something yet to be determined.



Figure 5: TuringA sculpture a) clay being added, b) while sanding.

Conclusion

The finished TuringB was invited to be part of a show at the Boulder Canyon Gallery, Boulder, CO. It was exhibited from Feb. 4 until Mar 28. See Fig. 6. It was displayed on a raised plexiglas pedestal over a mirror so that all the colored 2D sides could be viewed along with the 3D sculptured sides. Gallery visitors were encouraged to handle the object, and explore it by touch. In one case, a blind woman examined it for an extended period as Turing's thesis was explained to her. As she returned the sculpture she said, "It's like holding the beginnings of life in my hands." She perfectly captured the original motivation for doing RD sculptures. In the creation process, however, more pleasure was discovered in the intrinsic harmony, intricacy, and flow of the RD objects.

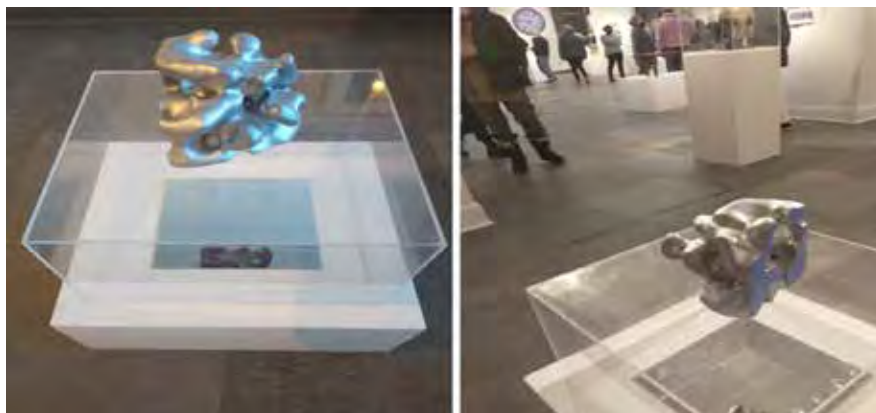


Figure 6: TuringB sculpture on exhibit, mounted on plexiglass over a mirror.

Acknowledgements

Swinburne UT, Melbourne, Aus. provided the subscription to MATLAB. Boulder Library, Boulder, CO provided the 3D printer, where Zack Weaver and Adam Watts gave generous guidance and help.

References

1. Chambers, P.; Rockwood, A, IEEE Computer Graphics and Applications 1995 Vol. 15; Iss. 5 · Visualization of solid reaction-diffusion systems.
2. Kassam, A., "Solving reaction-diffusion equations 10 times faster", Oxford University Computing Laboratory, Wolfson Bldg., Parks Road, Oxford OX1 3QD, UK.
3. Turing, Alan (1952). "The Chemical Basis of Morphogenesis" (PDF). *Philosophical Transactions of the Royal Society of London B*. **237** (641): 37–72.
4. Turk, G. Generating synthetic textures using reaction diffusion. ACM SIGGRAPH Computer Graphics Volume 25 Issue 4 July 1991 pp 289–298
5. Zhu, Jianfeng; Zhang, Yong-Tao; Alber, Mark S.; Newman, Stuart A. (28 May 2010). Isalan, Mark (ed.). "Bare Bones Pattern Formation: A Core Regulatory Network in Varying Geometries Reproduces Major Features of Vertebrate Limb Development and Evolution". *PLOS ONE*. **5** (5): e10892.
6. "DNA Methylation" in *What is Epigenetics?*
<https://www.whatisepigenetics.com/dna-methylation/#:~:text=DNA%20methylation%20is%20an%20epigenetic,genes%20and%20affecting%20gene%20expression.>
7. Lomas, A. (2014) Cellular Forms: an Artistic Exploration of Morphogenesis. AISB-50, London, UK, 1–4 April 2014. Goldsmiths, University of London.

*The Proceedings of the SMI-SCULPT 2020
Shape Creation Using Layouts,
Programs, & Technology (SCULPT)*

*In Cooperation
with ISAMA*



FASE Paper Chairs:

Oleg Fryazinov
Carlo H. Séquin

Proceedings Editor:
Ergun Akleman

Front Page Sculpture:
Alyn Rockwood, Irene Gironacci

Back Page Sculptures:
Emma Fogarasi, Oleg Fryazinov

Cover Design:

Ergun Akleman

Fractal Sculptures

