HYPERSEEING

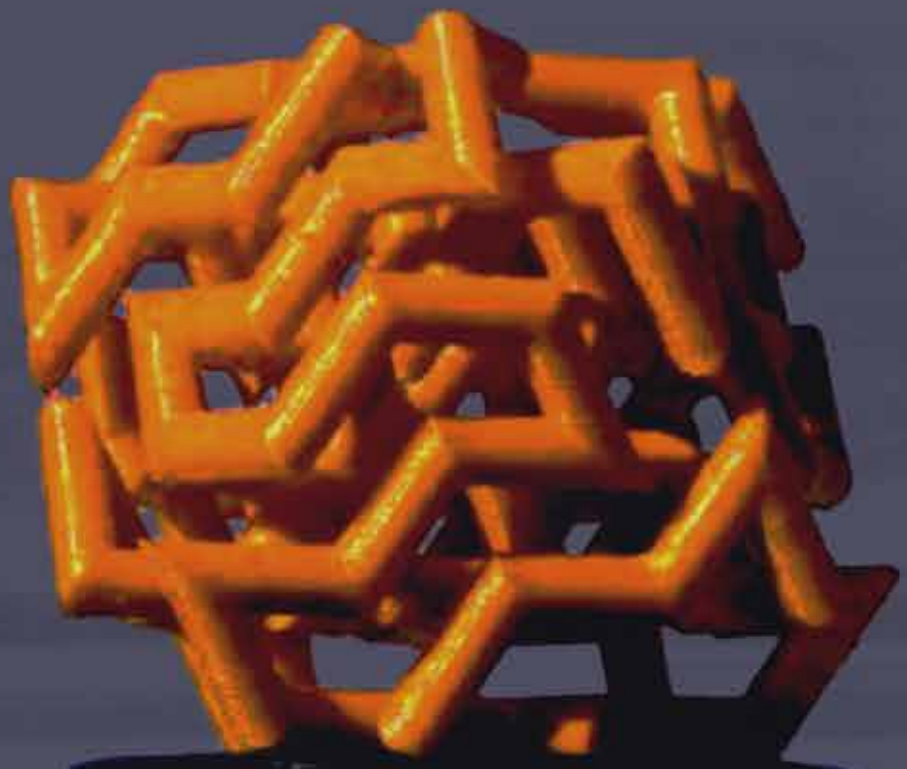
The Publication of the International Society of the Arts, Mathematics, and Architecture

FALL 2021
www.isama.org

The Proceedings of the SMI 2021 Fabrication & Sculpting Event (FASE)

Special Issue on Shape Modeling International 2021

14-16 Nov. 2021 Online



HYPERSEEING

Special Issue on SMI 2021

Shape Modeling International 2021
Fabrication and Sculpting Event

Twentieth Interdisciplinary Conference of the International Society of the Arts, Mathematics, and Architecture

Online Event
November 14-16, 2021

SMI Conference Chairs

Ergun Akleman (Texas A&M University, USA)
Stefanie Hahmann (Inria, France)
Jorg Peters (University of Florida, USA)

FASE Paper Chairs

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

Steering Committee: Ergun Akleman, Loic Barthe, Karina Rodriguez Echavarria, Oleg Fryazinov, Guiseppe 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

FASE Supported by

Bournemouth University
Texas A&M University
UC Berkeley

CONTENTS

Proceedings of Fabrication and Sculpting Event

Preface

History Fabrication and Sculpting Event (FASE)

The Fabrication and Sculpting Event started as an experiment in expanding the scope of shape modeling international (SMI) conference in 2012. 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.

The main difference with other math/art conferences is that FASE focuses solely on 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

Fabrication and Sculpting Event 2021

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.

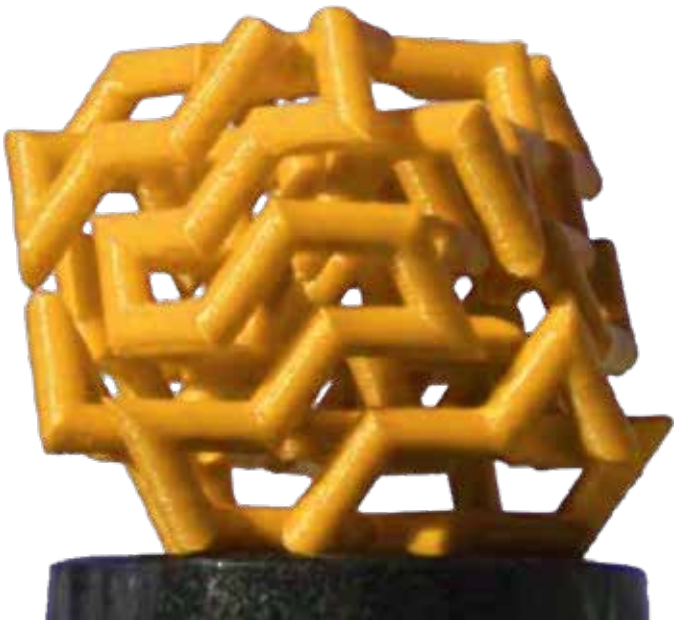
Thus, the scope of FASE is the intersection of shape modeling and fabrication methods/algorithms, and papers may focus on both the digital/theoretical and the physical domain or just one of these domains – as long as the connection to the other domain is clear. It is not a requirement that the techniques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

We received seven submissions this year and three of them were accepted as twelve-page or longer full papers, and another two 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 2021 Fabrication and Sculpting 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 2021 will enjoy this event of the conference.

Oleg Fryazinov, and Carlo Séquin

FASE Papers chairs

FULL PAPERS



SHELL-LATTICE CONSTRUCTION BASED ON REGULAR AND SEMI-REGULAR TILING VIA FUNCTIONAL COMPOSITION

Sumita Dahiya[1], Avi Shein[2] and Gershon Elber[3]

[1]Department of Computer Science, Technion, Israel; [1]sdahiya@campus.technion.ac.il,
[2] Tufts University
[3]Department of Computer Science, Technion, Israel

November 3, 2021

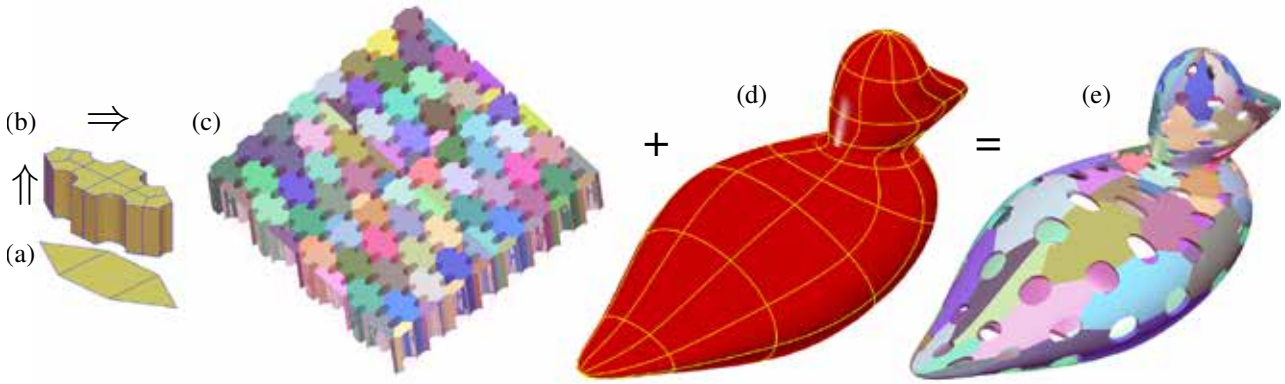


Figure 1: (a) Primal Semi-regular tile 43433. (b) Its dual tile 43433. (c) Periodically tessellated domain $[0, 1]^2$ using dual tile 43433, and randomly colored tiles. (d) Ruled shell volume between the duck surface and its offset. (e) Domain (c) is functionally composed into (d), yielding a shell-lattice structure.

ABSTRACT

This manuscript outlines a methodology and algorithm for the design of shell-lattice structures based on dual regular and semi-regular planar shapes. It introduces a 2-manifold, parametric freeform shell type of tiling to tessellate the domain of any given bivariate deformation map. The tessellated domain is functionally composed into the deformation map to obtain freeform shell-lattice structures in 3-space. Given any primal regular or semi-regular pattern, intrinsic controls over the thickened parameters of the dual graph are provided, results in a variety of tilings that differs in both physical properties and aesthetic appearance. Curve, surfaces, and trivariate tilings are offered, allowing analysis, optimization and fabrication of the shell-lattice results. This work concludes with several results from the implementation of the algorithms, including of 3D printed parts.

1 Introduction

1.1 Problem and Context

Regular prisms such as rectangular or cubical blocks are still the most used geometries to build up volume or shell structures, credit goes to their availability and ease of manufacturing. Architects and designers aim to obtain a prominent design space that consists of unconventional modular building blocks allowing variations in geometric, topological and

material properties. To achieve that, along with optimization of the geometry, topology and materials, a more systematic investigation is required in this direction.

For obtaining reliable and robust structures, there is a need for simple and formal approaches that provide intuitive designs and intrinsic control of a wide variety of modular building blocks. In this work, we introduce a dual version of semi-regular tiles which can be used to design and model intricate 2D lattice tiles. In the proposed approach, shell-lattice structures are obtained by functionally composing these dual (semi-) regular tiles and some deformation map. This dissociation between the micro and macro lattice structures provides the end users with a simplified and intuitive control over the design space and parameters. A simple yet flexible framework has been devised which can be effectively harnessed for modeling, designing, optimizing, and fabricating tile-able building blocks, in a variety of shell structures.

1.2 Inspiration and approach

Now a days, a wide variety of artifacts with intriguing and intricate properties can be fabricated with much ease, given the advancement achieved in additive manufacturing techniques. Use of heterogeneity and composites has expanded the manufacturing capabilities manifolds and opened up new frontiers to explore, for almost every industry. The approach proposed in this work leads to the construction of shell structures while optimizing the physical properties of the structures with respect to their geometry or their material, for example, striving for a high strength to weight ratio. Lattice structures are typically constructed using one or more unit cells that are periodic in nature and can be tessellated along any given axis [7]. Thin-shell or simply shell structures are lightweight constructions, made of typically shell (skin-like) elements, and are typically curved. These elements are then assembled to make large shell structures. The shell-lattice structures can widely be used in various industries, like automotive, aerospace and ship-building industries. These industries rely heavily on computer aided design to design structures and analyze their different physical behaviors, e.g. heat, fluid flow, or stress. Our goal is to construct these shell like structures and characterize their properties in a concise and intuitive way so as to allow for an informed exploration of the design space.

The foundation approach for the presented research work is the functional composition of the 2-manifold deformation map and the domain of the deformation map using copies of a dual (semi-) regular tile, i.e. functional composition [4, 10] of a given deformation map, $\tau : \mathbb{R}^2 \to \mathbb{R}^3$ and a tile, $T \in \mathbb{R}^2$, algebraically computing $\tau(T)$, i.e. image of T through τ.

1.3 Contributions

(i) A design space of 2D tilings based on dual regular and semi-regular tessellations of the plane, with results that are aesthetically pleasing.

(ii) Families of 2D tilings that can create shell-lattice structures of 3D geometries, while supporting analysis, optimization and fabrication of the same, with respect to the geometry, topology, and material/heterogeneity.

The rest of the paper is organized as follows. In Section 2, we survey related previous work. Then, Section 3 describes our entire algorithm to populate a general bivariate deformation map with tiles. Some examples and results are presented in Section 4, only to conclude in Section 5.

2 Related Work

Lattice-shell structures, widely exploited in civil, transportation and aerospace engineering, among others, are not always regular three dimensional structures nor planar. These thin wall structures like sandwich panels [17] and cylindrical shells with lattice cores [16] characterized by excellent mechanical and thermal properties, are used, for example, as draw tubes of space telescope, support structure of antenna in satellites and space platforms [22, 1]. Design and analysis of lattice-shells has gained a lot of attention in recent times. Wei et al. [18] worked on example based texture synthesis algorithms and Dumas et al. [3] extended this work to on-surface texture synthesizer that can work on voxel shell around the surface. Their algorithm can synthesize carved patterns along any given surface that can be fabricated over fused-filament printers. They have targeted surfaces in the form of a triangulated mesh, and the input is a colored image with binary pattern map to be synthesized along the surface. Although both the approaches results in structural patterns along surfaces, in the present study, our inputs are a B-spline bivariate surface and a dual (semi-) regular tile, which can be consisting of univariate curves, bivariate surfaces (possibly tessellated into a polygonal mesh), or trivariates.

Geometric regularities have always been intriguing for researchers working in the field of architecture and shape modeling. These aesthetically pleasing geometries, in the form of 3D tilings can show complex mechanical behavior, with different patterns characterizing different properties like anisotropy, variable stiffness and heat conductivity, etc.

Tessellations in the plane have been intensively studied and worked upon in mathematics [11] as well as in graphics [9]. Schumacher et al. [13] studied the mechanical properties of different polygonal tiling, by fixing the input base material and 3D printing different pattern prototypes. In their work, the sheet material is a thin extrusion of two-dimensional tilings. Minimal tileable units are identified and mapped to cylinders to study bending deformations corresponding to uniaxial curvature.

Trautz et al. [15] used regular, irregular and freeform geometries for the construction of shell and spatial structures based on the folded plate principle. They conclude that folding principle along with the freeform geometries can give rise to widely spanned light weight structures with high-strength. Weizmann et al. [19] experimentally studied the structural behavior of semi-regular and non regular topologically interlocking assemblies. They analyzed the structural performance based on the geometry and dimensions of topologically interlocking blocks. Given that each regular polygon is cyclic, Subramanian et al. [14] designed dual meshes relying on the cyclic properties of the regular and semi-regular tilings. A wide variety of artistic patterns of 2D tiling is shown by Kaplan [8], and Rao [12] has given an algorithm for construction of 2D pentagonal tiling.

Modeling constructor for micro-structures and porous geometries has been introduced by Elber [4] that can work for curve-trivariate, surface-trivariate and trivariate-trivariate function composition. By using 1-, 2- and 3-manifold based tiles and paving them multiple times inside the domain of a 3-manifold deforming trivariate function, smooth, precise and watertight, yet general, porous/micro-structure geometry might be constructed, via composition. Massarwi et al. [10] presented extensions to this micro-structure construction technique to construct fractal like heterogeneous, water-tight models with a high degree of precision.

Our presented work, in a sense, expands semi-regular planar tilings beyond tilings in 2D space to a rich design space of tilings in 3D space, while exploiting duality. Our goal is to construct shell-lattice structures, by utilizing the properties of semi-regular tilings, that can be studied and further utilized for, for instance, their mechanical and thermal characteristics. We aim to expand the design space that can be subjected to an informed exploration, and offer a continuum as families of such tilings, as univariate curves, bivariate surfaces, and trivariates shell elements, allowing for the design, analysis, optimization and fabrication of shell structures.

3 The algorithm

The space of regular and semi-regular tilings is partitioned into a set of distinct families, each of which admits certain tileability-preserving transformations on the shape of the tile. Parameterizing these shapes within a given family, can yield a continuous sub-space of patterns of the similar topology but potentially vastly different mechanical properties, for example solid-density[1], strength or heat conductivity.

This section is organized as follows: Section 3.1 presents the regular and semi-regular tiling topologies, and in Section 3.2, we discuss how the dual tile is constructed and controlled. In Section 3.2.1, we consider the tiling in the domain, tilings that are composed into the deformation function, as presented in Section 3.3.

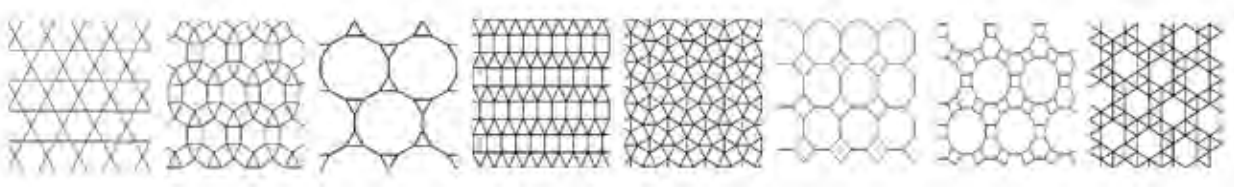
3.1 Regular and semi-regular tilings

A tiling in the plane comprising of regular polygonal faces and isomorphic vertices, is commonly called as Euclidean regular tiling. Given that for a regular tiling (m, n), m is the number of sides in each face and n is the valence of vertices, there exists only three regular tilings characterized as $(3, 6)$, $(4, 4)$ and $(6, 3)$. In semi-regular tilings, more than one type of polygons exist, provided that the polygons are regular and all vertices have exactly the same structure of polygons around them. One naming convention of the different tilings stems from the types of polygons around each vertex where 3, 4, 6 denotes triangle, square and hexagon. For example, the semi regular tiling 44333 has two squares and three triangles around each vertex. There exists only eight semi-regular tilings summarized as $3636, 4346, 12312, 44333, 848, 43433, 1246$ and 333336, as shown in Figure 2.

3.2 Thickened dual semi-regular tilings

The notion of duality is one of the central concepts in geometry. It is generally defined in a very definite way using concrete structures. Duality is an existing property of geometric and algebraic structures which states that two concepts or operations can be used interchangeably, with both formulations holding the same results. The dual of a regular or a semi-regular tiling can be formed by considering center of each polygon as vertex and thereby joining the centers of adjacent polygons. The triangular and hexagonal tessellations are duals of each other, while the square tessellation is

[1] We define solid-density as the ratio of solid matter vs. air in the lattice, e.g. 100% is a full solid.



(6363) (4346) (12312) (44333) (43433) (848) (1246) (33336)

Figure 2: Semi-Regular tessellations.

its own dual. It is interesting to note that the dual of any given semi-regular tiling always consists of only one type of polygon and even that can be irregular. Duality allows the user to have a higher level of control over the shape pattern. And for some specific values of the control points the dual becomes the primal itself.

Our dual tiles of the semi-regular tiling are constructed by adding some thickness and curvature to the edges of the dual graph. This thickness is controlled by two additional parameters: a curvature sharpness (D_{curv}) of the thickened edges of the dual graph, and a fraction of the length of an edge in the primal tile (D_{frac}), where the thickened edge ends. Figure 3 shows a few examples, where we adjust D_{curv} and D_{frac}. One should note that if the value of D_{frac} = 0, D_{curv} holds no meaning. Also, if $D_{frac} = 1$ and D_{curv} = 1, the dual tile outlines the corresponding primal tile.

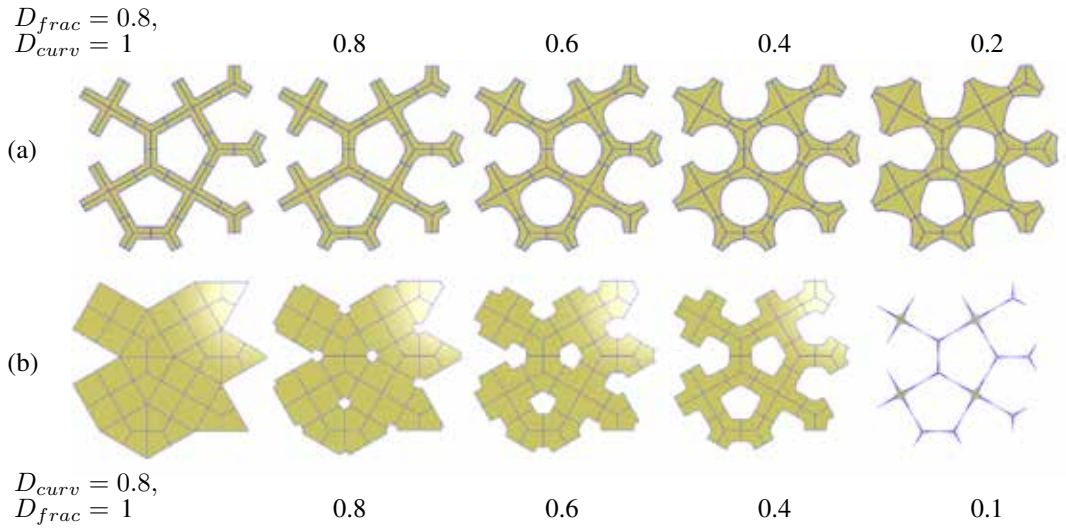


Figure 3: Visual variation in dual tile 43433 for (a) fixed value of D_{frac} and varying values of D_{curv} and (b) fixed value of D_{curv} and varying values of D_{frac}.

So far the tiles are merely univariate curves in the plane. However, we also allow surface and trivariate tiles. Surfaces are constructed by ruling opposite curves in the plane into planar surfaces, and trivariates are constructed by extruding the surfaces outside the plane to a desired elevation. See Figure 4.

3.2.1 Minimal tileable unit and lattice-tile

The minimal tileable units for planar tessellations, are the smallest translational units that can be paved across a 2D domain using translation operation only. See Figure 5, for minimal semi-regular tileable units, in black. The dual counterparts are shown in yellow color in Figure 5. Every tessellation shown in Figure 2 is built using one of these minimal tileable units or unit tiles. Each of these unit tiles create an infinite periodic tiling of the plane using translation only, emphasizing that (a regular or) a semi-regular tiling has translational periodicity.

3.3 Composing it all into a deformation map

Following [4], a tile is taken along with a deformation map. In this functional composition, we have used tiles that are either curves, or surfaces (possibly tessellated into a polygonal meshes), or trivariates, while the 2-manifold deforming function is a B-spline bi-variate surface. Clipping or bounding of the tiled domain is done by bringing both the tilings

(a)

(b)

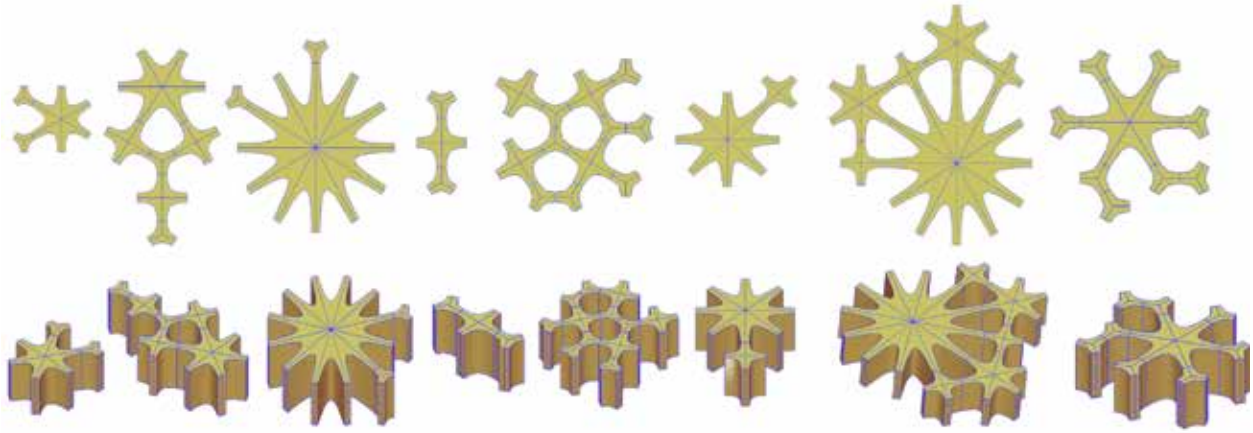
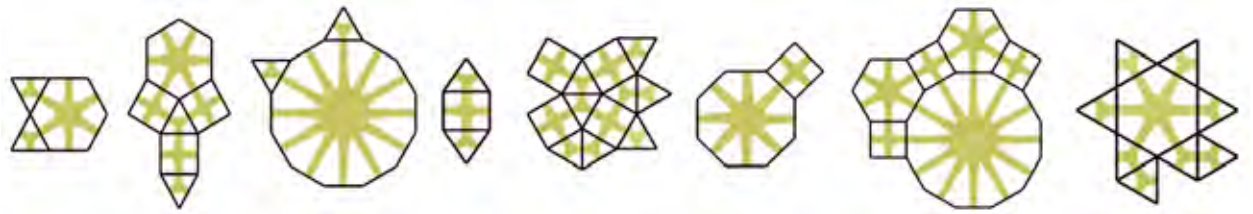


Figure 4: Given the univariate curves representing the dual tile, as in Figure 3, a planar surface representation for the tile is created by ruling opposite curves in the plane, as is seen in (a). By extruding the result of (a) outside the plane, trivariate tiles can be formed, as in (b). The order of the different tiles here is the same as in Figure 2, left to right.



(6363) (4346) (12312) (44333) (43433) (848) (1246) (33336)

Figure 5: Minimal tileable units for Semi-regular tilings (in black) and the corresponding dual tiles (in yellow). The order of the different tiles here is the same as in Figures 2 and 4, left to right.

Algorithm 1 CreateDualTile - computes a dual tile of curves, given a primal tile.

Input:

T := a primal (semi-) regular tile, in R^2; ▷ Section 3.1

$TileSize$:= edge length of T;

D_{frac} := a fraction $\in [0, 1]$ of a primal edge;

D_{curv} := curvature sharpness of the thickened dual edges;

Output:

T_C := a (semi-) regular dual tile, of univariate curves, in R^2;

Algorithm:

1: $\mathcal{D}$:= the dual graph of T;
2: $T_C := \emptyset$
3: **for** every edge $E \in T$ **do**
4: $\{P_0, P_2\}$:= points on E with fraction $(D_{frac} : 1 - D_{frac})$ and $(1 - D_{frac} : D_{frac})$;
5: P_1 := point interior to T, between $\frac{P_0+P_2}{2}$ and closest location on $\mathcal{D}$, with ratio D_{curv};
6: C := A quadratic Bézier curve, using (P_0, P_1, P_2) as control points;
7: $T_C := T_C \cup \{C\}$;
8: **end for**
9: return T_C scaled to $TileSize$;

---

and the deforming function to domain $[0,1]^2$. This clipping/bounding is done so to satisfy the condition that the tiled domain must be contained within the domain of the deformation map. Further, if the deformation function is closed or periodic, effort should be made to ensure the 2D tilings are also periodic along that edge. This entire process is summarized in Algorithm 1 and is depicted for a dual tile 44333 in Figure 1.

Algorithm 2 CreateShellLattice - computes a shell lattice, given a dual tile and the shell surface.

Input:
T_C := a (semi-) regular dual curve tile, in R^2; ▷ Algorithm 1
D := a bivariate deformation function, D: $R^2 \rightarrow R^3$;
$ShellThickness$:= offset amount to set the shell thickness;
Output:
$\mathcal{D}$:= A shell-lattice structure;
Algorithm:
1: $T_G := T_C$ either as curves, or elevated into surface- or trivariate-geometry; ▷ Section 3.2
2: $\mathcal{A}$:= arrangement of tile in $[0,1]^2$, using T_G;
3: $\overline{D}$:= D Reparametrized to domain $[0,1]^2$;
4: $\overline{D}_{ofst}$:= $\overline{D}$ offset inside, by amount $ShellThickness$;
5: $\mathcal{D}_T$:= ruled trivariate between $\overline{D}$ and $\overline{D}_{ofst}$;
6: $\mathcal{D} := \mathcal{D}_T(\mathcal{A})$, the image of $\mathcal{A}$ through $\mathcal{D}_T$;
7: return $\mathcal{D}$;

4 Results

The definitions and algorithm described in the previous sections are implemented in the IRIT [5] solid modeling kernel. We hereby present some additional results from this implementation. All present results were created on an Intel Core i7-7700K 4.2 GHz PC with 32GB of main memory, and a single thread. Table 1 gives total time elapsed in the functional composition of the tiled domain for different 2-manifold deformation maps, constructed with different dual semi-regular tiles. A B-spline bivariate duck surface and the Utah teapot with four B-spline surfaces are considered.

Table 1: CPU time (in seconds) for the composition of different dual tiles with two deformation maps, a bivariate B-spline duck surface, an an Utah teapot with four B-spline surfaces.

Dual Tile	Duck, D_{frac} = 0.8, D_{curv} = 0.6	No. of Polygons composed in Duck	Teapot, D_{frac} = 0.8, D_{curv} = 0.6
44333	0.62, see Figure 6 (d)	349,088	3.31, see Figure 9 (d)
6363	1.20, see Figure 6 (a)	633,600	6.14, see Figure 9 (a)
33336	2.06, see Figure 6 (h)	1,030,080	10.15, see Figure 9 (b)
4346	2.28, see Figure 6 (b)	1,112,624	11.46, see Figure 9 (c)
43433	2.75, see Figure 6 (f)	1,339,968	13.51

Tiles comprising of curves, surfaces (possibly tessellated), or trivariates have been used, and randomly colored for clarity. The tiles shown in Figure 5 have been paved across the domain and then composed in the bivariate map 1 (d) to obtain the shell-lattice structures shown in Figure 6. Each tile in these shell-lattice structures is shown in different color, a practice that we exploit throughout. Figure 6 shows compositions for the different dual tiles. Each composition in Figure 6 varies in density depending lattice properties because of the varying density distribution of the different design patterns. The duck in Figure 6(c), constructed with tile 12312, shown in Figure 5, varies much in solid-density from the duck in Figure 6(d), constructed with tile 44333. Concurrent or local adjustment of the design patterns, material used and shell layer shapes or the deformation maps can be a deciding factor, for example, for the required mechanical or thermal characteristics of the final shell-lattice structures.

The algorithm results in shell-lattice structures that can have different solid-density. For a given deformation map, there are several ways to vary this solid-density: using a different topology, varying the D_{curv}, or varying D_{frac}. Figure 7 shows four different versions of the deformation map with dual tile 33336, for different values of D_{curv} and D_{frac}. Varying these parameters, results in shell-lattice structures that are significantly different from each other in appearance, solid-density and other physical properties, for example, strength, for the same topology. Similarly, in Figure 8, four shell-lattice structures are shown, made out from the dual tile 44333, by varying the values of the control parameters.

The algorithm is also verified on a different shell structure, comprising of a number of B-spline surfaces. Figure 9 shows four images of the Utah teapot constructed with four different dual tiles. Figure 10 shows two Utah teapots. For

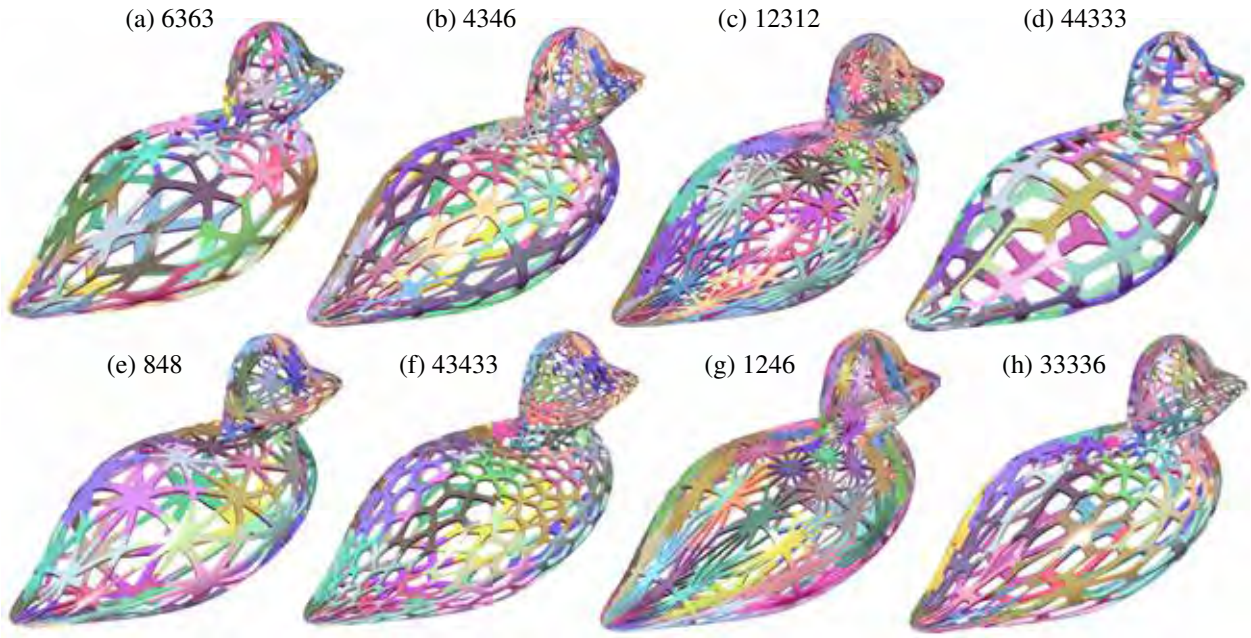


Figure 6: Different dual tiles composed into a duck surface.

each of the teapots, different parts of its body are composed with different dual tiles. In Figure 10 (a), the body of teapot is constructed using dual tile 44333, handle with dual tile 6363, spout with dual tile 43433 and lid with dual tile 33336. In 10 (b) dual tile 44333 used for body in 10 (a) is changed by the dual tile 4346. The proposed approach can be extended to shells with non-uniform thickness. Figure 11 demonstrates a tubular trivariate deformation map with a non-uniform thickness, shown in (a) and the composed non-uniform shell-lattice structure, composed with dual tiles 6363, shown in (b).

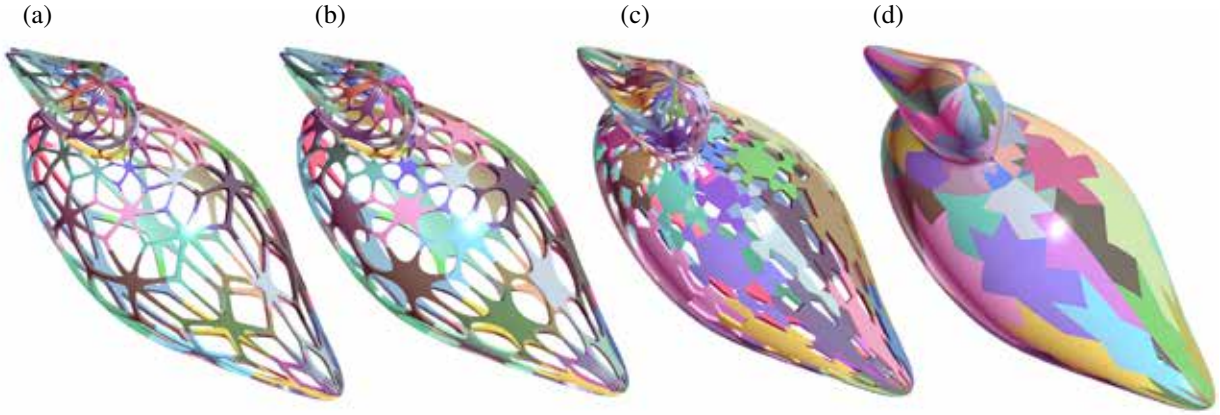


Figure 7: Tile 6363 (in (a) and (b)) and Tile 33336 (in (c) and (d)) composed into the duck surface for different values of D_{frac} and D_{curv}. (a) D_{frac} = 0.1 and D_{curv} = 0.9, (b) D_{frac} = 0.2 and D_{curv} = 0.5, (c) D_{frac} = 0.2 and D_{curv} = 0.1 and (d) D_{frac} = 1.0 and D_{curv} = 0.9

As stated in Section 3.2, we can construct trivariate dual tiles (recall Figure 4), and compose them into a trivariate deformation function to yield a trivariate volumetric representation of the shell structure. These trivariates can then be fed directly to analysis, using iso-geometric analysis [2]. Figure 12 shows one such example of a trivariate vase model composed with trivariate dual tilings based on 44333. Also shown in Figure 12 are hexa (cuboid) finite elements, that were sampled from these resulting trivariates, specifically in the Gmsh file format, and presented in the finite element software of Gmsh [6].

Finally, we present a pair of 3D printed model using geometry synthesized by the presented algorithms, in Figure 13. These models were tessellated into polygons and 3D printed on a J55 3D printer of Stratasys.

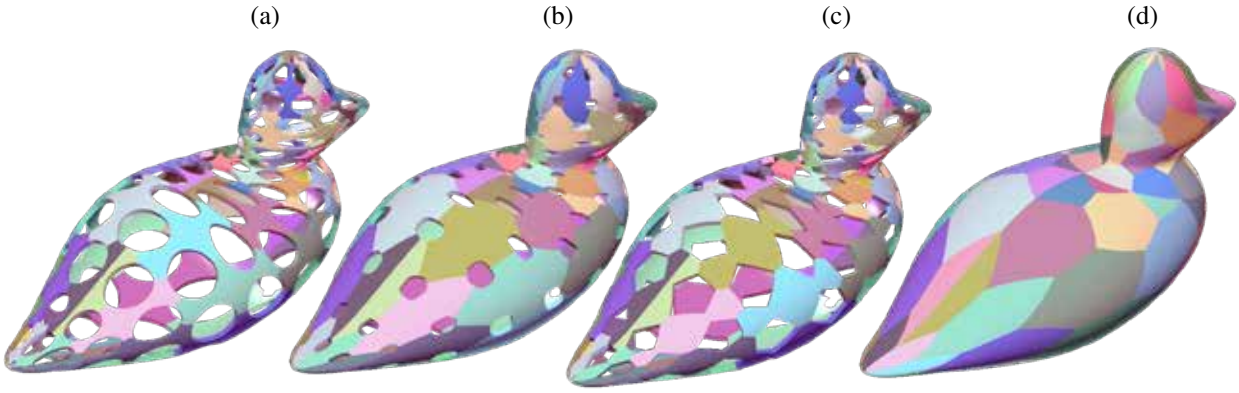


Figure 8: Tile 44333 shown in 5 (iv) composed into the duck surface for different values of D_{frac} and D_{curv}. (a) D_{frac} = 0.2 and D_{curv} = 0.4, (b) D_{frac} = 0.6 and D_{curv} = 0.6, (c) D_{frac} = 0.2 and D_{curv} = 0.0 and (d) D_{frac} = 1.0 and D_{curv} = 0.8

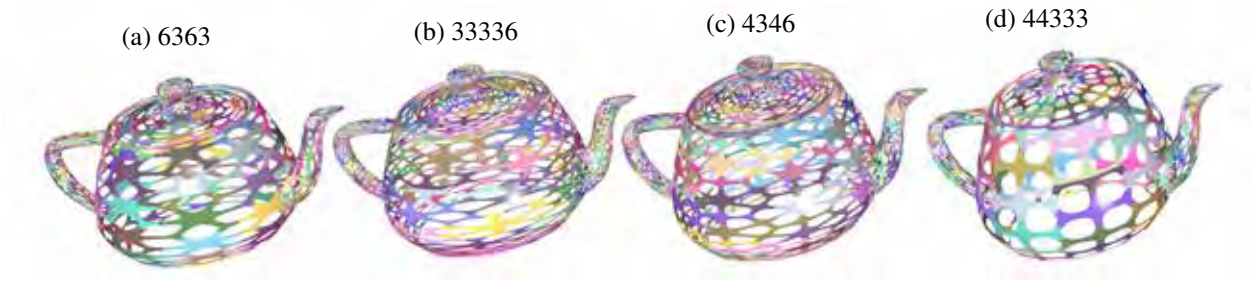


Figure 9: Different dual tiles composed into the Utah teapot surface. Dual tiles (a) 6363, (b) 33336, (c) 4346 (d) 44333 are used.

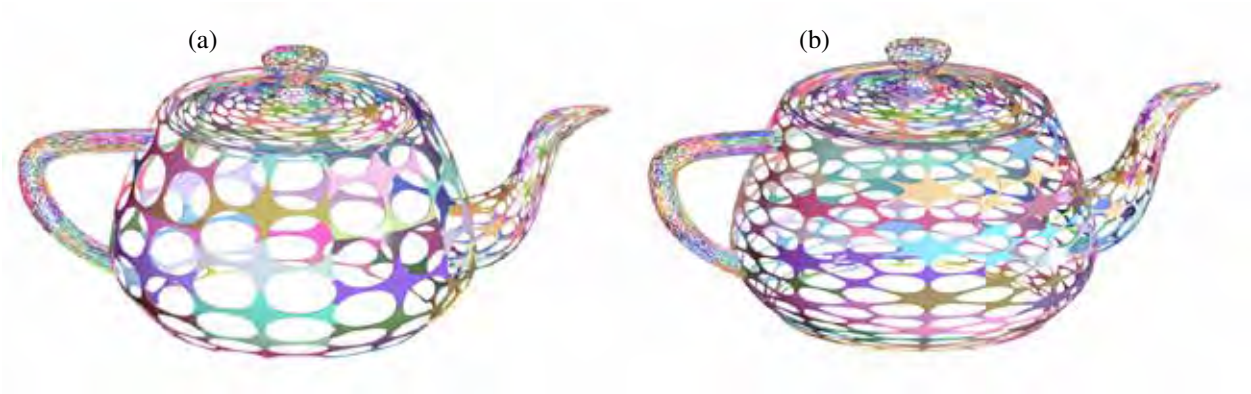
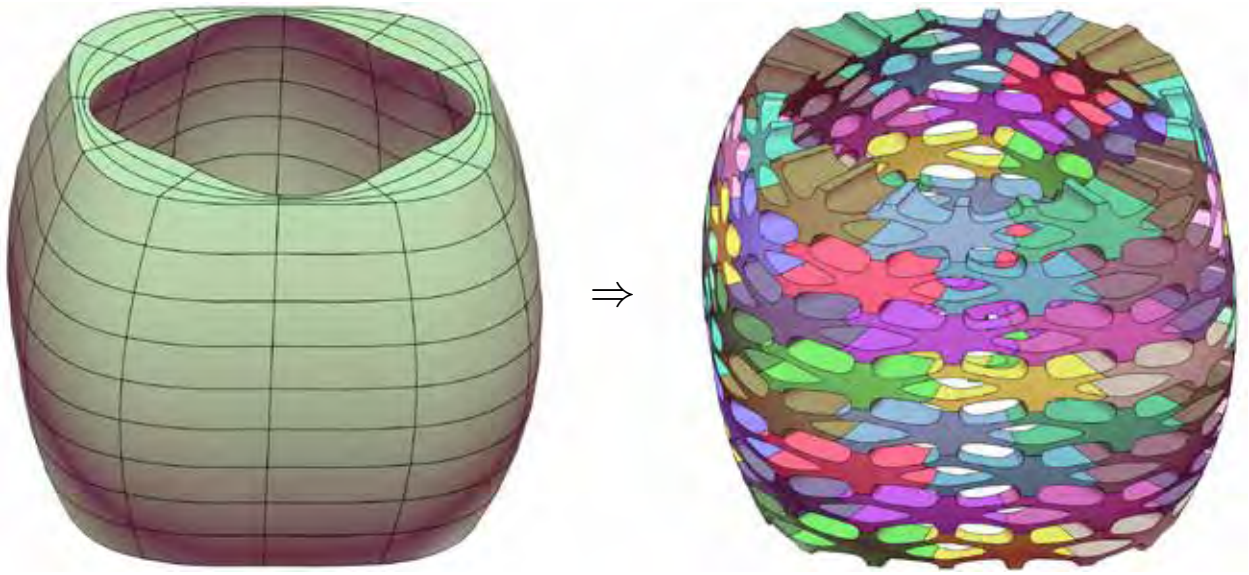
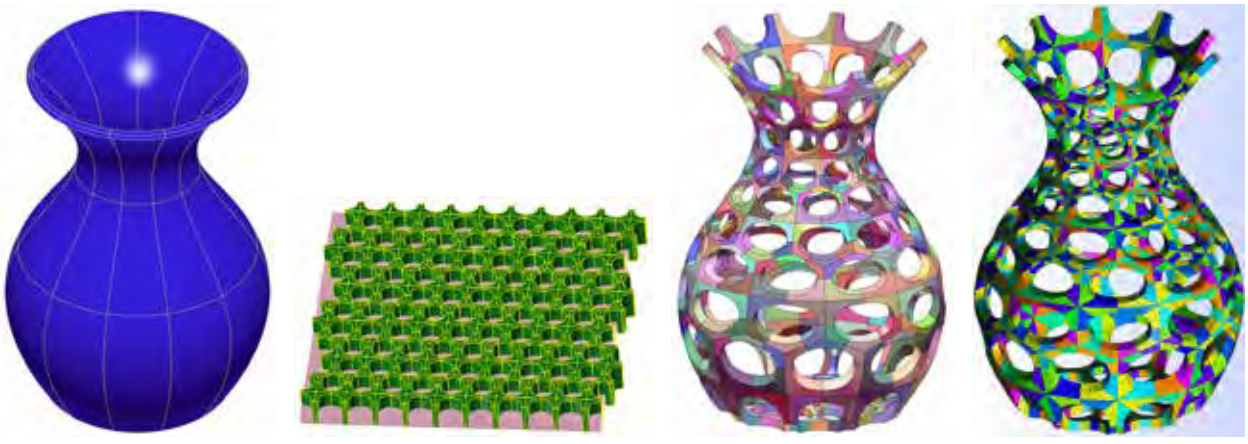


Figure 10: The different surfaces in the Utah teapot are composed with different tiles. In (a) body is made with dual tile 44333, handle with dual tile 6363, spout with dual tile 43433 and lid with dual tile 33336. In (b) body is changed to dual tile 4346.



(a) (b)

Figure 11: (a) A tubular trivariate with non-uniform thickness and (b) shell-lattice structure of non-uniform thickness, composed with dual tile 6363, that result.



(a) (b) (c) (d)

Figure 12: A vase model in (a), is used to create a composition of tilings of a trivariate tile 44333 into it. (b) shows the $[0, 1]^2$ 2D tiling, extruded into trivariate tiles. In (c), the composed trivariates are shown in 3D. Finally, in (d), a finite element mesh, formed by a direct sampling of (c) into hexa (cuboid) finite elements is shown, via the Gmsh finite element software.

Figure 13: 3D printed samples. The duck is similar to the model in Figure 6 (a). The vase is the same geometry as in Figure 12. Printed on a J55 3D printer of Stratasys.

5 Conclusions and Future Work

While 2D tessellations and space filling shapes are relatively well-understood, problems related to 3D tessellations and space filling shapes are least exploited and potentially have applications in a wide range of areas including chemistry and biology to engineering and architecture. Possible future work includes:

1. Development of semi regular tessellations that are based on 3D space filling tilings and extend to micro-structure construction scheme with trivariate tiles, allowing the encoding of volumetric characteristics leading to heterogeneity.
2. Study and analysis of the physical behavior of the lattice structures built using various different patterns [20, 21].
3. Correlate between specific dual (semi-) regular lattices and desired physical properties while possibly fulfilling aesthetic criterion.
4. Support of trimmed surfaces. Here, the different dual tiles will be required to be clipped against the trimmed curves and stitched to neighboring tiles on the other adjacent surfaces, along these trimming curves.

6 Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7 Acknowledgements

This research was supported in part by the ISRAEL SCIENCE FOUNDATION (grant No. 597/18) and in part with funding from the Defense Advanced Research Projects Agency (DARPA), under contract HR0011-17-2-0028. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] CHEN, L., FAN, H., SUN, F., ZHAO, L., AND FANG, D. Improved manufacturing method and mechanical performances of carbon fiber reinforced lattice-core sandwich cylinder. *Thin-Walled Structures 68* (2013), 75–84.

[2] COTTRELL, J. A., HUGHES, T. J., AND BAZILEVS, Y. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[3] DUMAS, J., LU, A., LEFEBVRE, S., WU, J., AND DICK, C. By-example synthesis of structurally sound patterns. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–12.

[4] ELBER, G. Precise construction of micro-structures and porous geometry via functional composition. In *International conference on mathematical methods for curves and surfaces* (2016), Springer, pp. 108–125.

[5] ELBER, G. The irit modeling environment, version 12. http://www.cs.technion.ac.il/~irit, 2021.

[6] GMSH. The gmsh finite element software. https://gmsh.info, 2021.

[7] HELOU, M., AND KARA, S. Design, analysis and manufacturing of lattice structures: an overview. *International Journal of Computer Integrated Manufacturing 31*, 3 (2018), 243–261.

[8] KAPLAN, C. S. Voronoi diagrams and ornamental design. In *Proceedings of the First Annual Symposium of the International Society for the Arts, Mathematics, and Architecture (ISAMA'99)* (1999), pp. 277–283.

[9] KAPLAN, C. S. Introductory tiling theory for computer graphics. *Synthesis Lectures on Computer Graphics and Animation 4*, 1 (2009), 1–113.

[10] MASSARWI, F., MACHCHHAR, J., ANTOLIN, P., AND ELBER, G. Hierarchical, random and bifurcation tiling with heterogeneity in micro-structures construction via functional composition. *Computer-Aided Design 102* (2018), 148–159.

[11] PAUFLER, P., GRÜNBAUM, B., AND SHEPHARD, G. Tilings and patterns. wh freeman and co. ltd., oxford 1987.

[12] RAO, M. Exhaustive search of convex pentagons which tile the plane,(2017), 2017.

[13] SCHUMACHER, C., MARSCHNER, S., GROSS, M., AND THOMASZEWSKI, B. Mechanical characterization of structured sheet materials. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–15.

[14] SUBRAMANIAN, S. G., ENG, M., KRISHNAMURTHY, V. R., AND AKLEMAN, E. Delaunay lofts: A biologically inspired approach for modeling space filling modular structures. *Computers & Graphics 82* (2019), 73–83.

[15] TRAUTZ, M., AND HERKRATH, R. The application of folded plate principles on spatial structures with regular, irregular and free-form geometries. In *Symposium of the International Association for Shell and Spatial Structures (50th. 2009. Valencia). Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures: Proceedings* (2009), Editorial Universitat Politècnica de València.

[16] WANG, Z.-W., HAN, Q.-F., NASH, D. H., AND LIU, P.-Q. Investigation on inconsistency of theoretical solution of thermal buckling critical temperature rise for cylindrical shell. *Thin-Walled Structures 119* (2017), 438–446.

[17] WEI, K., HE, R., CHENG, X., ZHANG, R., PEI, Y., AND FANG, D. Fabrication and mechanical properties of lightweight zro2 ceramic corrugated core sandwich panels. *Materials & Design 64* (2014), 91–95.

[18] WEI, L.-Y., LEFEBVRE, S., KWATRA, V., AND TURK, G. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR* (2009), Eurographics Association, pp. 93–117.

[19] WEIZMANN, M., AMIR, O., AND GROBMAN, Y. J. Structural performance of semi-regular topological interlocking assemblies. In *Proceedings of the Symposium on Simulation for Architecture and Urban Design* (2019), pp. 1–7.

[20] ZHANG, X., FANG, G., DAI, C., VERLINDEN, J., WU, J., WHITING, E., AND WANG, C. C. Thermal-comfort design of personalized casts. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (2017), pp. 243–254.

[21] ZHANG, X., LE, X., WU, Z., WHITING, E., AND WANG, C. C. Data-driven bending elasticity design by shell thickness. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 157–166.

[22] ZHAO, Y., CHEN, M., YANG, F., ZHANG, L., AND FANG, D. Optimal design of hierarchical grid-stiffened cylindrical shell structures based on linear buckling and nonlinear collapse analyses. *Thin-Walled Structures 119* (2017), 315–323.

3D Gosper Sculptures

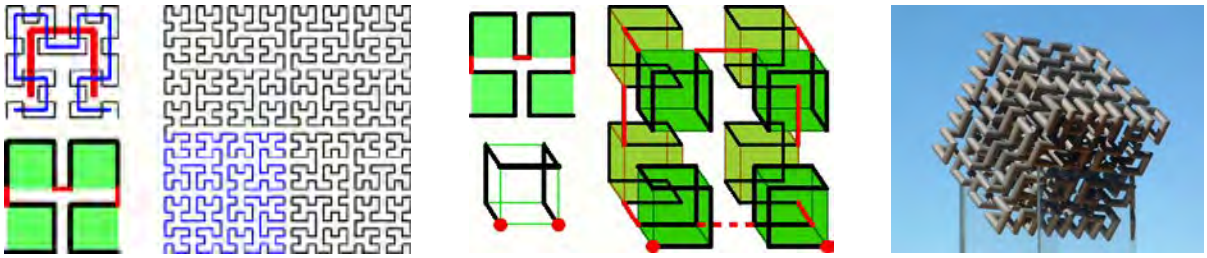
Carlo H. Séquin
CS Division, University of California, Berkeley
sequin@berkeley.edu

Abstract

Twenty years ago, I presented a recursive procedure to fill 3D Euclidean space with a single path inspired by the 2D Hilbert curve. A finite portion of this path then made an attractive constructivist sculpture. Now I am trying to do corresponding constructions inspired by the 2D Gosper curve. The goal is to create a modular polyline that connects nearest neighbor vertices in the face-centered cubic (FCC) lattice, while favoring the characteristic turning angles of 60 and 120 degrees. A complete recursive solution may not exist; thus the focus is on finite-size, modular sculptures. Promising looking designs are being realized as small sculptural maquettes on different 3D printers.

1. Introduction

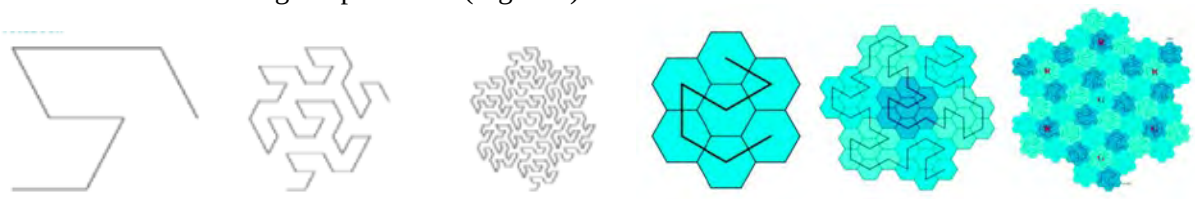
When designing abstract geometrical sculptures, it is often productive to look for symmetries and modularity that permit multiple re-uses of particular shape elements, possibly in different configurations. Suitable elements can often be found in 2D patterns or in plane-filling curves. This is how, in 2001, I came up with "Hilbert Cube 512" (Fig.1c) [2]. This was a third-generation recursive Hilbert curve in 3D space, developed based on a Hamiltonian circuit on the edges of a cube. Each vertex in this path was then replaced with a scaled-down version of an open Hamiltonian path on the cube. These cubelets were oriented in a way to maximize symmetry and to form an overall closed path that roughly follows the initial Hamiltonian circuit (Fig.1b).



(a) (b) (c)

Figure 1: *(a) 2D Hilbert curve; (b) recursive 3D Hilbert curve; (c) "Hilbert Cube 512."*

In this paper I study how another intriguing 2D space-filling curve, the *Gosper* curve (Fig.2) [6][9][10], may lead to a 3D space-filling curve or to finite-size sculptures that preserve the characteristic 60° and 120° turns of this curve. One difficulty is that there is more than one recursive construction to create a Gosper curve. The original presentation of the Gosper curve is based on a prototypical poly line, composed of seven unit-length segments. Each segment in this *proto-line*, is then replaced by a scaled-down version of the original proto-line (Fig.2a-c).



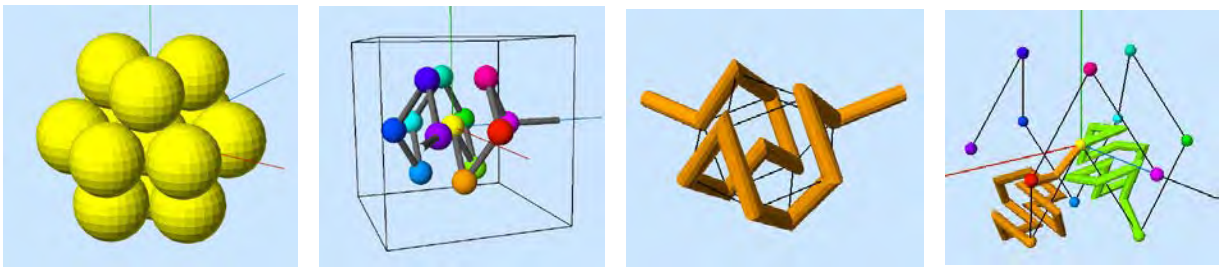
(a) (b) (c) (d) (e) (f)

Figure 2: *Two different Gosper curve constructions: (a-c) based on a 7-segment polyline; (d-f) based on seven hex-tiles [6].*

Another construction is based on a hexagon tiling, in which the proto-line connects the centers of adjacent hexagons [6]. With every recursive generation, a single hexagon is replaced with a clump of seven scaled-down hexagons. This is called the *node-Gosper* curve (Fig.2d-f). Both methods lead to similar-looking recursive polylines with characteristic turning angles of 60° and 120°; they can be drawn by selectively connecting adjacent vertices in a triangular lattice.

2. Basic Considerations and a First "Gosper Sphere"

To take the Gosper curve into 3D, a plausible first approach is to replace the roughly circular, compact clumps of vertices that define the original Gosper proto-line with a Hamiltonian path on a roughly *spherical* clump of equally spaced vertices. This is equivalent to replacing the basic square unit of the 2D Hilbert curve (Fig.1a) with a cubical unit (Fig.1b). For the Hilbert curve, an infinite recursive application of the same basic substitution step can lead to a curve that homogenously fills all of 3D Euclidean space. To construct a recursive 3D Gosper curve, a natural choice for a roughly spherical *proto-clump* might be the densest packing of 13 spheres (Fig.3a), in which the spheres are placed at the corners and at the center of a cuboctahedron. Figure 3b shows a Hamiltonian path on those 13 vertices, where all the bending angles are either 60° or 120°. This then constitutes a possible 3D *proto-line* for a 3D Gosper curve. Figure 3c shows a different, more sculptural view of the same proto-line.



(a) (b) (c) (d)

Figure 3: *(a) 13-atom proto-clump; (b) a Hamiltonian path in this clump; (c) corresponding prototype polyline; (d) two segments of the proto-line have been replaced with reduced proto-line copies.*

Line-Segment Substitution

The question now arises, how to best carry out the recursion step. Following the first approach (Fig.2a-c), every segment of the proto-line is replaced with a scaled down version of the path shown in Figure 3c. But this raises some problems. In the 2D case, there are four ways that a reduced proto-line can be placed onto the line segment that it replaces; in can optionally be mirrored, and it can also be rotated through 180°, which changes the direction in which one could pass through the proto-line. In 3D space, there are infinitely many ways in which the scaled-down copy can be inserted between two given points (Fig.3d). It can be rotated freely around the axis passing through the two endpoints. In addition, one can flip the direction of the inserted proto-line.

On the other hand, there are some constraints. The 13 small proto-lines must not interfere with one another, and, at the vertices where they join, they should create one of the desirable bending angles of 60° or 120°. It also is important to remember that all the vertices of the final 3D Gosper curve should fall on the sites of a regular lattice, and that the curve should densely populate all local lattice sites. For aesthetic reasons, we would like the construction to be as isotropic as possible. When starting with the 13-atom clump, these constraints reduce the choice of lattice to the face-centered cubic (FCC) lattice.

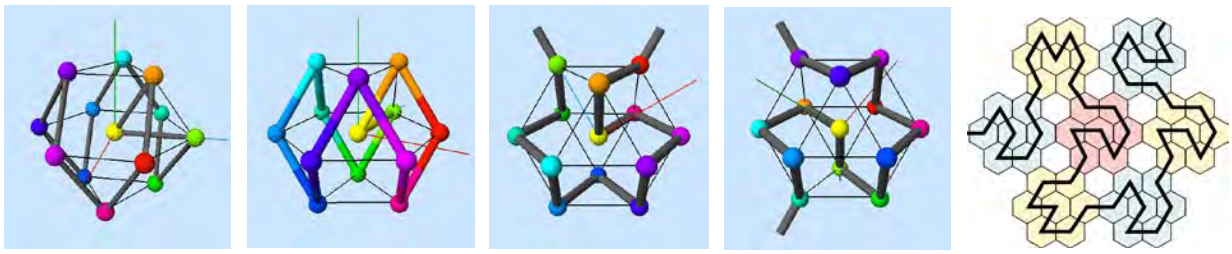
Node Replacement

Because it is difficult to figure out how to place all the proto-line copies so that all the above constraints are met, it seems easier to follow the node-replacement approach (Fig.2d-f). One starts by partitioning a suitable lattice into compact clumps that completely tile the lattice, – like the cube units in the 3D Hilbert curve (Fig.2b). For a 3D Gosper curve, the FCC lattice seems to be the natural choice. Every vertex has 12 nearest neighbours and thus offers 12 directions in which to continue the polyline curve, and the turning angles between subsequent neighbour-connections are either 0°, 60°, 90°, or 120°. Unfortunately, the natural clump of 13 atoms (Fig.3a) does not completely tile the FCC lattice.

In the fall of 2020, we had several Zoom meetings among the Bay-Area Artists & Mathematicians (BAAM), where we discussed ways to create completely-space-filling 3D Gosper curves. No nicely recursive method to create a homogeneous and isotropic path on the FCC lattice has emerged yet. A different proto-clump, and perhaps even a different 3D lattice, may be needed to yield a fully recursive space-filling curve; but there is a strong belief that a pure recursive solution may not exist [8].

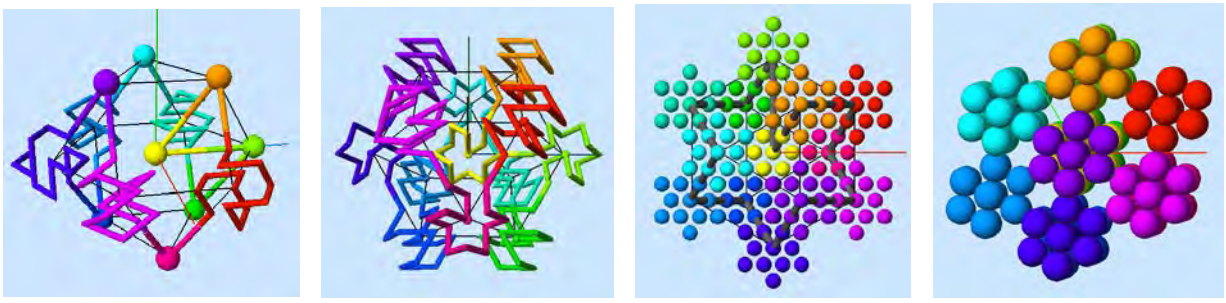
3. Finite Sculptures

In this paper, I thus focus on a different goal: to construct a finite-size, compact sculpture based on the Gosper curve. As in *Hilbert Cube 512*, I like to start with a *closed* Hamiltonian circuit on suitable starting clump (Fig.4a) to obtain a closed-loop Gosper curve, with no open ends sticking out. Such a closed-loop base-circuit may have angles of 60° at eight vertices and angles of 120° at five vertices (Figs.4b). I then place scaled-down copies of this clump at all 13 vertices. These replacement clumps are open-ended Hamiltonian paths, so that they can connect to two of their neighbouring clumps. Thus, two different replacement models are needed: one model with the two connection-stubs 60° apart (Figs.4c), and another one with 120° between connection points (Figs.4d). Both replacement paths can readily be realized with bending angles of 60° and 120°, so the recursive construction process can continue readily.



(a) (b) (c) (d) (e)

Figure 4: *Hamiltonian paths on 13-atom clump: (a) closed base-circuit; (b) highlighted bend-angles. Open paths with: (c) 60° between connectors, (d) 120° between connectors. (e) Hex-tiling with voids.*



(a) (b) (c) (d)

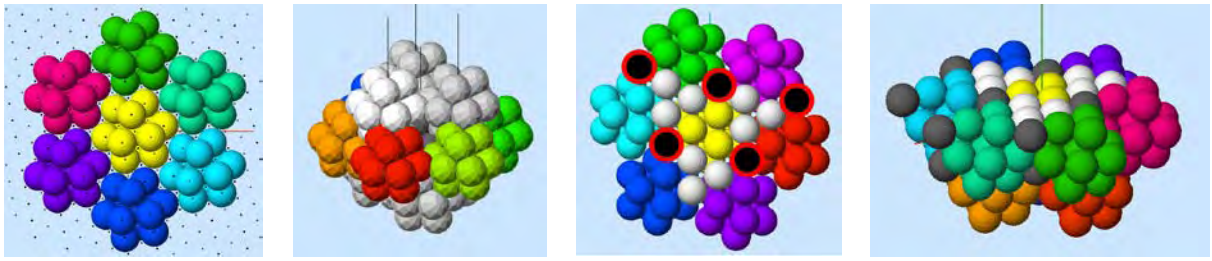
Figure 5: *Building a Gosper sculpture: (a) replacing the five 120° bends with clump Fig.4d; (b) replacing all bends with appropriate clumps; (c) clump connectivity marked in black; (d) view from another angle, revealing the voids between clumps.*

Stub-to-Stub Connections

With the replacement clumps suitably positioned and oriented to line up the stub-to-stub connections (Fig.5a,b), they will assemble into a single-loop Gosper-style path with the desired bending angles, and all the atoms will fall onto vertex sites of a shared FCC lattice (Fig.5c). On the other hand, not all lattice sites are actually visited by this sculptural path. Several sites between adjacent clumps are left unused (Fig.5d). Figure 4e shows an equivalent clumping and tiling in the 2D plane. These stub-to-stub connections render the 13 clumps quite visible and do not result in a dense space-filling path.

4. Looking for Denser Clumping

With the goal of obtaining a denser clumping in the envisioned sculpture, I followed more closely the packing of the hex-tiles in Figures 2d-f. I used the same arrangement to assemble seven spherical clumps so that the "interlocking" pattern of Figure 2e emerges in the equatorial plane of the assembly (Fig.6a). This approach places three atoms of every clump into the layers of spheres immediately above and below the equatorial layer. Additional clumps (white, grey) are now fit above and below the first seven clumps (Fig.6b); they each put an additional three atoms into these two layers. However, for each clump in the main layer, one lattice site in one of these adjacent layers remains unused. Thus, overall, one in 14 lattice sites still remains unoccupied (Fig.6c).

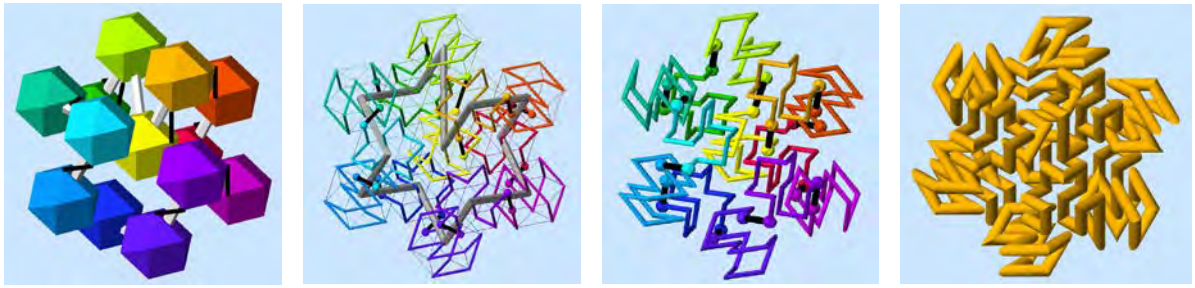


(a) (b) (c) (d)

Figure 6: *(a) Dense clumping of the 13-atom clumps in the equatorial plane (top view). (b) Three layers of 13-atom clumps (side view). (c) Void lattice sites (top view). (d) Voids filled with an extra (dark) atom added to each 13-atom clump.*

In a first approach to make a finite, compact, roughly spherical sculpture, I simply ignored this flaw. These isolated voids in lattice occupancy may not be noticeable by people who are not explicitly looking for them. Stub-to-stub connections cannot be used with this new clump packing. But, two neighboring clumps touch with more than one pair of atoms, and a connection can be established between any pair of touching atoms.

Figure 7a shows such connections as black rods between cuboctahedral clumps. White is the path of the initial base circuit that defines what clumps should be connected. My hope was that I could find a connection pattern, where the two connection points for each clump always are 120° degrees apart. This would then allow me to use just a single type of replacement clump, so that all 13 clumps would have the same internal connectivity and just needed to be rotated to match-up with two of the appropriately placed connector rods. However, after trying quite hard, I have not yet found such a connection pattern, and I am almost certain, that it does not exist. Thus, I made a special replacement clump for the central (yellow) unit, which has its connection points 90° apart. Moreover, because of the chirality of the replacement clump where the connection points are 120° apart, I also needed to make use of the mirrored version. Figures 7b and 7c show a realizable 3D Gosper path. Here one can check what bend-angles result at the ends of the black connector rods. It turns out, the angles are 0°, 60°, 90°, and 120°, and this nicely fits the Gosper paradigm. Figure 7d shows what this composite path would look like as a tubular sculpture.



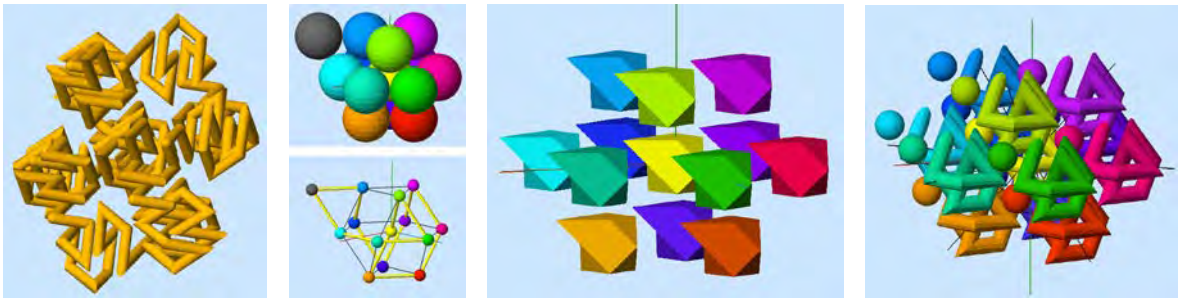
(a) (b) (c) (d)

Figure 7: *"Gosper Ball": (a) Connections between cuboctahedral clumps; (b) connected clumps around the initial base-circuit; (c) resulting 3D Gosper path; (d) tubular sculpture.*

5. Filling the Voids

From most viewing directions, the tubular sculpture depicted in Figure 7d looks reasonably compact and homogeneous. But from a few special directions, the same sculpture shows 13 distinct clumps with some voids between them (Fig.8a). Thus, it is desirable to fill those voids to obtain a more uniform fill-density through the core of the Gosper Ball.

All FCC lattice sites can be filled if we give the 13-sphere clump an additional atom (Figs.8b). Figure 8c shows how these 14-atom clumps might populate densely the FCC lattice sites. However, the extra bump produced by this atomic extension reduces the symmetry of the clumps and the isotropy of the overall structure. It also makes it impossible to rotate individual clumps to line up properly with any pre-defined connection points as exemplified in Figure 7b. This would then require individual designs of the Hamiltonian paths through all 13 clumps in the sculpture in order to obtain the required connections to neighboring clumps.



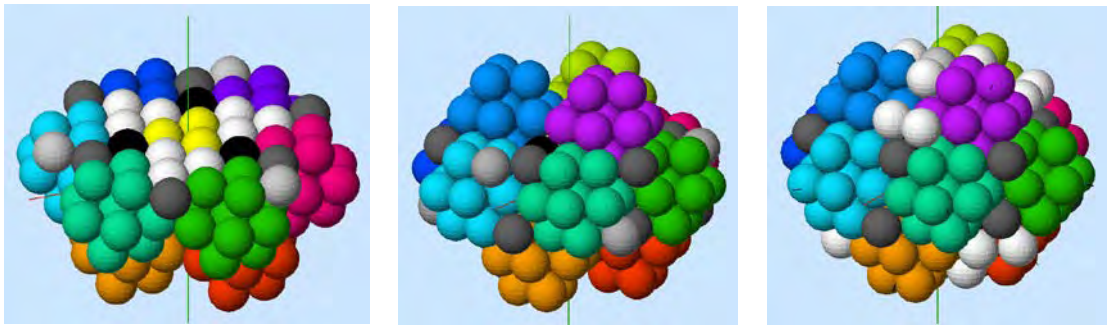
(a) (b) (c) (d)

Figure 8: *(a) Tubular sculpture revealing the voids. (b) Adding a 14th atom to the 13-atom clump. (c) 14-atom clumps on the FCC lattice. (d) The filler atoms in the context of the Gosper path.*

A more flexible approach adds isolated atoms onto all unoccupied FCC lattice sites. This allows one to not only fill the inner voids in the spherical 3D Gosper sculpture, but also to smooth out the rugged surface produced by the assembly of 13 roughly spherical clumps and to make the overall sculpture more spherical. Since there is no solid recursive construction that places the filler atoms automatically with the placement of the 13 clumps, the choice of where to place filler atoms is mostly an issue of aesthetics.

Figure 9a explores some options of which voids one might want to fill. The three black filler atoms touching the three yellow atoms of the central clump are needed to fill these completely internal voids. The six medium gray atoms are part of the surface of the sculpture and are more optional filler places. Finally, the three light grey atoms at the periphery are even more questionable: While filling some concave spaces, they also create additional new concavities next to themselves. Of course, whatever we do in this plane immediately above the equatorial plane, we should also do in the plane below the equatorial plane.

Figure 9b shows three layers of 13-atom clumps. I have maximally filled voids in both atom planes adjacent to the equatorial plane. Thus, these three planes of atoms look rather dense and homogeneous. But now the upper and lower ends of the sculpture, with just three naked clumps each, look rather loose, and the fact that one can still see one of the black filler atoms (Fig.9b) clearly indicates that there are some holes to be filled. Thus, I have added some filler atoms there as well. Figure 9c shows a complete *Gosper Ball* composed of 13 clumps and a total of 38 filler-atoms. With some selective filling-in in all layers, I was able to generate a reasonably smooth surface. – However, one may ask, is this the proper thing to do? Perhaps the raggedness of the Gosper surface is part of the essence of the Gosper curve. Any finite part of a 2D Gosper curve also occupies a "Gosper-island" with a ragged outline. Yet, in 3D, the raggedness of the surface without any filler atoms seems much more pronounced.

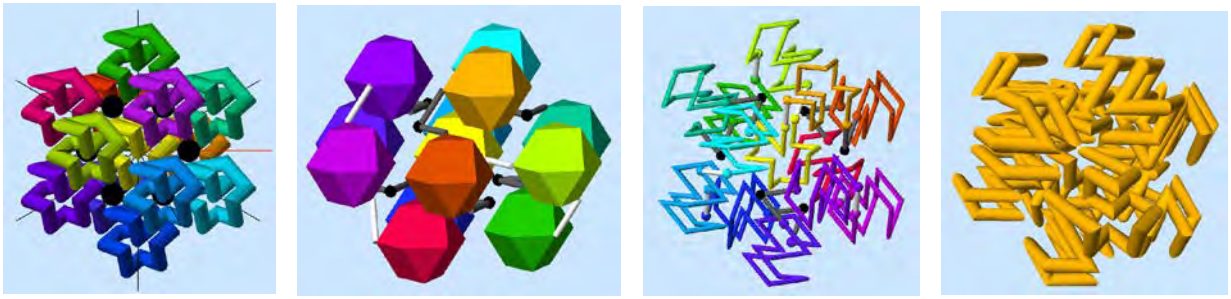


(a) (b) (c)

Figure 9: *(a) Different void filler options. (b) Both near-equatorial planes filled densely. (c) Also filling the voids between the upper and lower three clumps (white atoms).*

Deciding what lattice sites to populate with filler atoms is only half the problem. Next, for all these extra atoms, one must decide how to incorporate them into the final 3D Gosper path. Figure 10a shows the six innermost, required filler atoms in the context of the original 3D Gosper path (Fig.7). Should these atoms be integrated as a local detour into one of the nearby clumps? (Is this always possible without breaking the Gosper characteristics of the local path through that clump?) – Or is it better to use these atoms to transition from one clump to another neighbor? (Does this then require that every clump path must be modified in a different manner to accommodate these new entry- and exit-points?)

Figure 10b shows one possibility to link the six black filler atoms (Fig.10a) to two nearby clumps that need to be connected according to the chosen base-circuit. Six of the required clump-to-clump connections are made by such 2-step links, shown in grey (Fig.10b), while the remaining seven connections are made by direct rod connections, shown in white. Again, I was able to find a solution where the 12 outer clumps have their connection points 120° apart. Yet again, the central (yellow) clump is different; but now it has its connection points only 60° apart. Figure 10c displays the resulting Gosper path, and Figure 10d shows that path rendered as a tubular sculpture.



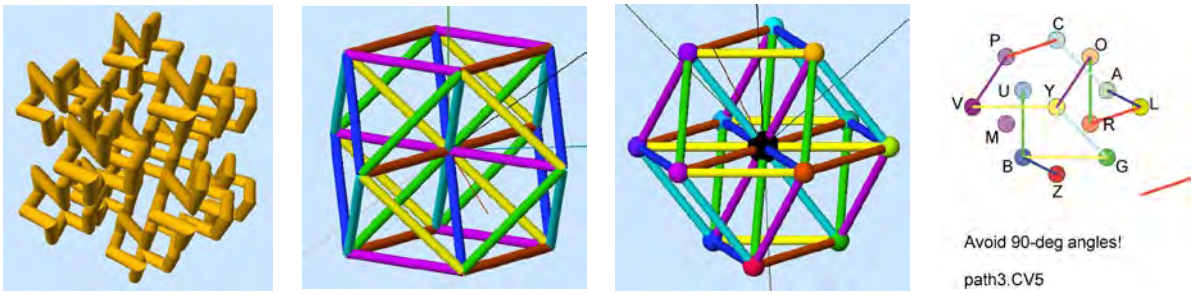
(a) (b) (c) (d)

Figure 10: *(a) The six crucial filler atoms (black) in the core; (b) using them as transition steps; (c) a resulting Gosper path; (d) corresponding tubular sculpture.*

Evaluation

Much tedious handy-work has been spent to make all the connections between the individual clumps. Was it worth the effort? The core of the sculpture definitely looks more compact; but the outer surface of the sculpture looks more "ragged." The 3D sculpture is missing the regular, structured look of the 2D Gosper curve. Also, the connections between clumps have led to some collinear consecutive segments. In some projections of the model, this seems to imply that the tube segments are of varying length.

More experiments with a different base-circuit and with different paths through the 13 clumps and the six filler-atoms need to be explored. A different approach might start with 13 identical clumps with the same closed Hamiltonian circuit on them. All are placed with the same orientation (Fig.13a). Then we break the 13 loops in different ways to make connections to neighboring clumps. But, with all clumps oriented in the same way, it becomes crucial to pick a loop that is as isotropic as possible. On the FCC lattice, tube segments can point in six different directions (Fig.11b,c). Ideally, the loop uses all six directions about equally often. Figure 11d shows a drawing utility that I generated to make it easier to look for such isotropic solutions: Segments in different directions are shown in different colors. However, isotropy is difficult to achieve in a Hamiltonian circuit with only 13 segments. Moreover, it may be even more difficult to obtain a truly isotropic look, because we might also want the pointy 120°-angle-turns to point isotropically in all directions; – there are 24 possible orientations! (In 2D the problem is less pronounced: There are only three segment directions and six pointy angle orientations.)



(a) (b) (c) (d)

Figure 11: *(a) The look of 13 orientation-aligned clumps. (b,c) The six possible segment directions. (c) A drawing utility to find good, isotropic Hamiltonian paths.*

Given all the various difficulties addressed above, perhaps starting with 13 clumps assembled in densest sphere-packing manner may not be the best way to make a "spherical" Gosper sculpture. Before discussing other possible approaches, I want to discuss some additional important, high-level concerns that should be considered before attempting to construct such sculptures.

6. Size, Complexity, Stability …

When choosing the sizes of the base-circuit and of the replacement clumps, overall complexity must be kept in bounds. First, there is an aesthetic concern: At too high a level of complexity, the tube pattern seen on the surface of a sculpture may look more like a texture on the overall (spherical) shape of the sculpture, rather than like a 3D constructivist tubular sculpture.

For a physical sculpture, there is also the issue of stability. My 6-inch model of a third generation *Hilbert Cube 512* (Fig.1c) is quite flexible, even though it is made of metal! If this 512-segment polyline circuit were stretched out, it would form a loop that stretches out for about 600 tube diameters before it turns back to the origin. With the next generation it would be going 8 times farther. Gravity may cause visible bending and warping on such a long loop without any intermediate supports.

The same limitations apply to a Gosper sculpture. At practical scales, from desk-top models to human-scale sculptures, a single-loop tubular sculpture is probably restricted to a few hundred vertices. This limits the overall design to a maximum of two hierarchical levels. In a 3-level sculpture, the

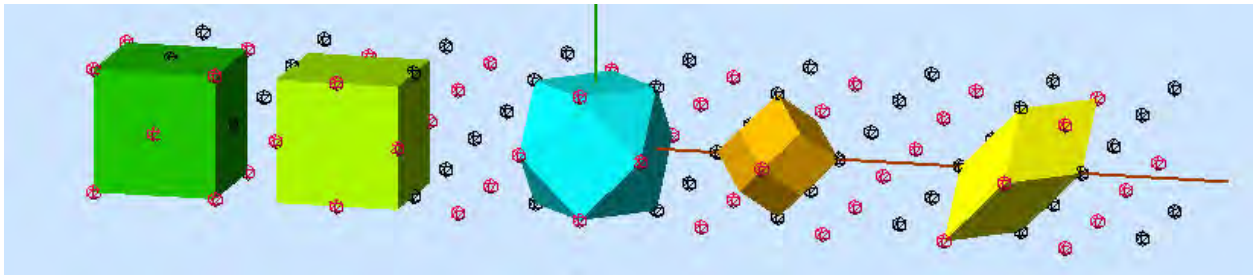
individual clumps at each level would have to consist of less than 10 atoms, and this would not be sufficient to capture the Gosper character within each clump. Thus, the 2-generation designs described above, which are composed of 13 clumps of 13 or 14 atoms, are an appropriate start.

The best way to increase complexity by a limited amount is to add more complexity into the 13 replacement clumps. Starting with a base-circuit that has more than 13 segments seems less important. The base-circuit mostly defines the overall shape of the sculpture. The connectivity between the replacement clumps is not so important, since in a busy sculpture it will be hard to discern how the clumps are connected to form the basic loop. On the other hand, the path through the individual clumps primarily defines the characteristic of the displayed space curve, and it will determine whether the sculpture looks like a 3D Gosper path.

7. Modeling Issues

I ended up doing much of the detailed modeling work twice, with different orientations of the coordinate system. I had started out describing the densest clump of 13 spheres by lining up one of its 3-fold rotational symmetry axes with the z-axis of the modeling system. This led to a pleasing projection of this clump that made it easy to obtain 3-fold rotational symmetry and to place seven spheres tightly packed into the equatorial x-y-plane. However, as the models became more complex, with filler atoms added, it got more difficult to figure out the exact placement of all the features and the mutual offsets between them, which were needed to add exact connecting branches between the various clumps.

Thus, at some point, I started again by constructing first an FCC lattice with all vertices on integer coordinate values. This ended up as a stack of simple uniform square grids parallel to the x-y-plane, alternatingly offset in the x- and y- directions by one unit as the z-height increased (Fig.12). Now all the vertices of the basic cuboctahedron and the placement of all the clumps and filler atoms ended up on integer coordinates, and maintaining precision was no longer a problem.

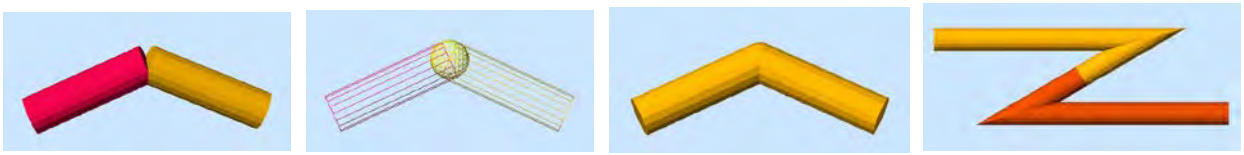


(a) (b) (c) (d) (e)

Figure 12: *The FCC lattice and some of its tiles: (a) cube; (b) cube shifted; (c) cuboctahedron; (d) rhombic dodecahedron; (e) rhomboid.*

Another modeling issue concerns the generation of good “water-tight” boundary representations in the STL files used for the fabrication of 3D-print models. For aesthetic reasons, I found it desirable to model the various Gosper path fragments as sweeps of a circular cross-section, with sharp, mitered corners at all the turns of the sweep path – as generated easily by the Berkeley SLIDE software [4].

It is impractical to model the complete Gosper path as one single sweep. To preserve some modularity, individual clump paths need to be stitched together. One solution is to close off individually all partial sweeps at both ends with *end-caps*. However, when two perpendicularly sealed-off sweeps are joined at an odd angle, the two boundary representations will partly overlap and also leave some wedge-shaped gap (Fig.13a). To correct for the missing material in the gap, a small sphere with the same diameter as the two cylinders is placed at the junction point (Fig.13b); this leads to a suitable geometry for the joint. This approach assumes that the slicing software of the 3D-printer will properly identify multiply-overlapping *internal* areas as simply *internal* [1]. Successful 3D-prints using this approach have been fabricated on Ultimaker printers [7] as well as on Stratasys machines [5].



(a) (b) (c) (d)

Figure 13: *(a) Two cylindrical elements joined at an arbitrary angle yielding overlap and a gap. (b) Adding a sphere to yield a seamless joint. (c) Resulting smooth joint. (d) Colinear joint.*

However, such nicely rounded bends may not be the preferred way to capture the look-and-feel of a Gosper curve. Properly mitered, sharp, spiky joints may be preferable at places where the path makes a 120° turn. In that case, it is better to join separate swept path fragments halfway between two FCC lattice sites, with the two tubular path segments joining in a lined-up collinear manner (Fig.13d). Representing cylinders as prisms with 30 or more sides typically is good enough to produce STL files with "water-tight" boundary representations at these junctions, even without using *end-caps*. If prisms with a much lower number of sides are preferred, I use the *twist*-parameter in the SLIDE sweep construct [4] to align the meshes of the two abutting tube cross-sections.

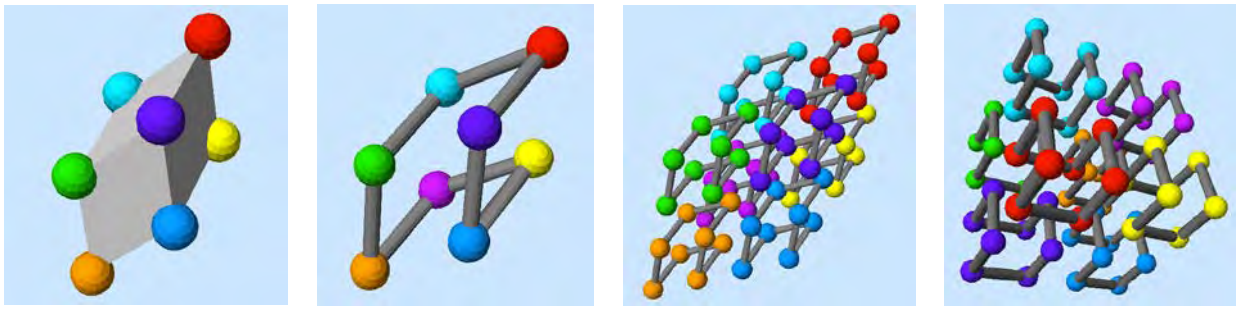
8. Partitioning the FCC Lattice

Much of the problems discussed in the first half of the paper are a result of the fact that the 13-atom clumps do not fit together snuggly in 3D space without any voids in between. On the other hand, the FCC lattice is my top choice, because every site has 12 nearest neighbors. This readily permits to construct Gosper-type paths with unit-length steps and with bending angles of 0°, 60°, 90°, or 120° between subsequent steps. Thus, I will explore next how the FCC lattice can be partitioned into 3D polyhedral tiles that fit together seamlessly.

Once the goal of finding a truly recursive definition of a Gosper path that fills all of 3D space had been pushed aside, I could focus on two separate issues that are important in the construction of a finite 3D sculpture: the definition of a suitable base circuit that defines the overall shape of the sculpture, and the construction of appropriate modular clump tiles.

Rhombic Tiles

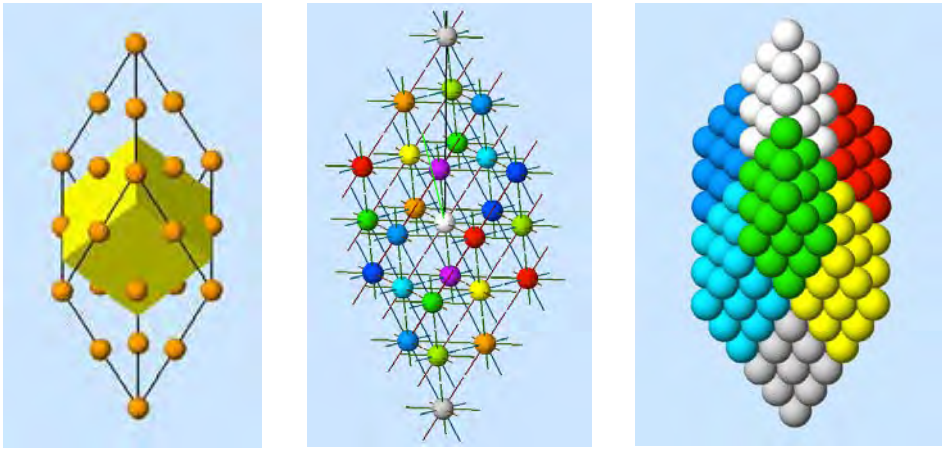
An affine distortion of a cubic lattice, that stretches the unit cube by a factor of 2.0 in the direction of one of its space diagonals (Fig.14a) produces a rhomboid lattice that also places all vertices at the sites of an FCC lattice. This lattice can thus readily be tiled with a rhomboid composed of just 8 atoms. This is just large enough to capture the key angles of 60 and 120 degrees of the Gosper curve. But if this prototype curve (Fig.14b) is used in a recursive manner, the resulting assembly looks more like a stretched 3D Hilbert curve (Figs.14c,d) rather than like a Gosper curve.



(a) (b) (c) (d)

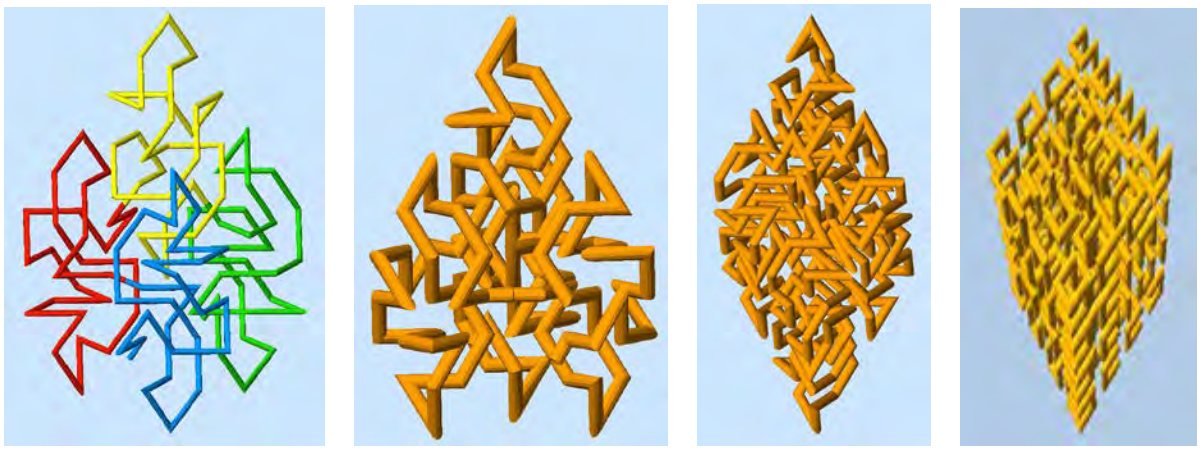
Figure 14: *(a) Stretched cube; (b) proto-path on 8 atoms; (c) an assembly of 8 such clumps; (d) a look at this assembly from the pointed end.*

A larger rhomboid tile composed of 27 lattice sites (Fig.15a,b) offers a richer variation in local path behavior and a better chance of capturing the nature of the Gosper curve. Indeed, an interesting sculpture results when eight 27-atom clumps are assembled (Fig.15c) and properly interconnected (Fig.16).



(a) (b) (c)

Figure 15: *(a) Diagonally stretched cube; (b) atoms placed on FCC lattice. (c) Eight 27-atom clumps.*



(a) (b) (c) (d)

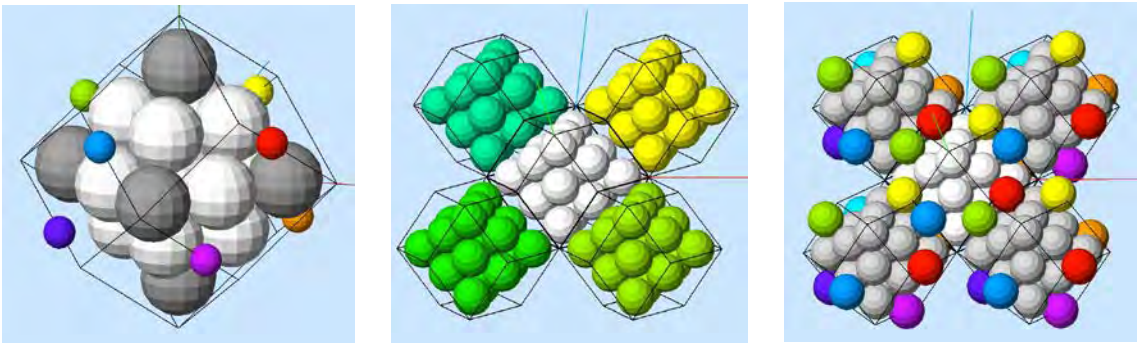
Figure 16: *Rhomboid Gosper sculpture: (a, b) path through the top four 27-atom clumps; (c) "Gosper-Pole_216" using 8 clumps. (d) Diagonally stretched Hilbert Cube.*

Figure 16a shows four suitable Hilbert paths on the 27-atom clumps that capture the spirit of the Gosper curve and that also properly interconnect adjacent clumps. Figure 16b shows this path in the form of a tubular sculpture. Combining two of these assemblies, results in the complete sculpture (Fig.16c), which has a notably different character from an affinely stretched *Hilbert Cube* (Fig.16d).

Rhombic Dodecahedron (RD)

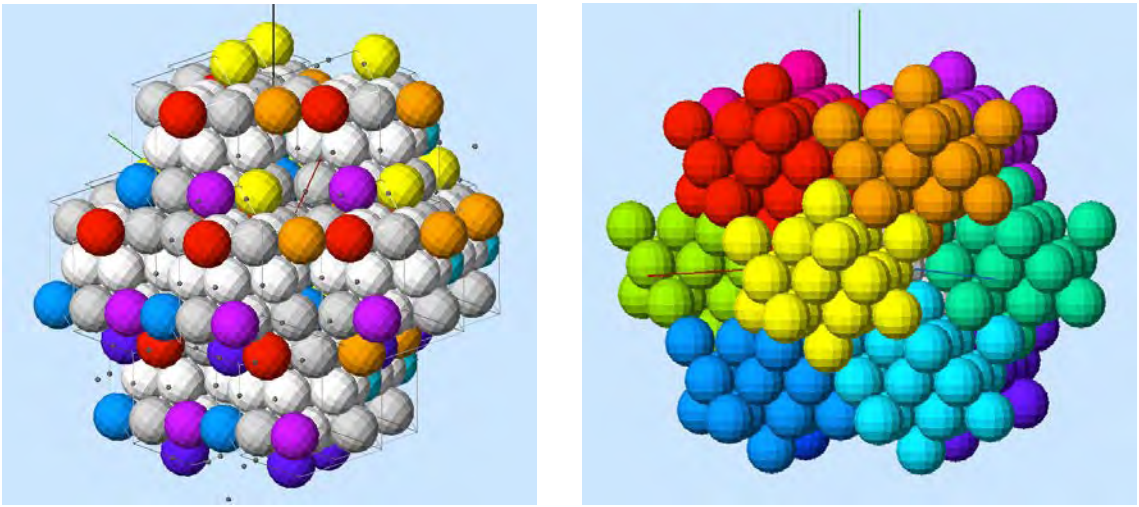
If our goal is to construct a more "spherical" and isotropic 3D Gosper sculpture, then we need to look for a more "spherical" tiling unit. A natural choice then is the *Rhombic Dodecahedron* (RD). To use this cell in the current context, atoms must *not* be placed at the 14 vertices of this polyhedron, but they should be placed *inside* this cell. In addition, where neighboring cells are abutted by using the faces of the RD, empty lattice sites must be filled with atoms that overlap with more than one RD cell.

One way to fill a RD-shaped cell is to add six additional atoms around the 13-atom clump used earlier in this paper, by placing them inside the six valence-4 vertices of the RD (grey atoms in Fig.17a). When trying to fill 3D space with this RD tile containing 19 atoms, not all sites of the FCC lattice are getting covered (Fig.17b). There is a site on every edge of the RD that is left unoccupied. However, an atom placed at such a site extends into the RD cells of the three neighboring tiles that share this edge. Thus, this atom must not be assigned to more than one of those three clumps, or multiple overlapping occupancies would result. In designing the basic tile clump, there are 24 edges on the RD where an atom might be placed; but the tile must only use *eight* of these atoms (colored atoms in Fig.17a). Those must be chosen judiciously, so that in a clump of 13 densely packed tiles no overlaps occur. Figure 17c shows how the interstitial sites are covered in the same 5-clump assembly (shown in Fig.17b). Wherever two rhombic faces connect between adjacent clumps, each clump contributes two shared atoms.



(a) (b) (c)

Figure 17: *The RD clump: (a) 19 atoms inside the RD shell, plus a subset of 8 atoms on the 24 edges; (b, c) an assembly of five RD clumps without and with the colored filler atoms.*

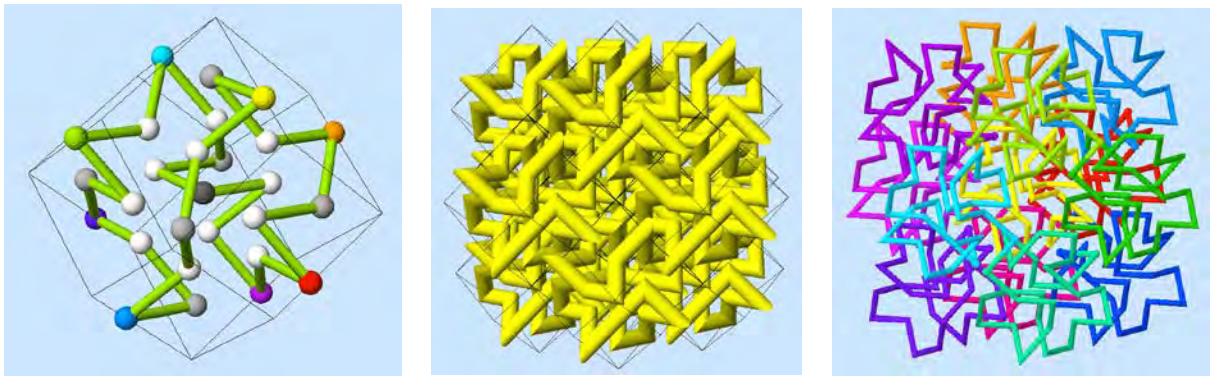


(a) (b)

Figure 18: *Thirteen 27-atom clumps: (a) "filler" atoms highlighted; (b) uniformly colored clumps.*

Figure 18 shows an assembly of 13 clumps, with the shared atom locations shown in the colors used in Figure 17 (Fig.18a), and with all 13 clumps in different colors (Fig.18b). The shared "filler" atoms fill the core of the sculpture densely. On the other hand, they produce a rather irregular surface structure. To obtain a smoother surface, some of these atoms might be selectively removed. However, this would destroy the nice modularity of the underlying RD tiling and would require different connectivity patterns in most clumps. It seems more appropriate to accept this irregular, corrugated surface; – after all, every finite portion of the 2D Gosper curve also occupies an island with a serrated fractal outline (Fig.2f).

Figure 19a shows a hand-crafted Gosper-like path connecting the 27 atoms in this RD cell. All bending angles are either 60° or 120°. In Figure 19b I have packed 13 such RD cells into a densest sphere-packing configuration to obtain an idea what a 3D Gosper sculpture with 351 path segments would look like. Even though the 13-atom base circuit has only two bend-angles of 60° and 120°, respectively (Fig.4a), I still needed to design several different open-ended clump paths, since the filler atoms make these clumps anisotropic and they no longer can be rotated freely into all the required orientations to make the proper connections to two neighbor clumps. The result is shown in Figure19c.

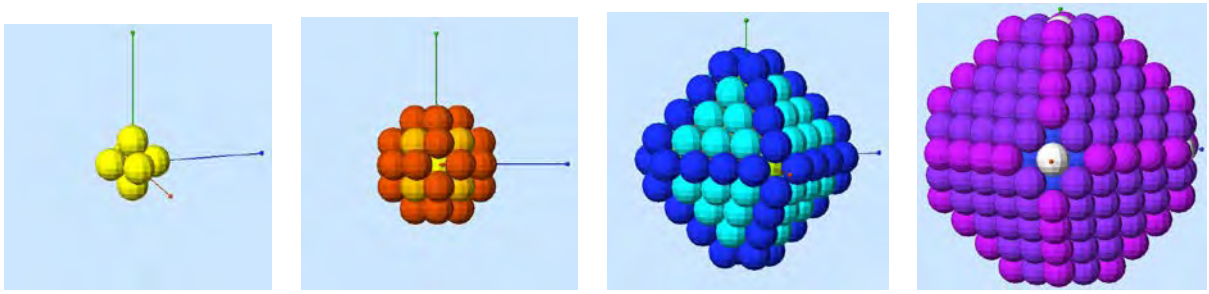


(a) (b) (c)

Figure 19: *Gosper path in RD-27: (a) Connecting the 27 atoms. (b) 13 clumps tightly packed together as a preview of a 351-node Gosper-Lattice. (c) The 13 clumps properly interconnected.*

9. Onion-Shell Approach

The difficult trade-off between the roundness of the clumps and the number of interstitial voids between them prompted me to investigate also a different approach. Here I am trying to form a nicely rounded overall shape as a sequence of nested shells of atoms. I start with an octahedral core composed of six atoms (Fig.20a). This core is surrounded with a second shell with 8 + 6*4 = 32 atoms (Fig.20b). The next shell comprises 6 (lime) atoms on the coordinate axes, 12*4 = 48 (blue) atoms on the edges of an octahedral shape, and 8*6 = 48 (cyan) atoms to fill in the faces, for a total of 102 atoms (Fig.20c). The forth shell has 6 + 48 + 120 = 174 atoms (Fig.20d).



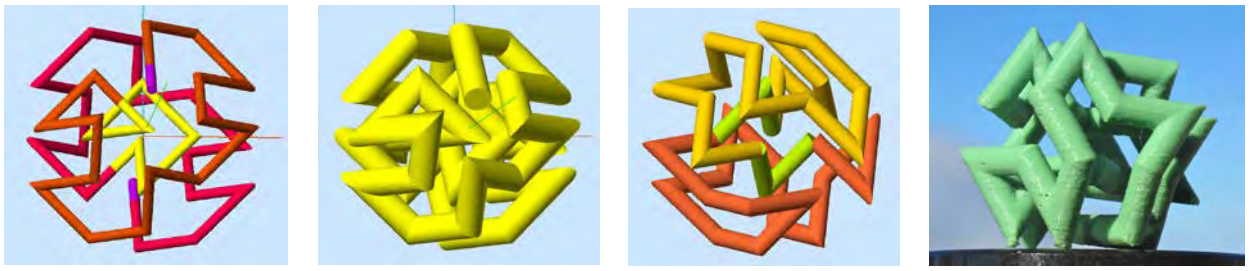
(a) (b) (c) (d)

Figure 20: *Onion shells: (a) central octahedral core, 6 atoms;. (b) 2nd shell, 32 atoms; (c) third shell, 102 atoms; (d) forth shell, 174 atoms.*

The main idea is to draw a bent version of a 2D Gosper pattern in each shell. Fortunately, the outermost shell, which is most important for conveying the Gosper-look, is also the one with the least curvature. Each shell connects with a minimum of two points to the shell below; these points correspond to the beginning and the end of the Gosper curve in the upper (outer) shell.

Figure 21a shows this approach for the first two shells. The Hamiltonian path on the octahedron is shown in yellow. At both ends, it transitions into the second shell and continues for another 15 segments

to cover half the atomic sites in that shell, exhibiting two 7-segment Gosper patterns (Fig.2d) on two of the octahedral faces visible in Figure 20b. It ends in the two purple stubs, through which it may connect to shell #3. Figures 21b shows this combined path as a tubular sculpture; one of the connection stubs to the third shell can be seen clearly.



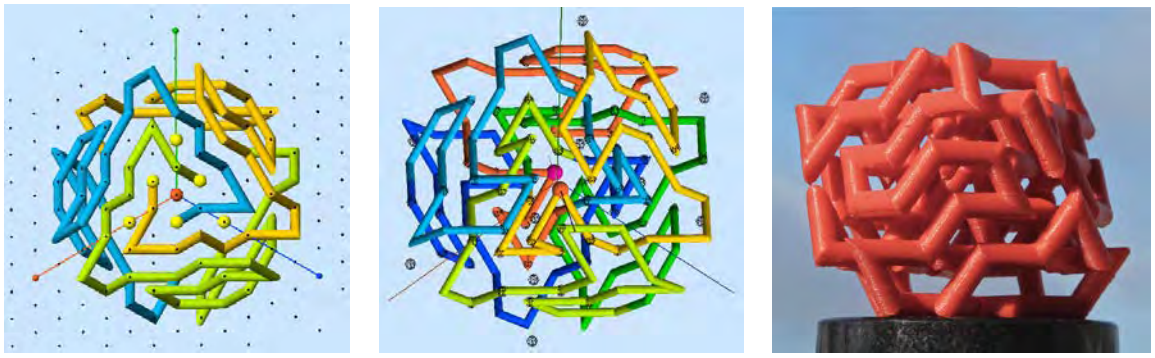
(a) (b) (c) (d)

Figure 21: *Onion shell paths: (a) Octahedral path (yellow), surrounded by two 15-segment paths in the 2nd shell, ending in the purple stubs that will connect to shell #3. (b) The combined path shown as a tubular sculpture. (c) A different path forming an overall closed circuit. (d) The closed path presented as a tubular "level-2 Onion" sculpture.*

The plan is to continue with the same spherically layered approach for a couple more shells. However, I am concerned about the stability issues discussed earlier in the paper. If a large outer shell is supported through only two points on the next lower shell, it may easily bend, shift, or sag so that it touches the inner shell in some manner. Thus, it seems advisable to partition that shell into two or three parts that are separately supported by the lower shell. This may also yield an opportunity to give this sculpture some overall symmetry and thus some modularity in design.

Figure 21c shows this approach for shell #2. The shell is partitioned into two identical (orange and red) paths that have their end-points 90° apart from one another. Two 2-segment (lime) paths on the octahedron vertices make two connections between the half-shell paths, thereby forming a single closed circuit (Fig.21c). Figure 21d shows the same path as a tubular sculpture. I think, this "*Gosper-Onion_38"* has some merits as a stand-alone mini-sculpture, and it looks like a promising start for onion sculptures with more shells.

Figure 22 shows my approach to designing a 3-level *Gosper-Onion*. I have partitioned the outermost shell into three identical (orange, blue, and green) parts, each covering roughly two octants of a sphere. Each path then connects at both ends to fragments of the second shell that each cover one octant of a sphere. The three composite paths then get joined in the "level-1 shell" by three edges of the central octahedron. So far, the structure has nice D_3-symmetry.



(a) (b) (c)

Figure 22: *A 3-level Gosper Onion: (a) Three path fragments arranged with D_3-symmetry. (b) Adding detours into the prototypical path fragment to visit all FCC lattice sites. (c) The complete "Gosper-Onion_140" as a sculptural maquette.*

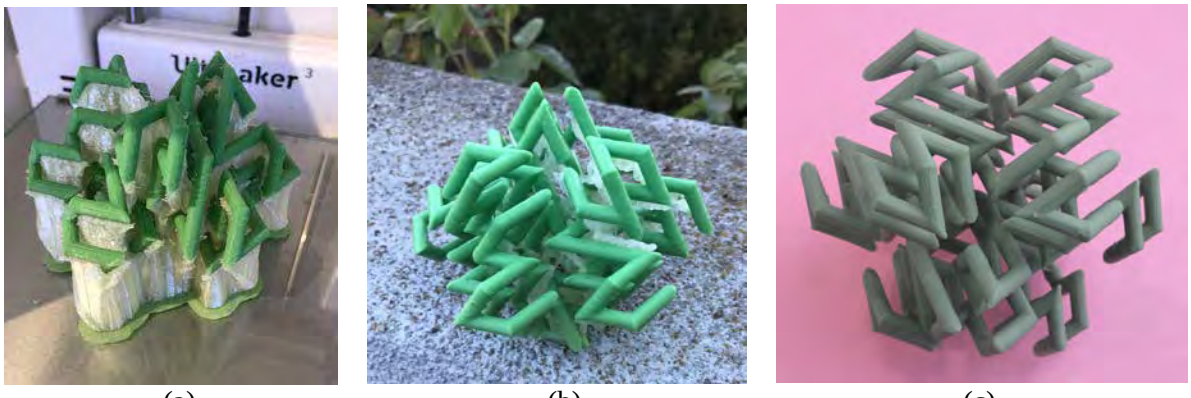
Unfortunately, not all FCC sites have been covered yet. In particular, the two opposing octants through which the D_3-axis passes have sites that still need to be included in the overall path. A site in shell #2 is shown in red, and six sites in shell #3 are shown in yellow in Figure 22a. The orange branch in shell #3 makes a detour to include two of these yellow atoms, and so do the other five copies of the orange half-path in Figure 22b. In addition, the orange branch also makes an extra step to include the vacant red atom on shell #2. This move is not included by the blue and green path copies; it thus breaks the D_3-symmetry. The complete level-3 Gosper-Onion path is shown in Figure 22c as a 3D print.

10. Fabrication of 3D-Print Models

The fabrication of good 3D print models of the above designs turned out more difficult than anticipated. The default setting in the CURA slicing program for the Ultimaker printers [7] is not suitable to provide good enough support for the various tubular elements slanted at different angles. In particular, any downward pointing, spiky, mitered joints need a much smaller filament overhang angle (about 30°) than the default of 60° (Fig.23a). This support-extension then results in a rather solid block of support, which is difficult to remove from the inner parts of any Gosper sculpture (Fig.23b), particularly, if the default *gridded* support structure is used (Fig.23c). After several initial trial runs on the Ultimaker printers readily available to me, it became clear that any fabrication technique that requires manual removal of the support structure is not suitable for Gosper models with more than about 50 path segments. The inner portions of such sculptures are difficult to reach, and the overall single-loop path is too fragile to allow pulling out inner support elements by brute force.

(a) (b) (c)

Figure 23: *Fabrication difficulties: (a,c) Onion shell path (Fig.21b); (b) Gosper-Pole_216 (Fig.16c).*



(a) (b) (c)

Figure 24: *Results on Ultimaker with soluble support: (a) fresh off the printer; (b) with most of the support material washed away; (c) after final manual cleanup. (Photos a,b by N. Weaver).*

After some experimentation with the advanced parameter settings in the CURA slicing software, good results have recently been obtained on an Ultimaker printer using PLA for the model and a water-soluble support material (PVA). A model corresponding to Figure 10d is shown as it came off the printer (Fig.24a), after most of the support has been washed away (Fig.24b), and after manual cleanup of the last remnants (Fig.24c).

I have also started to explore the use of an FDM printer from Stratasys [5] that has a soluble support material. Models that are more complicated will be sent off to the Shapeways 3D printing service [3] for nylon prints using Selective Laser Sintering.

11. Discussion

How do I judge whether I have composed a successful 3D Gosper Sculpture? I think that the following criteria can help answer this question: The result should be a closed polyline with segments of uniform length on a suitable 3D lattice, with the FCC lattice being the preferred choice. The polyline should exhibit the characteristic "claw" shapes seen in Figure 2, and it should favor bending angles of 60° and 120°. The local geometry should have some modularity and repeat the same local pattern several times, while completely filling all lattice sites in an appropriately shaped 3D island domain on the FCC lattice.

The last characteristic seems to be the most difficult one to achieve, because it is hard to find a good clump of atoms that tessellates the FCC lattice without any interstitial voids. Cubic clumps can readily tile the FCC lattice, but clumps with just $2^3 = 8$ atoms (at the corner of a small cube) are not enough to define a suitable 3D Gosper claw; and clumps with $4^3 = 64$ atoms emphasize the cubic shape too strongly to pass for a Gosper curve. A clump of $3^3 = 27$ atoms may be acceptable, particularly, if it is affinely deformed into a rhomboid shape (Fig.16) and if we accept an overall elongated shape for the resulting "Gosper-Pole" sculpture.

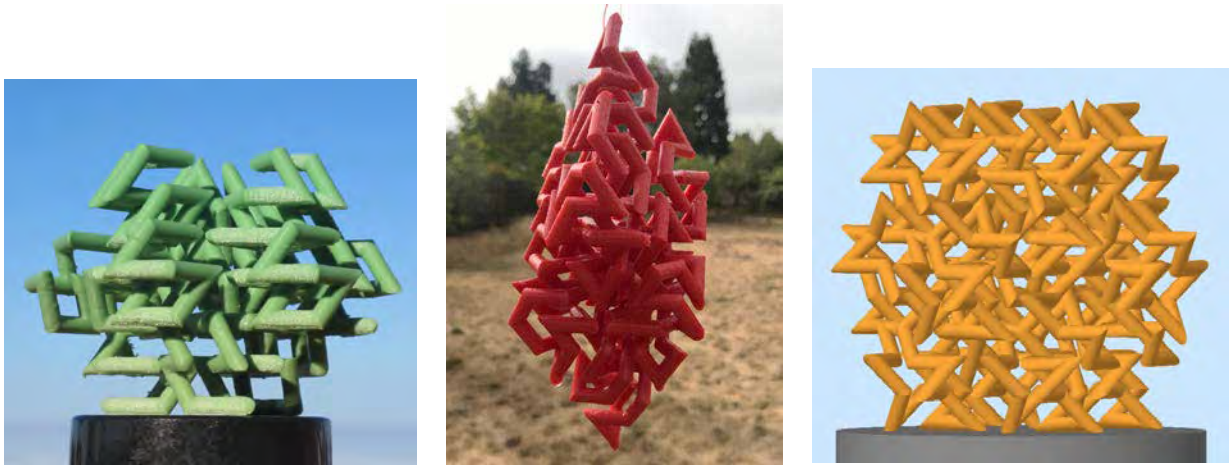
If we want to obtain a more "spherical" sculpture, as implied by the hexagonal patterns in Figure 2, we face a challenging trade-off. It is easier to find small clumps that seamlessly partition the FCC lattice, but they may not offer a rich enough set of path options to represent a convincing Gosper characteristic. As we look for larger and rounder clumps, they will leave more interstitial voids, which must be included in the overall polyline, – which appears difficult to do in an algorithmic manner. The "Onion-Shell" approach does not offer a completely satisfactory solution. While CAD models look quite promising on the computer screen, already the level-3 model has too many consecutive coplanar path segments in its outermost shell. The path patterns in that shell may look like parts of a 2D Gosper curve, but, overall, this would at best yield a "2.5D" Gosper sculpture.

Based on the above criteria, I offer three design approaches as my currently preferred and most promising realizations of 3D Gosper Sculptures. They are in the process of being fabricated as small maquettes on a variety of different 3D printers: "Gosper-Ball_175" (Fig.25a) is based on Figure 10d, "Gosper-Pole_216" (Fig.25b) is based on Figure 16c, and "Gosper-Lattice_351" (Fig.25c) is based on Figure 19c; the last model offers a particularly rich and varied set of projections from different viewpoints. The "Gosper-Onion" models look nice, but they don't really display true 3D Gosper curves.

12. Conclusions

For the Hilbert curve, a well-chosen recursion step can lead to a complete, uniform filling of all of Euclidean 3D space, as well as to finite-size parts thereof that can lead to attractive tubular sculptures with the topology of the unknot. For the Gosper curve, an infinite recursion that fills all of 3D Euclidean space has not yet been found, and it may not exist [8]. On the other hand, this work shows several possible approaches to make compact sculptures that reflect the character of the various Gosper curves, and which could be constructed physically at a human-size scale by using a few hundred tubular segments.

Because of the lack of an elegant recursive formulation of the overall path, I ended up doing a lot of detailed handiwork to draw 3D Gosper-like path fragments and to make the appropriate connections between different clumps in different orientations. An open challenge remains to find a better algorithmic approach to define the overall Gosper-style polyline that makes all the needed connections between the local clumps and possible interstitial lattice sites.



(a) (b) (c)

Figure 25: *Promising styles: (a) Gosper-Ball_175; (b) Gosper-Pole_216; (c) Gosper-Lattice_351. [Model and photo (b) by Nicholas Weaver]*

Acknowledgements

I would like to thank my BAAM friends (Bay Area Artists and Mathematicians), and in particular Jeffrey Ventrella, for their inspiration and stimulating discussions concerning Gosper curves.

I am indebted to Nicholas Weaver and the 'Earl and Pauline Weaver Merit Scholarship' for successfully printing several 3D models on an Ultimaker printer with soluble support material. I also acknowledge Nicole Panditi and other staff members at the Jacobs Institute for Design Innovation for their assistance in realizing some of these difficult-to-fabricate models.

References

[1] M. E. Newell and C. H. Séquin: "The Inside Story on Self-intersecting Polygons." LAMBDA Magazine of VLSI Design, Vol.1, No.2, pp 20-24, May 1980.

[2] C. H. Séquin, "Hilbert Cube 512." Artist's Sketch, SIGGRAPH'06, Boston, July 30 - Aug. 3, 2006.

[3] Shapeways. "3D printing," – https://www.shapeways.com/

[4] J. Smith. "SLIDE design environment." (2003). – https://people.eecs.berkeley.edu/~ug/slide/

[5] Stratasys. "FDM Printers." – https://www.stratasys.com/fdm-technology

[6] V. Uher, P. Gajdo, V. Snasel, "Towards the Gosper Space Filling Curve Implementation." 2017 3rd IEEE International Conference on Cybernetics. – https://www.researchgate.net/publication/318575433_Towards_the_Gosper_Space_Filling_Curve_Implementation

[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] Wolfram MatWorld: "Peano-Gosper Curve." – https://mathworld.wolfram.com/Peano-GosperCurve.html

Dithering by Wires of Orthogonal Images

Firas Habib[1], Sagit Asman[1], Gershon Elber[1]

[1]Computer Science Department, Technion, Israel

Abstract

In this article, we present a method for dithering by wires (DBW) a pair of images by carefully positioning a selected set of 3D line segments or linear wires in 3-space, in the interior of an axis-aligned cube, while the images are on two orthogonal faces of the cube. Wires can either be of arbitrary length or start and end on the faces of the cube, possibly between predefined anchor points. The projections of the aforementioned wires on the XZ and the YZ planes reconstruct dithered versions of the two original images.

The presented approach can operate either with black wires, over a white background, for grey scale input images, or with colored wires, toward the reconstruction of colored images. The output is a set of 3D line segments that represent (linear) wires in space. Any assembly approach for wires can be employed to reconstruct a real tangible representation of the results, while, in this work, we only demonstrate computer-based results, ready for such assemblies.

Keywords: Line-art, Random algorithm, Color and B&W dithering

1 Introduction

The idea of dithering by wires (DBW) is not new. Quite a few artists have employed these ideas over the years. Figure 1 shows one example, created using the scheme presented in this work. In (a), the input image is shown. In (b), the DBW is shown, where the extent of the wires varies. In other words, the wires float in the plane. To ease possible tangible assembly, we also offer the option of having wires that stretch throughout the square domain, as can be seen in (c). More importantly, we seek to place the wires in 3-space, either floating inside an axis-aligned cube or spanning from one face of the cube to another (as in Figure 1 (c)), in such a way that the wires project into two different directions and yield two different DBWs of two independent images.

The common solution to DBW, solves a 2D problem of one given image. As stated, in this work, we seek to simultaneously dither two images by 3D lines. Consider two square independent images, $\mathcal{I}_i$, $i = 1, 2$ of the same size, and assume the two input images are on the XZ and YZ planes of a cube. We like to find a set of 3D lines of finite thickness (i.e., linear wires, or simply *wires* hence after) that when projected on the XZ and YZ planes, will dither-approximate the two input images. For now, and unless otherwise stated, we assume black wires over a white background.

A naive approach to solve this problem can be combinatorial. Select n^2 anchor points in a 2D grid over each of the six faces of the cube. Then, exhaustively search for a possible optimal wire between the $O(n^4)$ pairs of possible locations on two different faces, in each step of the algorithm. Considering that typically, $O(n)$ operations are required to evaluate the contribution (or score) of a wire, this approach is going to be prohibitively expensive, for any reasonable n.

Hence, we consider a stochastic algorithm that randomly generates spatial wires that are either between points on different faces of the cube, or of arbitrary lengths inside it. The algorithm controls how many wires are created and allows some fairness control between the two input images (possibly giving some priority to one of the images).

The rest of this work is organized as follows. In Section 2, some related previous work is discussed. In Section 3, we present the stochastic algorithm, whereas in Section 4, results are presented. Finally, we conclude in Section 5.

(a) (b) (c)

Figure 1: *An example of dithering by wires (DBW) the picture of the Mona Lisa, in (a). 4500 lines are employed in this dithered (converted to grey level input) image, in (b) and (c). As a 2D problem, (b) shows a 2D result, for arbitrary length wires. However, the wires can also be in 3-space, stretched between two faces of an axis-aligned cube, as is shown in (c) in a general view, only to be projected onto 2D planes (faces of the cube) in multiple directions, resulting in (independently) dithered images by wires. This is where this work begins...*

.

2 Previous Work

While academically little can be found in dithering and rendering by wires or string art, in the arts there are some relevant results, all planar. Petros Vrellis [14] creates a way to knit and dither portraits using threads and the placement of the threads is based on 2D computer simulation. Linify [2] shows another 2D example of line art computer simulation for dithering planar images. In [11], Roberto Reif also shows an attempt at a computer-based simulation of line art.

Knitter [12] by Christian Siegel is a software package for creating knitting, with circular/square frames, from an image and is publicly available. Several approaches are presented to solving the 2D problem, adjusting the locations of the pins (randomized, equally spaced and other patterns of spatially spreading the pins).

KnitReport [5] by Malo Malcom Drougard studied previous results like [12] and implemented 2D line art simulation, that is made public. This report also displays quantitative comparisons between the different variants.

This work aims to employ wires in space so they project into independent dithered images, in different directions. In this sense, it is related to similar previous work, such as [6, 10], in which the synthesized 3D geometry projects into independent 2D shapes, in different directions. For examples, Voxel based synthesized geometries, deformed spherical volumetric cells, and deformations of silhouette areas from different directions, were exploited.

In [3], a multi-axis robotic arm is employed toward the automation of the assembly process of planar string arts. A similar multi-axis robotic approach can possibly be used to fabricate 3-space cuboid string arts, as we proposed in this work.

In summary, solutions to the planar problem are pretty much known and some previous results have used a similar stochastic approach as in here. However, in this work, we seek to take the problem into a higher dimension, placing lines in 3-space and have their projection onto orthogonal planes yield different, independent, dithered images.

3 Algorithm

Before we present the stochastic algorithm in Section 3.1, we need to introduce some preliminaries. Consider an intermediate step of the algorithm, having the current state of the dithering process and a new wire to select next. The wire that will be selected is the one that maximizes some optimization function, for example, covering most uncovered pixels so far. The algorithm is greedy in the sense that it chooses the best wire at each step and updates the stored data to that effect. As a result and given a budget of n wires, the algorithm ensures no global optimum in any sense. Not only that different locations of n wires might yield a superior result, but clearly, it can very well be that a less optimal local choice of a wire at some step will yield a more optimal answer, globally, at the end.

During the execution of the algorithm, we hold two auxiliary versions of each of the input images:

1. **StateImage** that is initialized with the original input image. In each iteration, we subtract the new selected wire from **StateImage**, so that in the following iteration, that same wire will score little.
2. **DitheredImage** - Initialized as a background image, and in each iteration, the selected wire with the maximized score is added to this image.

The score of a wire (and possibly the selected color of the wire) is calculated by iterating over all the pixels that are covered in the image by the wire. A variation of the Bresenham algorithm [8] is employed toward this end. We further assume that no selected wire will be parallel to the cubes' faces. I.e., given $wire(P_0, P_1)$, with end points $P_0(p_0^x, p_0^y)$ and $P_1(p_1^x, p_1^y)$, we enforce $p_0^x \neq p_1^x, p_0^y \neq p_1^y$ and $p_0^z \neq p_1^z$. This restriction inhibits the possibilities that the algorithm will independently dither the two images with wires that project to points in the other image.

Given the two input images, where one image is dominantly darker (considering the average intensity of the pixels in the given image), the scores of the wires will be dominated by the darker image, leading to wires that neglect the lighter image. To alleviate this difficulty, we introduce a user-controlled multiplicative fairness score for each of the images, effectively controlling the average intensity.

The number of wires we use should be limited. Too few will clearly fail to approximate and dither the input images to a satisfactory level. Too many wires and we will end up with data sets that will be difficult to fabricate. Hence, striving to minimize the number of wires while achieving dithering results that are appealing, we introduce a heuristic in our algorithm to elevate the importance of critical zones in the input images. We assign pixels that lay on important features in the input images a higher score compared to pixels that lay inside a monotone zone. In essence, we will force the algorithm to focus on edges, in a similar way artists typically work and sketch. While the non-photo-realistic rendering (NPR) community has invested a significant effort into the computation of saliency of visual features [9], herein the salient edges are derived by using the simple Sobel edge detection operator [13]. See, for example, Figure 2.

3.1 The Stochastic Algorithm

Consider two square independent images, $\mathcal{I}_i,\ i = 1, 2$ of the same size and relative fairness $\mathcal{F}_i$, to be dithered with N_{wires} 3D wires. Algorithm 1 presents the top level steps. Algorithm 1 receives $\mathcal{I}_i,\ \mathcal{F}_i,\ i = 1, 2$ and N_{wires}, but also the number of random trials, *NumRndm*, that will be conducted for each selection of a new wire. Algorithm 1 iterates N_{wires} times and fetches the best wire it can find at each step, in Line 4, by calling **FindNextWire**, in Algorithm 2. Then, the best wire found is subtracted from the two **StateImage** buffers, in Lines 7 and 8 of Algorithm 1, and is added to the respective **DitheredImage** buffers, in Lines 9 and 10.

Algorithm 2 randomly generates a new wire, in Line 4: for wires that end on the faces of a cube, **GenRandomWire** generates random wires between two random points on two different random faces of the cube, while ensuring the created wire is co-planar with no face of the cube. For wires interior to the cube, **GenRandomWire** simply generates wires between two random locations inside the cube. For each such new

Figure 2: *An image of Michelle Obama is presented on the left. Automatic features extraction using a Sobel edge detection filter is presented on the right.*

Algorithm 1 StochasticDither - a randomized wire selection algorithm.

Input:

$\mathcal{I}_0$, $\mathcal{I}_1$: Two square images of similar size. Holding both the **StateImage** and **DitheredImage**;

N_{wires}: Number of wires to dither $\mathcal{I}_0$ and $\mathcal{I}_1$ with;

$\mathcal{F}_0$, $\mathcal{F}_1$: Fairness weights for $\mathcal{I}_0$ and $\mathcal{I}_1$;

$NumRndm$: Number of random wire trials, per new wire;

Output:

$\mathcal{W}$: A list of N_{wires} wires in 3-space dithering $\mathcal{I}_0$ and $\mathcal{I}_1$;

Algorithm

1: $N := 0$;
2: $\mathcal{W} := \emptyset$;
3: **while** $N \neq N_{wires}$ **do**
4: $\{P_0, P_1\}$:= **FindNextWire**($\mathcal{I}_0$, $\mathcal{I}_1$, $\mathcal{F}_0$, $\mathcal{F}_1$, $NumRndm$); ▷ Algorithm 2
5: P_0^{xz}, P_1^{xz} := **ProjectXZ**(P_0, P_1); ▷ Project points P_0, P_1 onto the XZ plane
6: P_0^{yz}, P_1^{yz} := **ProjectYZ**(P_0, P_1); ▷ Project points P_0, P_1 onto the YZ plane
7: **SubtractWire**($\mathcal{I}_0$, P_0^{xz}, P_1^{xz}); ▷ Subtr. the pixels in line (P_0^{xz}, P_1^{xz}) from $\mathcal{I}_0$.**StateImage**
8: **SubtractWire**($\mathcal{I}_1$, P_0^{yz}, P_1^{yz}); ▷ Subtr. the pixels in line (P_0^{yz}, P_1^{yz}) from $\mathcal{I}_1$.**StateImage**
9: **AddWire**($\mathcal{I}_0$, P_0, P_1); ▷ Add the pixels in line (P_0^{xz}, P_1^{xz}) to $\mathcal{I}_0$.**DitherImage**
10: **AddWire**($\mathcal{I}_1$, P_0, P_1); ▷ Add the pixels in line (P_0^{yz}, P_1^{yz}) to $\mathcal{I}_1$.**DitherImage**
11: $\mathcal{W} := \mathcal{W} \cup \{P_0, P_1\}$; ▷ Add wire to returned set
12: $N := N + 1$;
13: **Return** $\mathcal{W}$;

---

random wire, the algorithm computes its score with respect to the two input images, in Lines 7 and 8, by calling **CalculateWireScore**, in Algorithm 3. The weighted score is then compared to the best score so far and kept if better.

Algorithm 2 FindNextWire - find the next best wire to use in the DBW.

Input:
$\mathcal{I}_0$, $\mathcal{I}_1$: Two square images of similar size;
$\mathcal{F}_0$, $\mathcal{F}_1$: Fairness weights for $\mathcal{I}_0$ and $\mathcal{I}_1$;
$NumRndm$: Number of random wire trials, per new wire;
Output:
W: The best wire found in the random search for the next wire;
Algorithm

```
 1: BestScore := 0;
 2: BestPins := ∅;
 3: for i := 0; i < NumRndm; i++ do
 4:     P0, P1 := GenRandomWire();
 5:     Score0 := CalculateWireScore(I0, "XZ", P0, P1);          ▷ Algorithm 3
 6:     Score1 := CalculateWireScore(I1, "YZ", P0, P1);          ▷ Algorithm 3
 7:     ScoreSum := Score0 * F0 + Score1 * F1;
 8:     if ScoreSum > BestScore then
 9:         BestScore := ScoreSum;
10:         BestPins := {P0, P1};
11: return BestPins;
12:
```

Finally, **CalculateWireScore**, in Algorithm 3, computes the score of a give wire, in the current dithering state. For every pixel in that wire, it computes the difference, in Line 5, between the desired grey level of that pixel and the black color of the wire. Line 5 also takes into consideration the weight of that pixel, following the saliency of the pixel. The function **Diff** in Algorithm 3, Line 5, is the function the defines the optimum criteria. For grey level images, we compute the level in which a candidate wire darkens a specific pixel with respect to the desired level of the pixel and its level in the current state image. A wire will be chosen, if its average contribution per pixel is the highest. This, so longer wires will not necessarily be favored over shorter ones, due to their length.

4 Results

All the examples presented in this work were created using an implementation of the presented algorithms in the IRIT geometric modeling kernel and are now available as part of [7]. All presented examples employed input images of around 300×300 pixels. The number of random trial, in the stochastic algorithm, per new wire ($NumRndm$ in Algorithm 2) was typically 1500, unless stated otherwise. Computation was conducted on an i7 3.4GHz machine with 32G of RAM, and a single thread.

As stated at the beginning, the combinatorial exhaustive approach is feasible but extremely expensive. We provide one example that employs the exhaustive combinatorial search, in Figure 3. Note the equally spaced pins, on the left and right sides of Figure 3 (b) and (d), where the wires end. The computation time of this specific result for 35^2 pins on each face was 52 hours! Figure 4 shows a similar example using the stochastic algorithm that was executed in about 10 seconds. Interestingly and while subjective, the quality of the stochastic result is not visually inferior to the combinatorial exhaustive search.

Algorithm 3 CalculateWireScore - compute the score of the given wire with respect to image $\mathcal{I}$.

Input:

$\mathcal{I}$:= Input image;

$\mathcal{P}$:= Plane to project the wire on;

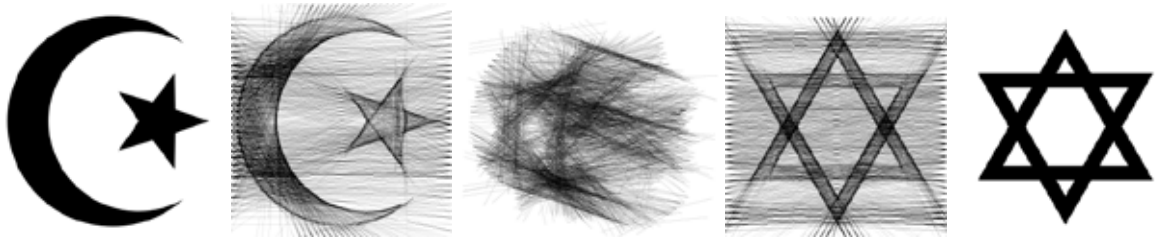
P_0, P_1: End points of a wire to consider its score;

Output:

$Score$: The score of this specific wire in $\mathcal{I}$;

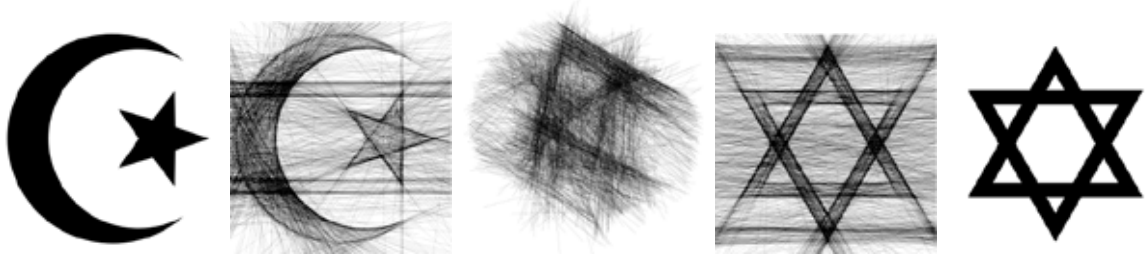
Algorithm

1: $Score := 0$;
2: $Len := 0$;
3: $P_0^{\mathcal{P}}, P_1^{\mathcal{P}} :=$ Project$(P_0, P_1, \mathcal{P})$;
4: **for** every $Pixel \in$ **Line**$(P_0^{\mathcal{P}}, P_1^{\mathcal{P}}, \mathcal{I})$ **do** ▷ E.g. a Bresenham-like algorithm
5: $\quad Score := Score +$ **Diff**$(\mathcal{I}[Pixel].$**StateImage**$.Color, BlackColor) * \mathcal{I}[Pixel].Weight$;
6: $\quad Len := Len + 1$;
7: **return** $Score/Len$;
8:



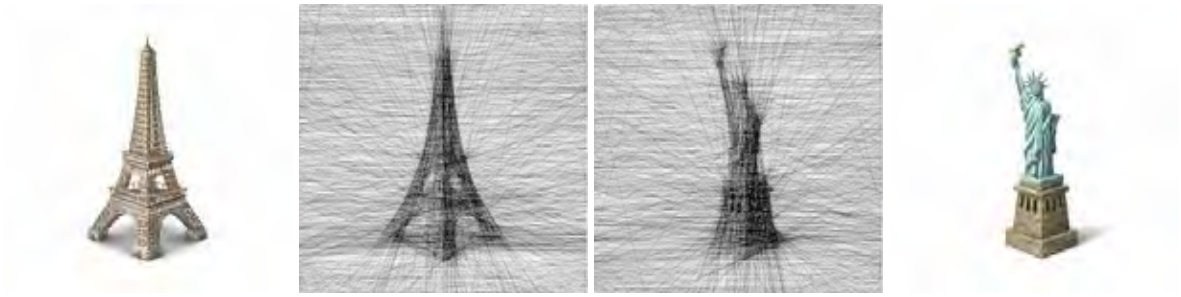
(a) (b) (c) (d) (e)

Figure 3: *The two icons, in (a) of the Islamic crescent moon and the Jewish David Star in (e), were simultaneously DBW with 3D wires that terminate on the faces of a cube and using a combinatorial approach, as can be seen in (b) and (d), while (c) shows a general view of the 3D wires. 2000 wires, $1:1$ Fairness, 35^2 pins on each face of the cube. Over 52 hours of computation time! Icons (a) and (e) are from https://commons.wikimedia.org/wiki/File:Symmetric_religious_symbols.svg. Compare with Figure 4.*



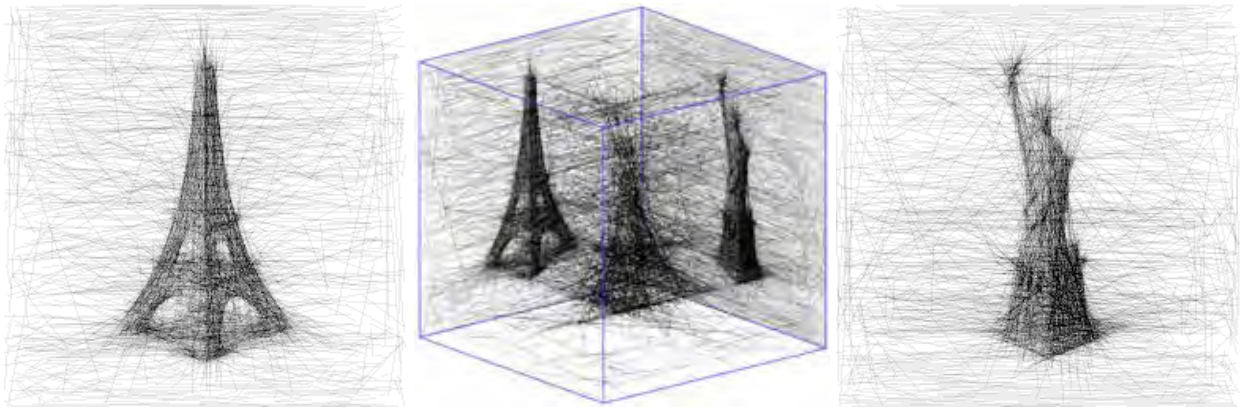
(a) (b) (c) (d) (e)

Figure 4: *The two icons, in (a) of the Islamic crescent moon and the Jewish David Star in (e), were simultaneously DBW with 3D wires that terminate on the faces of a cube and using a stochastic approach, as can be seen in (b) and (d), while (c) shows a general view of the 3D wires. 2000 wires, $1:1$ Fairness. About 10 seconds of computation time. Icons (a) and (e) are from https://commons.wikimedia.org/wiki/File:Symmetric_religious_symbols.svg. Compare with Figure 3.*



(a) (b) (c) (d)

Figure 5: *The two icons, in (a) of the Eiffel tower and in (d) of the statue of liberty, were simultaneously DBW with wires that terminate on the faces of a cube and using a stochastic approach. 2000 wires, 1 : 1 Fairness. Input icons are from https://iconarchive.com (http://www.iconka.com).*



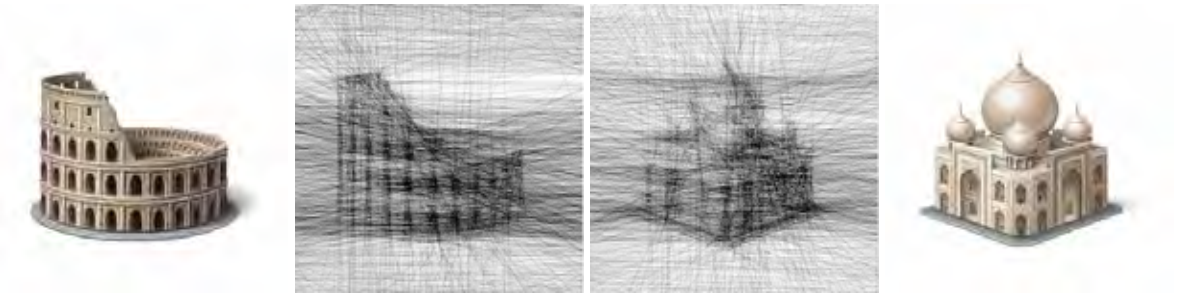
(a) (b) (c)

Figure 6: *The two icons From Figure 5 were simultaneously DBW with wires that float inside a cube, in (a) and (c), using a stochastic approach. (b) shows the same set of wires of (a) and (c) but in a general 3D view along with the two dithered projections, on two different faces of the cube. 2000 wires, 1 : 1 Fairness.*

The rest of the results presented in this section were created using the stochastic approach. Figures 5 to 8 show examples of iconic monuments that are simultaneously DBW. Figure 6 shows a similar result to Figure 5 using wires that float inside the cube, and also shows the wires in 3-space and how they project into two dithered images, in two different faces of the cube. Figure 7 shows a second DBW example for a set of two icons, while Figure 8 employs the same icons and setup as in Figure 7, to exemplify the effect of modifying the number of random trials per wire, that as stated, is 1500 trials by default. Clearly, the more trials the better the expected result will be, but with a direct cost over the computation times. Figure 9 depicts this more systematically, showing (as expected) that the more trials we perform, the better the score of the selected wire, throughout the selection process, of 2000 wires. More interesting are the slopes of these plots in Figure 9. A steep plot denotes the quick loss of contribution, for the n'th wire, whereas a shallow plot means the better preservation of the contribution of the n'th wire.

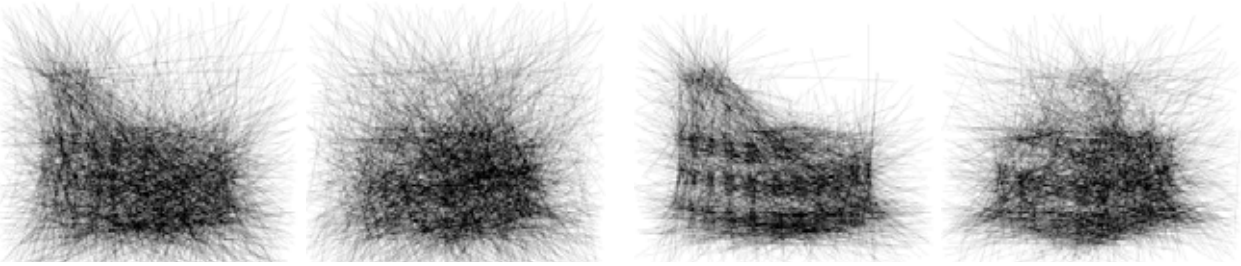
Figure 10 shows a simultaneous DBW of the pictures of Albert Einstein and Stephen Hawking, using wires that span between faces of a cube. 5500 wires were employed. Clearly, the number of wires has an immediate affect over the result and in Figure 11, we show the same result as in Figure 10, while varying the number of dithered wires.

Figure 12 shows the Obamas simultaneously DBW, using wires that span between faces of a cube. 4000 wires are employed with fairness that is very weakly biased toward Barack. In Figure 13, we experiment

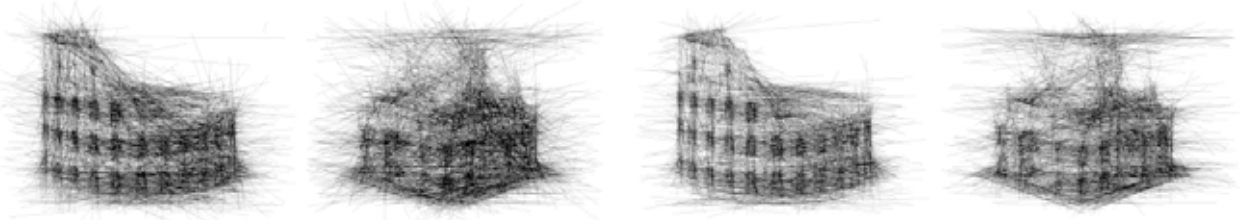


(a) (b) (c) (d)

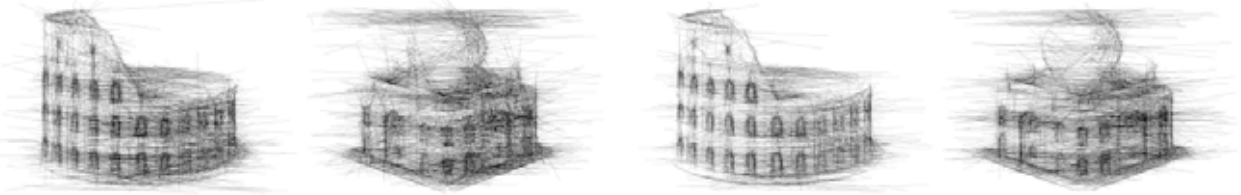
Figure 7: *The two icons, in (a) of the Colosseum and in (d) of the Taj Mahal, were simultaneously DBW with wires that terminate on the faces of a cube and using a stochastic approach. 2000 wires, 1 : 1 Fairness. Input icons are from https://iconarchive.com (http://www.iconka.com).*



(a) 5 Trials (0.27 Sec). (b) 25 Trials (0.36 Sec).



(c) 125 Trials (1.05 Sec). (d) 625 Trials (4.66 Sec).



(e) 3125 Trials (21.0 Sec). (f) 15625 Trials (102.7 Sec).

Figure 8: *The NumRndm trials per wire, in the stochastic Algorithm 2, should be a balance between computation times and the quality of the results. Here we show several results similar to Figure 7, using 3D wires that float in the cube, where only NumRndm trials is being modified. 2000 wires, 1 : 1.5 Fairness. See also Figure 9.*

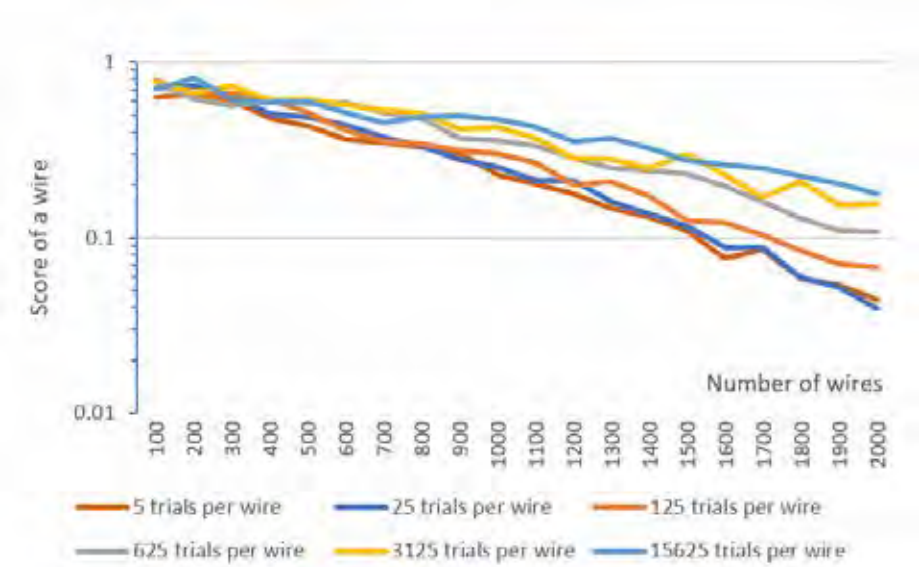
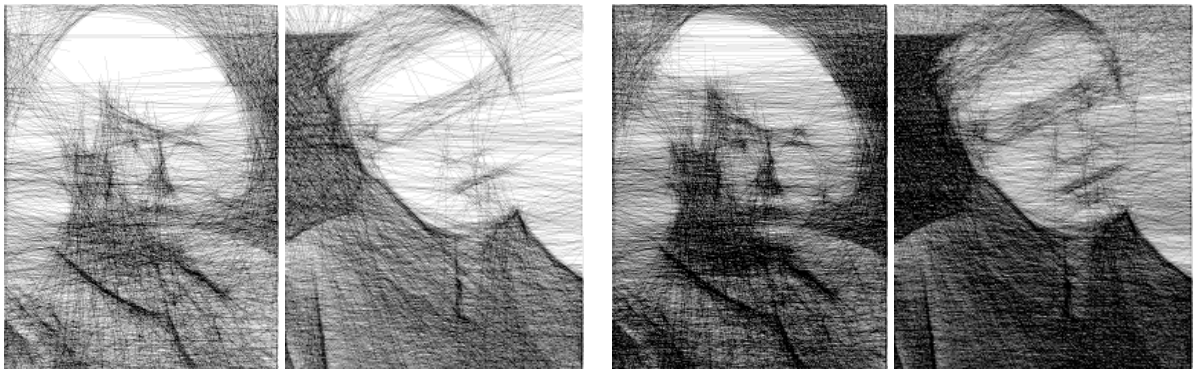


Figure 9: *Some statistics on the scores of the 2000 selected wires (horizonal axis), using data from Figure 8. Shown is the relative score (larger is better, vertical axis) of the selected wire, for the six examples in Figure 8, with different values of NumRndm trials per wire. Note the better scores, for more random trials, as well as the preservation of the scores, as the selection process progresses.*

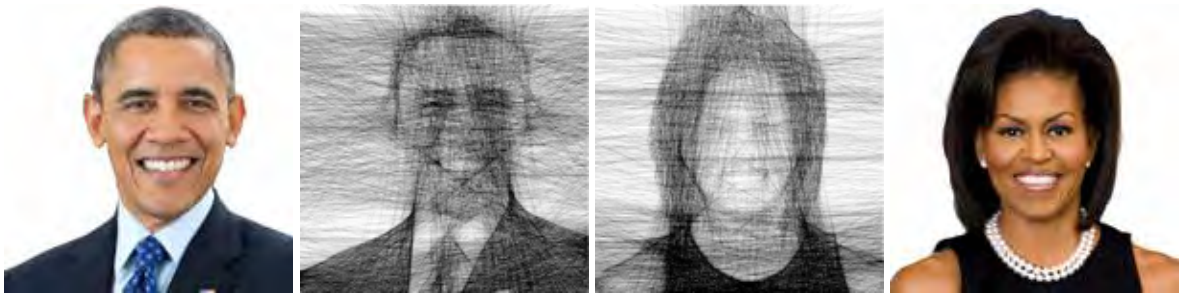
(a) (b) (c) (d)

Figure 10: *The image of Albert Einstein in (a) and Stephen Hawking in (d), are simultaneously DBW, with wires that terminate on the faces of a cube and the results are shown in (b) and (c). 5500 wires, 1 : 1 Fairness.*



(a) 2500 wires. (b) 4500 wires.

Figure 11: *The number of wires has an immediate affect over the results. Here we employ all the parameters as in Figure 10 except the number of wires. Compare with Figure 10.*



(a) (b) (c) (d)

Figure 12: *The image of Barack Obama on the left and a picture of Michelle Obama on the right are simultaneously DBW, with wires that terminate on the faces of a cube and the results are shown at the center. 4000 wires, 1.5 : 1 Fairness.*

with the fairness degree of freedom, in which high fairness for an image means that this image is considered more important. Finally, in Figure 14, we examine the influence of the features (saliency) in the image (recall Figure 2).

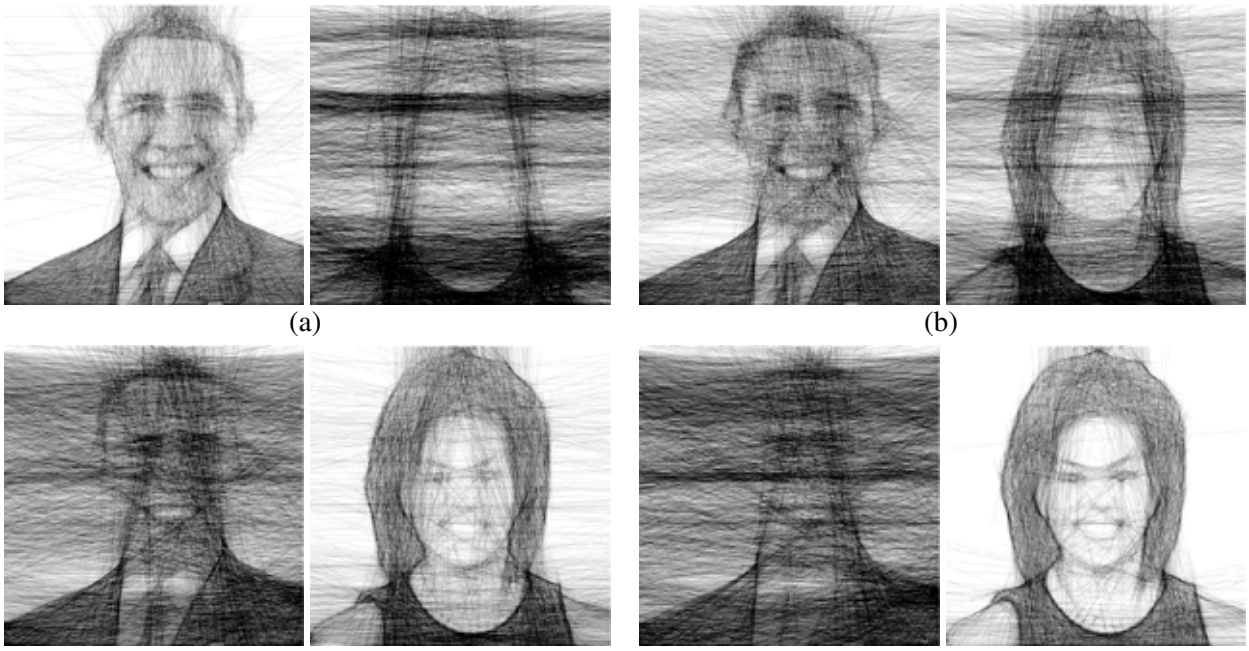
We complete our results section with a few colored ditherings. So far we have assumed the wires are black. Specifically, in Algorithm 3, Line 5, we compared against a black wire. However, one can perform two traversals over the pixels of a wire. The first pass will accumulate the current colors in the input image along the wire, only to compute an average color that is the best for this wire. Then, the second pass will be similar to Algorithm 3, Line 5, only this time the comparison will not be to a black color, but to the computed average color from the first pass.

In Figure 15, two independent examples are dithered from outdoor sceneries, using colored wires of arbitrary length. Figure 16 simultaneously DBW two sunflower drawings of Van Gogh, with wires of arbitrary length, and Figure 17 shows an example of a simultaneous DBW of two outdoor desert sceneries, with wires that span and ends on different faces of a cube.

Finally, Table 1 provides some statistics on several of the examples presented in this work. Timing information, input image sizes and number of wires are portrayed.

5 Conclusions and Future Work

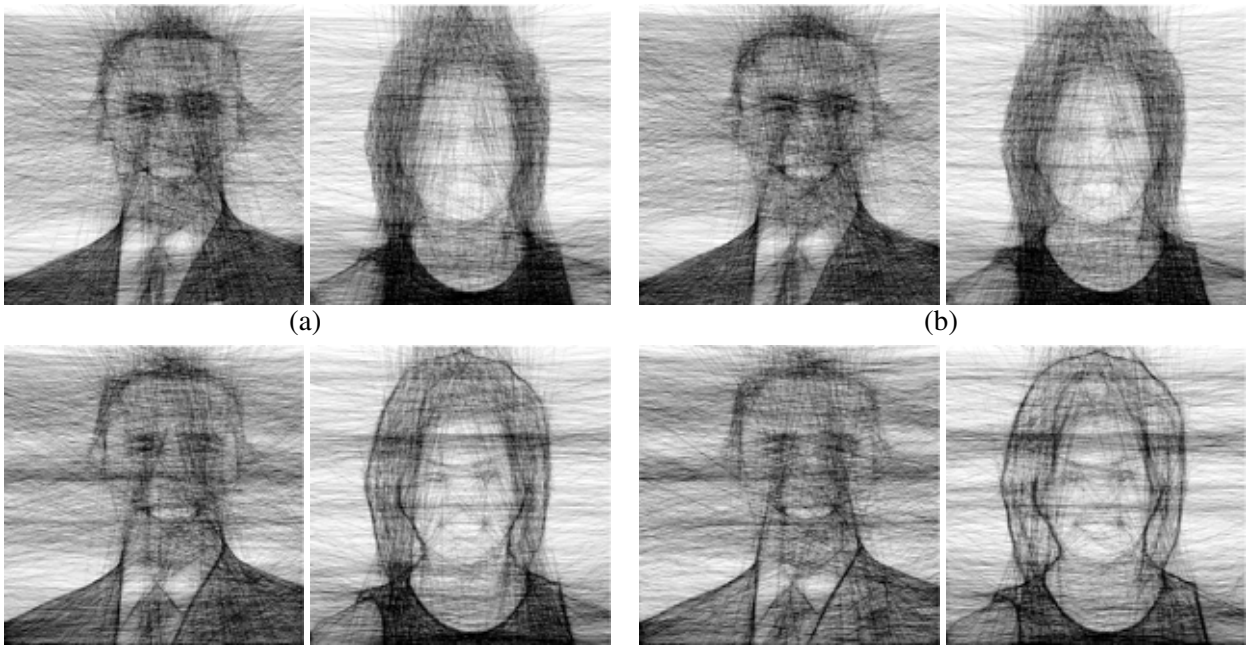
In this work, we have shown that 3D dithering by wires is feasible, for 3D wires that dither a pair of images, simultaneously. Clearly, this work opens directions for further possible extensions. To begin with and while



(a) (b)

(c) (d)

Figure 13: *Figure 12 showed the Obamas with a minor biased fairness toward Barack. Here, in (a), the same example is shown with a strong biased toward Barack, (b) is weakly biased toward Barack, (c) shows similar fairness and (d) is weakly biased toward Michelle. A weak bias toward Barack is probably yielding the best balanced result for both. 4000 wires.*



(a) (b)

(c) (d)

Figure 14: *Examining the influence of feature enhancements (here edge detection - recall Figure 2). Reduced to no feature importance in (a), minimal feature importance is shown in (b), larger than usual feature importance is presented in (c) and very large importance to features is displayed in (d). All other parameters are the same as in Figure 12.*

(a) (b) (c) (d)

Figure 15: *Two independent colored wire ditherings, in (b) and (d), of outdoor sceneries of lakes in Utah, in (a) and (c). The dithering was performed on each individual image, with wires of arbitrary length. 5000 wires, 1 : 1 Fairness.*

(a) (b) (c) (d)

Figure 16: *A simultaneous colored wire dithering, with colored wires of arbitrary length, in (b) and (c), of Sunflowers by Van Goch in (a) and (d). 5000 wires, 1 : 1 Fairness.*

(a) (b) (c) (d)

Figure 17: *A simultaneous colored dithering, with colored wires that terminate on the faces of a cube, in (b) and (c), of outdoor sceneries from parks in Utah, in (a) and (d). 5000 wires, 1 : 1 Fairness.*

Figure	Computation time	Input images size	# of wires
5	10.1 sec.	256×256	2000
6	7.25 sec.	256×256	2000
10	23.8 sec.	196×240	5500
12	66.3 sec.	400×400	4000
16	120.1 sec.	400×500	5000

Table 1: *Some statistics on some of the presented examples.*

we focused on two dithered images in this work, the support of three orthogonal images (on the XY, XZ, and YZ planes) is feasible as well. Interestingly, more than three views could also be considered, not necessarily orthogonal!

The assembly of these results as real tangible objects is clearly a worthy goal, and the fact that we can handle wires that start and end on the faces of a cube makes it a bit simpler. In [3], a robotic arm has been used for 2D assembly of string arts. One can foresee a multi-axis robot with a long needle as its end-effector, stitching these wires one at a time, between two faces of the cube, possibly made from some fabric or net. For example, the needle will be inserted into the cubical space along the wire's direction, from the entry face, stitching the far end at the other face, only to retract and stitch the near end at the entry face.

To make sure no two wires collide in this stitching process, the presented algorithms could be augmented with a collision detection test between wires. Newly selected wires that are too close to previously selected wires will be simply purged in favor of other random wires. Then, the final result will ensure the needle-assemblability of the entire wire arrangement.

Fixing the location of the pins, as is done in the combinatorial exhaustive algorithm (recall Figure 3) can help in the assembly as well. Further, one can aim at DBW of a single long wire, where the end point of the i'th wire will serve as the starting point of wire $i+1$th. Such an arrangement will further ease the assembly process.

In this work we employed linear wires. However, bent (metal) wires could be considered as well, with a significant added complexity of searching the next best wire and its shape.

As shown in the end of Section 4, initial experimentations with colored wires have shown promise, and should be further explored. In fact, wires in images with high resolution sharp details, like trees, performed surprisingly well (i.e., Figure 15).

We have introduced one optimization criteria to select the best wire, which examines the best darkening contribution for grey level. Clearly other possible optimization criteria should be examined.

Another future direction to consider is to possibly employ the (inverse) Radon transform [4] toward DBW from multiple directions, a transform that is typically employed in medical volumetric reconstruction. Similarly, one can consider the exploitation of the Hough transform [1], possibly seeking line (or arc) features, simultaneously, in the input images. Finally, a more advanced saliency computation approach and taking aliasing in the wires into account, in the computation of the best wire, could improve the quality of the result.

6 Acknowledgement

The authors would like to thank anonymous reviewers for their comments which were very useful in improving the quality and expository style of this paper.

This research was supported in part by the ISRAEL SCIENCE FOUNDATION (grant No. 597/18) and in part with funding from the Defense Advanced Research Projects Agency (DARPA), under contract HR0011-17-2-0028. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

7 Videos

is

Several videos exemplifying the 3D wires in space, can be seen in https://youtu.be/OJs6pLu5DCg, https://youtu.be/BR5tOlm5YAU, and https://youtu.be/NFCzW1qfdX4

References

[1] The hough transform, https://en.wikipedia.org/wiki/Hough_transform.

[2] Linify me, http://linify.me/.

[3] Birsak, M., Rist, F., Wonka, P., and Musialski, P. String art: Towards computational fabrication of string images. *Computer Graphics Forum 37*, 2 (May 2018), 263–274.

[4] Deans, S. R. *The Radon Transform and some of Its Applications*. Dover Publications, Inc., Mineola, New York, 1993.

[5] Drougard, M. M. Knit report, https://github.com/MaloDrougard/knit/blob/master/Doc/knit-final-report.pdf. Tech. rep., Swiss Federal Institute of Technology Lausanne, 2017.

[6] Elber, G. Ortho-pictures: 3d objects from independent 2d data sets. In *Advanced in Architectual Geometry* (Vienna, 2010), Springer-Verlag, pp. 175–192.

[7] Elber, G. The irit modeling environment, version 12. "http://www.cs.technion.ac.il/~irit, June 2021.

[8] Foley, D., van Dam, A., Feiner, S. K., , and Hughes, J. F. *Fundamentals of Interactive Computer Graphics*. Addison Wesley, second edition, 1990.

[9] Gooch, B., and Gooch, A. *Non-Photorealistic Rendering*. A.K. Peters Ltd. Publishers ISBN: 1-56881-133-0, 2001.

[10] Mitra, N. J., and Pauly, M. Shadow art. In *ACM SIGGRAPH Asia 2009 Papers* (New York, NY, USA, 2009), SIGGRAPH Asia '09, Association for Computing Machinery.

[11] Reif, R. Drawing with straight lines, https://www.robertoreif.com/blog/2018/1/7/drawing-with-straight-lines.

[12] Siegel, C. Knitter, https://github.com/christiansiegel/knitter.

[13] Sobel, I. The sobel edge detector, https://en.wikipedia.org/wiki/Sobel_operator.

[14] Vrellis, P. A new way to knit (2016), http://artof01.com/vrellis/works/knit.html, 2016.

SHORT PAPERS

CONSTRUCTION OF PLANAR AND SYMMETRIC TRUSS STRUCTURES WITH INTERLOCKING EDGE ELEMENTS

A PREPRINT

Anantha Natarajan
Departments of Visualization
Texas A&M University
College Station, Texas 77831
ananthanatarajan@tamu.edu

Jiaqi Cui
Departments of Visualization
Texas A&M University
College Station, Texas 77831
jiaqi96@tamu.edu

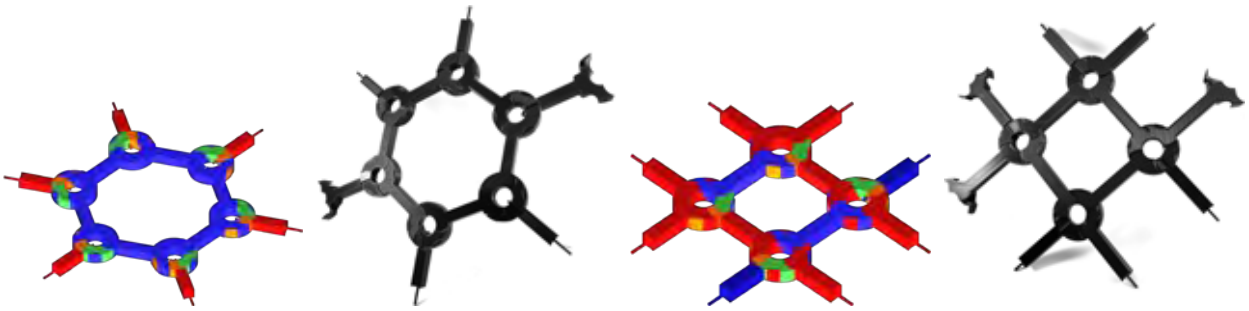
Ergun Akleman*
Departments of Visualization &
Computer Science and Engineering
Texas A&M University
College Station, Texas 77831
ergun.akleman@gmail.com

Vinayak Krishnamurthy
Departments of Mechanical Engineering &
Computer Science and Engineering
Texas A&M University
College Station, Texas 77831
vinayak@tamu.edu

November 7, 2021

ABSTRACT

In this short paper, we present an algorithmic approach to design and construct planar truss structures based on symmetric lattices using modular elements. The method of assembly is similar to Leonardo grids as they both rely on the property of interlocking. In theory, our modular elements can be assembled by the same type of binary operations. Our modular elements embody the principle of geometric interlocking, a principle recently introduced in literature that allows for pieces of an assembly to be interlocked in a way that they can neither be assembled nor disassembled unless the pieces are subjected to deformation or breakage. We demonstrate that breaking the pieces can indeed facilitate the effective assembly of these pieces through the use of a simple key-in-hole concept. As a result, these modular elements can be assembled together to form an interlocking structure, in which the locking pieces apply the force necessary to hold the entire assembly together.



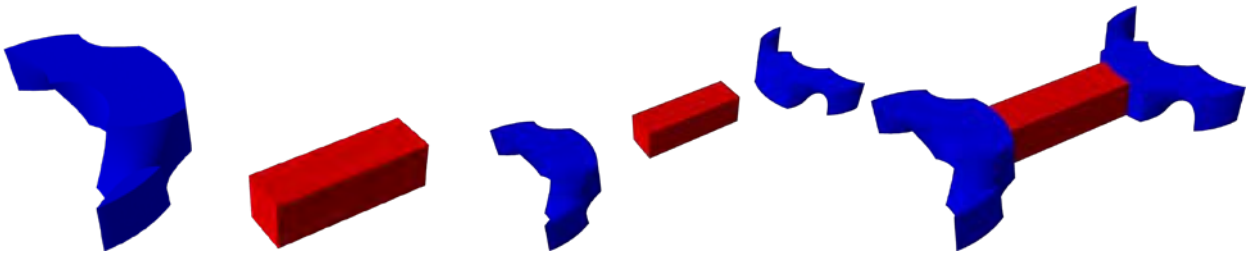
(a) (b) (c) (d)

Figure 1: *Design and Fabrication of modular elements to construct 2D frames: (a) Design of Hexagonal frame, (b) Physical Assembly after fabrication, (c) Design of Quadrilateral frame, (d) Physical Assembly after fabrication*

*website:people.tamu.edu/ ergun

1 Introduction and Motivation

History is rich with examples of puzzle-like interlocking structures that have been studied under the concepts of stereotomy [12, 13, 14], Leonardo grids or Nexorades [4, 6], and topological interlocking [10, 7]. One of the most remarkable examples of interlocking structures is the Abeille flat vault, which was designed by the French architect and engineer Joseph Abeille [15, 3]. Leonardo grids or Nexorades are structures that are constructed using notched rods that fit into the notches of adjacent rods resembling fabric weaves [20, 17].



(a) (b) (c) (d)

Figure 2: *(a) Connector, (b) Truss Edge, (c) Creation of Edge Element, (d) Basic Edge Element*

(a) (b) (c) (d)

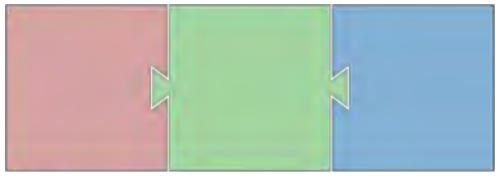
Figure 3: *3D printed versions of three types of interlocking Edge Elements and key pieces to construct 2D hexagonal frames: (a) Two views of a basic Edge Element, (b) Edge Element where one end is replaced by key, (c) Edge element with two keys, (d) Split Interlocking Connector that can locked by a key.*

In this paper, we use **geometric interlocking** concept to develop a framework for designing edge elements for 2D frame structures that can be assembled to form a strong structure [11]. Our process to construct these edge elements is simple simple and intuitive *control* (see edge elements in Figure 3 as an example that can allow to hexagonal frames). Assembled examples are shown in Figure 5. Our method stems from Topological interlocking principle introduced by Estrin et al. [10]. Topological interlocking is defined by Estrin et al. [10] as follows: *topological interlocking is a design principle by which elements (blocks) of special shape are arranged in such a way that the whole structure can be held together by a global peripheral constraint, while locally the elements are kept in place by kinematic constrains imposed through the shape and mutual arrangement of the elements.* The elements support each other by local kinematic constrains resulted from their shapes and mutual arrangements [7, 9, 8, 10].

This paper comes from the observation that a topologically interlocking assembly can be considered a manifold with boundary and the forces are applied in the boundary. If we turn any topologically interlocking assembly into a closed manifold, we can obtain a geometrically interlocking assembly, which is impossible to assemble purely using them without applying at least one or more of the following operations: (1) lifting the pieces up into a higher dimensional space, (2) deforming at least one object (i.e. applying a non-rigid transformation), and (3) cutting at least one piece into at least two or more pieces. Figure 4 provides a comparison of topological and geometrical interlocking. Note that these cases are actually 1-manifold with boundaries, i.e. open curves. If we organize them as a closed curve, i.e., as a 1-manifold; even the topologically interlocked puzzle shown in Figure 4 will turn into a geometrically interlocked puzzle shown in Figure 4c.

Our general conceptual framework to design these connectors is based on partitioning the space using Voronoi decomposition by choosing high-dimensional Voronoi sites (curves and surfaces) [21, 3, 16]. Using this approach, we decompose toroidal shaped volumetric domains using Voronoi decomposition. This decomposition provides us geometrically interlocking assembly, as a 3-space extension of decomposition shown in Figure 4c. Then, we further

(a) *A Topological Interlocking puzzle in 2-space. This puzzle can be assembled together without lifting any piece into 3-space*



(b) *A Geometrically Interlocking puzzle in 2-space. This puzzle cannot be assembled without lifting any piece into 3-space*

(c) *A Geometrically Interlocking puzzle in 2-space that cannot be assembled without cutting one of pieces or lifting one of the pieces up into 3-space.*

(d) *Cutting one piece of the puzzle in 4c makes it possible to assemble the puzzle without lifting any piece up into 3-space.*

Figure 4: Examples of interlocking puzzles in 2D and their relationship with the space in which they are embedded. Note that if a originally topologically interlocking puzzle forms a closed cycle, it corresponds into a 1-manifold with no boundary (see 4c). This particular closed form turns into a geometrically interlocking puzzle and it cannot be assembled without lifting one of the pieces into 3-space or cutting one piece (see 4d). In this paper, we use this peculiar property to obtain connectors.

decompose one of the tiles as shown in Figure 4d to allow assembly. This decomposed piece works as a lock and the small yellow piece in Figure 4d works as a key that can make the whole assembly locked together.

(a) (b) (c)

Figure 5: *Assembly of 3D printed edge elements shown in Figure 3 to construct of 2D hexagonal frames: (a) Assembly of a three sided corner that use all types of Edge Elements, (b) Assembly of a three sided corner that use two basic Edge Element and one Edge Element with two keys, (c) Assembly of one hexagon. This structure can indefinitely be extended to form a hexagonal grid*

In this work, we have developed a new concept to produce symmetric 2D patterns that are constructed as interlocked truss structures. Our concept is based on designing modular *Edge Elements* that can be assembled to create symmetric planar trusses and frames. These modular Edge Elements support each other by virtue of their geometry without the need for any adhesive or binding materials. Our underlying principle is to decompose the joints of a given truss/frame as a set of interlocking shapes. These interlocking shapes which we call *Connectors*, in combination with the edges of the truss, result in *Edge Elements* (Figure 2). These Edge Elements can be connected with each other to physically build the truss (Figure 5). While we demonstrate these Edge Elements through examples of trusses based on uniform and regular planar tessellations, our approach as such can theoretically be expanded for complex 3D trusses to create free form and free spanning grid shell structures. On the other hand, the practical extension of this method to construct 3D trusses is not that simple. We have developed another methodology that directly stems from geometric interlocking

example shown in Figure 4 for designing general n-sided joints for 3D trusses that is geometrically interlocked. Using this method, we have also created one such 3D truss as shown in Figure **??**.

2 Previous Work

Our Edge Elements are inspired from our recent methodology for designing interlocking structures using higher-dimensional Voronoi sites [21, 3]. Allowing any type of shape to serve as Voronoi sites provides **a simple and systematic design methodology** to construct a large variety of interlocking structures for any volumetric domain in 3-space. Based on this point of view, the key parameters are essentially the topological and geometric properties of Voronoi sites and their overall arrangements that are usually be obtained by symmetry transformations (rotation, translation, and mirror operations). The types of shapes that serve as Voronoi sites and their transformations uniquely determine the properties of how the space is partitioned. For instance, let us consider certain special types of partitions such as Delaunay's Stereohedra [5, 18], the Delaunay lofts, generalized Abeille tiles, and bi-axial woven tiles [21, 3, 16].

For Stereohedra, the shapes of Voronoi sites are points, 3D L_2 norm is used for distance computation, underlying space is 3D, and any symmetry operation in 3D is allowed [5, 18]. Based on these properties, we conclude that Stereohedra can theoretically represent every convex space-filling polyhedra in 3D. Since the points are used as Voronoi sites, and L_2 norm is used, the faces must be planar, and edges must be straight in the resulting Voronoi decomposition of the 3D space.

For Delaunay lofts, on the other hand, the shapes of Voronoi sites are curves that are given in the form of $x = f(z)$ and $y = g(z)$, for every planar layer $z = c$ where c is a real constant, a 2D L_2 norm is used to compute distance, underlying space is 2.5 or 3D, and only 17 wallpaper symmetries are allowed in every layer $z = c$ [21]. Based on these properties, we conclude that Delaunay lofts (1) consist of stacked layers of planar convex polygons with straight edges, and (2) in each layer, there can be only one convex polygon. In Delaunay lofts, the number of sides of the stacked convex polygons can change from one layer to another. In conclusion, the faces of the Delaunay lofts are ruled surfaces since they consist of sweeping lines. Edges of the faces can be curved. For generalized Abeille tiles, Voronoi sites can be ruled surfaces or tree-structures, which can significantly extend design space [3]. However, they do not provide geometric interlocking properties.

If the shapes of Voronoi sites are curve segments obtained by decomposing planar periodic curves that are closed under symmetries of bi-axial weaving patterns, the result becomes geometrically interlocking assemblies such as Bi-axial woven tiles [16]. In this paper, we use this particular type of Voronoi decomposition to create our Connectors. Song at al. developed a recursive process to obtain interlocking puzzles in 3D [19]. Using these Connectors as basic building blocks, we construct all types of fundamental modular blocks that we call Edge Elements (see Figure 3).

3 Methodology

Our process starts with a planar graph that represents the truss/frame topology. In this paper, we demonstrate the basic idea through the square and regular hexagonal grids as our truss topology. In our process, all edges of the grid are replaced by square prisms, and all vertices are replaced by a 1-holed toroidal surface with a square cross-section. This toroidal surface is constructed by rotating a square about an axis parallel to a vertical side of the square for a given radius. The toroidal surface encloses a volumetric domain that serves as the interlocking region. Our goal is to decompose this volumetric domain into geometrically interlocked pieces. There exists a variety of methods to obtain topologically, or geometrically interlocking pieces for a given domain, such as Delaunay lofts [21], generalized Abeille tiles [3], or bi-axial woven tiles [16]. In this work, we decompose the domain using an extension of the method that is used to design bi-axial woven tiles. Our process decomposes the toroidal volumetric domain into geometrically interlocking woven tiles which we call Connectors and use these Connectors to create Edge Elements, and it consists of five steps.

1. **Voronoi Site Generation:** The key step in our process is the generation of higher dimensional Voronoi sites. In this case, we use sinusoidal curves as the higher dimensional Voronoi sites to create geometrically interlocking pieces. The number of rotations and the rotation angles depend on the grid pattern to be constructed. For regular hexagonal grids, each vertex is shared by three hexagons. The angle between the edges that share the same vertex is 120 degrees. The toroidal domain at this vertex (Figure 6a) should be decomposed into three identical geometrically interlocking pieces that can be attached to a polygon edge. This means that the initial sinusoidal curves must be rotated 120 degrees about the toroidal domain's axis of revolution. Performing this rotation 3 times gives us three sinusoidal curves that are non-intersecting and non-planar (see Figure 6b). These curves are then deformed uniformly from the axis of revolution to match the horizontal curvature of the

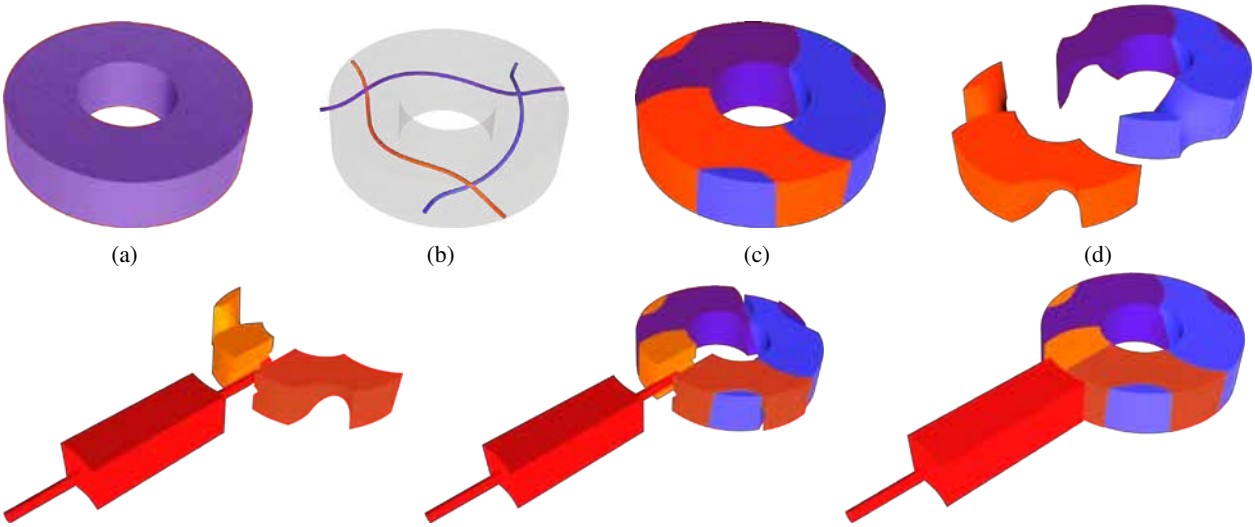


Figure 6: *Visuals that demonstrate the steps of the process: (a) The initial 1-holed toroidal surface with a square cross section, (b) Sinusoidal curves as Voronoi Sites, (c) Voronoi decomposition of the toroidal domain using Sinusoidal curves as Voronoi Sites, (d) One of the Connectors (red one) must be further decomposed, (e)The red Connector is further decomposed to be used as locking pieces, (f) Assembly process, (g) Complete assembly of Connectors*

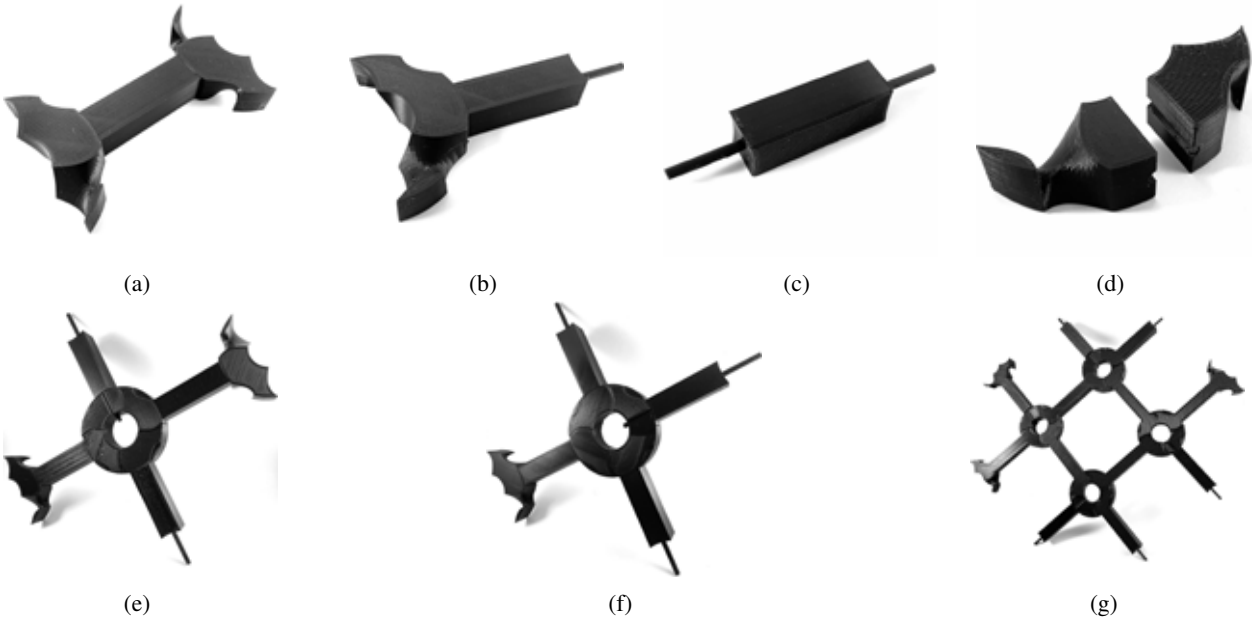


Figure 7: *3D printed interlocking Edge Elements and key pieces to construct 2D quadrilateral frames: (a) A basic Edge Element for quadrilateral grid, (b) One sided locking Edge Element, (c) Two sided locking Edge Element, (d) Split Connector tiles, (e) An assembly of a four sided corner that use all the pieces, (f) Another assembly of a four sided corner that use all the pieces, (g) A complete assembly of a single quadrilateral*

torus. For a vertex with a different valency, the weave pattern generated can be adjusted to create geometrically interlocking tiles by changing the length, amplitude, amount of deformation and number of the curves.

2. **Voronoi Decomposition to Obtain Connectors:** Voronoi decomposition is generally described for points as Voronoi sites. In order to compute Voronoi decomposition for our sinusoidal curves, we draw from previous works [21, 3, 16] and sample points on these curves. We subsequently label all the points belonging to the same curve into a cluster. As a result, we obtain a group of co-labeled Voronoi sites for each curve. We then obtain Voronoi decomposition of the domain using these points. This process decomposes the domain into convex polyhedral regions. Taking the union of the Voronoi cells corresponding to co-labeled sites (i.e., sites belonging to the same curve) results in the partitioning of the toroidal domain into three interlocking woven tiles (Connectors) in a hexagonal grid case. The shapes of these Connectors are identical because of the rotational symmetry of the curves (See Figure 6b). **Remark:** For constructing a square grid pattern, the domain is decomposed into 4 identical Connectors, which means the sinusoidal curve is rotated 90 degrees 4 times to produce 4 non intersecting, non-planar control curves.
3. **Creation of Locking Pieces by Cutting one of the Connectors:** The resulting structure is geometrically interlocking (shown in Figure 6c), and it cannot be assembled without cutting one of the Connectors. We therefore choose one Connector (See the red tile Figure 6d) and decompose it into three pieces, where the peg is the third piece. In our implementation, decomposition is achieved by vertically splitting the Connector into two equal halves with a socket to house the locking peg (See the decomposition in Figure 6e). The pair of half tiles serve as the locking pieces. The peg that is vertically and horizontally centered is shared between the halves and plays the role of a key that holds the assembly(See Figures 6f and 6g).
4. **Creation of Edge Elements** The interlocking Connectors are joined together by a square tube to obtain Edge Elements. The side of the square tube is equal to the side of the cross-section of the toroidal domain. Apart from the locking pieces mentioned in the previous step, there are three types of Edge Elements needed to create any grid pattern. The basic Edge Element type is constructed by attaching a Connector (the original tiles obtained by Voronoi decomposition) to both ends of a square tube. In addition, there are two locking Edge Element types- (1) one-sided locking Edge Element - square tube with the interlocking Connector on one end and a locking peg on the other (2) two-sided locking Edge Element - square tube with a locking peg on both ends.
5. **Printing and Shape Assembly** In order to construct physical assemblies of truss structures, we printed multiple copies of each type of Edge Element and assembled them to create structures as shown in Figures 5a and 5b. This assembly is repeated multiple times as needed and put together to create the grid pattern as shown in Figure 5c. We have created two types of grids: regular hexagon and square as shown in Figures 5c and 7g respectively. In order to create square grid, we designed the Connectors using four sinusoidal curves and printed the Edge Elements similarly (Figure 6a, 6b, 6c, 6d). Resulting vertex assemblies are shown in 7e and 7f and grid assembly is shown in 7g.
6. **Printing and Shape Assembly** While printing our Edge Elements, it was important to understand the tolerances needed for the interlocking Connectors to produce a water-tight fitting and the tolerances needed for the locking peg to be housed effectively by the pair of half Connector tiles. Therefore, to introduce tolerances in out Connector assemblies, we reduced the volume of the individual tile by 0.01 percent. The housing for the locking peg was 0.05 times the locking peg. These values are also based on the PLA material we used and the scale in which our pieces were printed. We print a few of each type of Edge Element. We then assemble them to create structures as shown in Figures 5a and 5b. This assembly is repeated multiple times as needed and put together to create the grid pattern as shown in Figure 5c.

4 Discussion

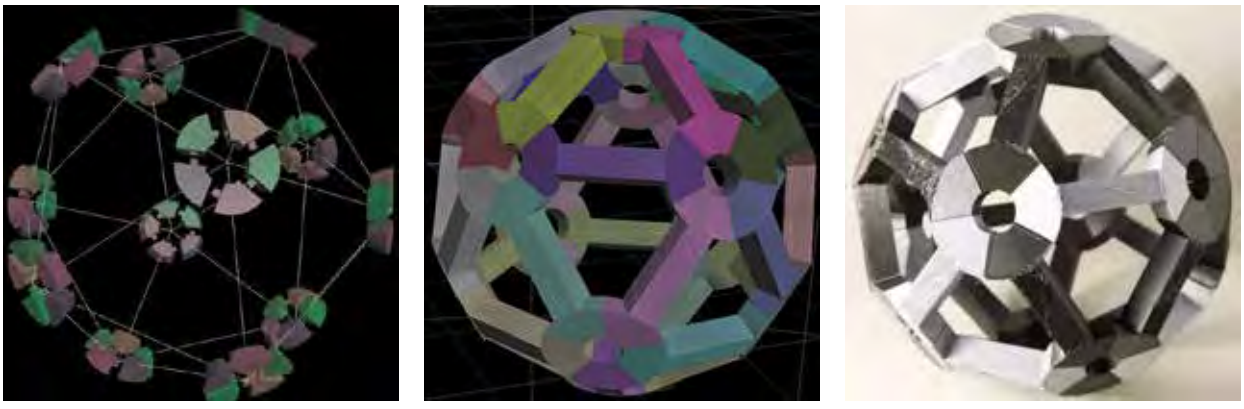
Our preliminary observations show that locking the connectors with keys results in a stable and sturdy assembly. The main advantage of this method is that our Edge Elements can be duplicated economically by mass production such as casting. The modular property of our Edge Elements makes transportation and construction simpler. Our Edge Elements makes the creation of symmetric patterns in architectural and sculptural structures efficient. In a different perspective, they can also be used as construction toys that can help build cognitive ability in kids. However, this work is still preliminary, and there is a lot more that can be done to extend this work. We list some of these extensions that can be practically useful below.

- **Trusses based on Uniform Regular Tessellations:** Our method is not limited to regular mesh structures such as square or regular hexagonal grids. The idea in this paper can be generalized to any grid in which all the vertices are isometric with each other. The simplest of such grids are rectangular grids. This space of grids is

very rich. However, since the angles between consecutive edges around the vertex may not necessarily be the same, there is a need for additional care in organizing sinusoidal curves. The main advantage of this type of grids, the elements can also be mass-produced since they can consist of a very limited set of Edge Elements.

- **Trusses based on Semi-regular and Irregular Tessellations:** The method can directly be extended for trusses based on general tessellations of the plane. Here, we note that the regularity or uniformity (or the lack thereof) of the tessellation would dictate whether we can mass produce a finite set of repeating Edge Elements or if we each Edge Element is completely different from all others (as would happen for a completely arbitrary tessellation of the plane).
- **Regular Polyhedral Trusses:** We observe that the repetitive characteristic of our Edge Elements for regular shapes can be effectively utilized for regular polyhedra, i.e. platonic solids. Since all vertices in Platonic are isomorphic to each other, the Edge Elements can be mass-produced. The sharp angle that exists between edges can be handled by changing the cross-section of the toroidal domain. If a right tangential trapezoid is used instead of the square as the cross-section, the slanted leg of the trapezoid can accommodate the sharp angle created by the edges of the 3D grid. An interesting extension is to build other regular meshes using a similar approach [2, 1, 22]. The Connectors and the corresponding Edge Elements for these will also be one type. Therefore, they can also be mass-produced.
- **Trusses based on Arbitrary Polygonal Meshes:** For meshes that represented arbitrarily shaped surfaces, the design of the Edge Element would naturally be dictated by the curvature characteristics of the surface. Here, we envision that the curvature of such shapes can be achieved by altering the cross-section of the toroidal domain based on the incoming edge. This leads to the creation of a torus with a unique cross-section at every point along its toroidal direction. The sinusoidal curves also need to be organized to fit all the incoming edges at a vertex. However, there is a need for solving both theoretical and practical problems starting from reducing the number of the different elements for economical manufacturing.
- **Mechanical Characterization:** We believe that our Edge Elements, while primarily envisaged for artistic applications, could make a potentially powerful candidate for engineering applications. Indeed, our inspiration to create trusses came directly from structural mechanics. Having said that, a series of deeper numerical, as well as physical experiments are necessary to understand the fundamental behavior of our Connectors and Edge Elements. This could even prove useful in the very design of our proposed structures by allowing us to strategically distribute the key-in-hole locations on the Connectors in an effective way. We use thin pegs in these examples. It is possible to make them thicker and stiffer.
- **Topologically Interlocking Tiles:** In this case, we used a type of woven tiles, which are already geometrically interlocking. We observe that similar results can be obtained by using topologically interlocking pieces such as Delaunay lofts or Generalized Abeille tiles. This can further simplify the design of Connectors and the assembly of elements.

We are grateful to reviewers for their detailed review. We have tried to include their suggestions in the document by creating new models such as 3D dodecahedral shape truss shown in Figure 8.



(a) (b) (c)

Figure 8: *An example of 3D truss constructed by simpler interlocking edge elements. This particular design does not require pegs, however, the pieces needs to be slightly elastic to be able to put together.*

References

[1] E. Akleman and J. Chen. Regular meshes. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 213–219, 2005.

[2] E. Akleman and J. Chen. Regular mesh construction algorithms using regular handles. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 27–27. IEEE, 2006.

[3] E. Akleman, V. R. Krishnamurthy, C.-A. Fu, S. G. Subramanian, M. Ebert, M. Eng, C. Starrett, and H. Panchal. Generalized abeille tiles: Topologically interlocked space-filling shapes generated based on fabric symmetries. *Computers & Graphics*, 89(8):156–166, 2020.

[4] O. Baverel, H. Nooshin, Y. Kuroiwa, and G. Parke. Nexorades. *International Journal of Space Structures*, 15(2):155–159, 2000.

[5] B. N. Delaunay and N. N. Sandakova. Theory of stereohedra. *Trudy Matematicheskogo Instituta imeni VA Steklova*, 64:28–51, 1961.

[6] C. Douthe and O. Baverel. Design of nexorades or reciprocal frame systems with the dynamic relaxation method. *Computers & Structures*, 87(21):1296–1307, 2009.

[7] A. Dyskin, Y. Estrin, A. Kanel-Belov, and E. Pasternak. A new concept in design of materials and structures: Assemblies of interlocked tetrahedron-shaped elements. *Scripta Materialia*, 44(12):2689–2694, 2001.

[8] A. Dyskin, Y. Estrin, and E. Pasternak. Topological interlocking materials. In *Architectured Materials in Nature and Engineering*, pages 23–49. Springer, 2019.

[9] A. V. Dyskin, Y. Estrin, A. J. Kanel-Belov, and E. Pasternak. Topological interlocking of platonic solids: A way to new materials and structures. *Philosophical magazine letters*, 83(3):197–203, 2003.

[10] Y. Estrin, A. V. Dyskin, and E. Pasternak. Topological interlocking as a material design concept. *Materials Science and Engineering: C*, 31(6):1189–1194, 2011.

[11] Y. Estrin, V. R. Krishnamurthy, and E. Akleman. Design of architectured materials based on topological and geometrical interlocking. *Journal of Materials Research and Technology*, 2021.

[12] R. Evans. *The projective cast: architecture and its three geometries*. MIT press, 1995.

[13] G. Fallacara. Toward a stereotomic design: Experimental constructions and didactic experiences. In *Proceedings of the Third International Congress on Construction History*, page 553, 2009.

[14] S. Fernando, R. Saunders, and S. Weir. Surveying stereotomy: Investigations in arches, vaults and digital stone masonry. *of Architectural Research*, page 82, 2015.

[15] F. Fleury. Evaluation of the perpendicular flat vault inventor's intuitions through. large scale instrumented testing. In *Proceedings of the Third International Congress on Construction History, Cottbus*, pages 101–109, 2009.

[16] V. R. Krishnamurthy, E. Akleman, S. G. Subramanian, K. Boyd, C.-a. Fu, M. Ebert, C. Startett, and N. Yadav. Bi-axial woven tiles: Interlocking space-filling shapes based on symmetries of bi-axial weaving patterns. In *Proceedings of the 46th graphics interface conference, GI*, 2020.

[17] R. Roelofs. Three-dimensional and dynamic constructions based on leonardo grids. *International Journal of Space Structures*, 22(3):191–200, 2007.

[18] M. W. Schmitt. *On Space Groups and Dirichlet–Voronoi Stereohedra*. PhD thesis, Berlin: Freien Universität Berlin, 2016.

[19] P. Song, C.-W. Fu, and D. Cohen-Or. Recursive interlocking puzzles. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[20] P. Song, C.-W. Fu, P. Goswami, J. Zheng, N. J. Mitra, and D. Cohen-Or. Reciprocal frame structures made easy. *ACM Transactions on Graphics (TOG)*, 32(4):94, 2013.

[21] S. G. Subramanian, M. Eng, V. R. Krishnamurthy, and E. Akleman. Delaunay lofts: A biologically inspired approach for modeling space filling modular structures. *Computers & Graphics*, 82(8):73–83, 2019.

[22] J. J. Van Wijk. Symmetric tiling of closed surfaces: Visualization of regular maps. *ACM transactions on graphics (TOG)*, 28(3):1–12, 2009.

A New Method to Obtain Positive Genus space-filling Tiles that do not form Links or Knots

A Preprint

Ergun Akleman*
Departments of Visualization &
Computer Science and Engineering
Texas A&M University
College Station, Texas 77831
ergun.akleman@gmail.com

Vinayak Krishnamurthy
Departments of Mechanical Engineering &
Computer Science and Engineering
Texas A&M University
College Station, Texas 77831
vinayak@tamu.edu

Courtney Starrett
Departments of Visualization
Texas A&M University
College Station, Texas 77831
cstarrett@tamu.edu

Chia-an Fu, Doyeon Kim, and Nicholas Cropper
Department of Visualization
Texas A&M University
College Station, Texas 77831

Sai Ganesh Subramanian, and Mathew Ebert
Department of Mechanical Engineering
Texas A&M University
College Station, Texas 77831

November 7, 2021

Abstract

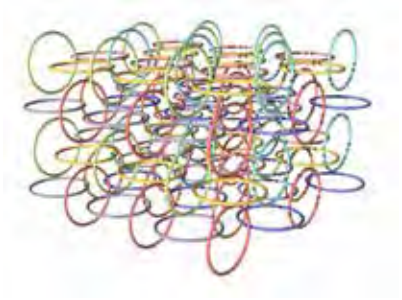
In this work, we present a method to obtain positive-genus space-filling shapes. An important property of these these space filling shapes is that they do not form links and knots. They, therefore, can potentially be assembled and disassembled to construct larger shapes. Our methodology is simply to use close curves that do not forms links and that are closed under symmetry operations as Voronoi sites. Another property of these shapes, they can be combined to create higher genus surfaces. The new combined shapes can also fill the space without having any voids between them. From our preliminary structural evaluations, we posit that higher genus tiles may allow for better inter-locking behavior as well. Therefore, many further studies need to be done to characterize and categorize this class of Voronoi sites that could produce different types of higher-genus tiles and the corresponding mechanical behaviors.

1 Introduction and Motivation

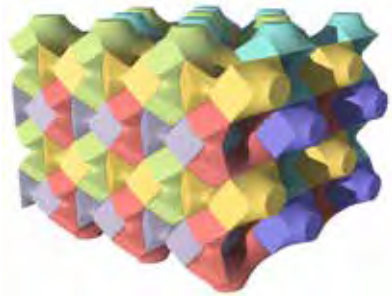
Space-filling shapes are defined as unit cells whose replicas together can fill all of space watertight, i.e. without having any voids between them [9]. In other words, a space-filling shape can be used to generate a tessellation of space. While 2D tessellations and space-filling shapes are relatively well-understood, problems related to 3D tessellations and space-filling shapes are interesting and have applications in a wide range of areas from chemistry and biology to engineering and architecture.

The space-filling shapes are usually expected to be "*cellular*" in the sense that a polygon in 2-space (or in general a disc in $\mathcal{R}^2$) is a cell. In other words, 2D regions with holes (e.g. an annulus) cannot be tiled to fill 2D space. This, however, is not true for 3D shapes. Existence of genus-1 tiles that are knotted and linked has been known since 1990's [1, 8, 10]. However, one problem with these tiles, they cannot be assembled or disassemble if they are manufactured as a single pieces. Carlo Séquin developed a clever way to assemble such linked genus-1 tiles by fabricating them two or more separate pieces that can be interlocked [12]. Decomposing single genus-1 tiles into genus-0 pieces helped to assembled and disassemble the whole linked structures. Egon Schulte showed the existence of positive genus tiles that can be assembled and disassembled [11] without decomposing or breaking. He also provided a methodology to construct them. On the other hand, he did not fabricate them to demonstrate the effectiveness of his methodology.

*website:people.tamu.edu/ ergun



(a) A 3D symmetric assembly of un-linked circles as Voronoi sites.



(b) Voronoi decomposition of the space using circles.

(c) Single tile that can fill 3D space.

(d) Photographs of 3D printed tiles that shows assemblies of one, two, three and four tiles.

Figure 1: An example of the process to obtain positive genus space filling tiles using circles.

In this work, we have developed a new methodology using Voronoi decomposition of space using closed curves that are not linked. We demonstrate that using such closed curves that are closed under symmetry operations it is possible to obtain non-cellular space filling tiles (i.e. positive genus shapes such as a torus) that (1) are connected and (2) can be tiled to tessellate 3D space. Figure 1 shows the basic methodology. Instead of using closed curves, we can also use 3D graphs that are not tress, in other words, the graphs that include cycles. Using such graphs, we can obtain higher genus surfaces, an example is shown in Figure 2. There also exists methodologies for designing Voronoi sites that can result in such positive-genus un-linked space-filling shapes [3, 5].

2 Methodology

Conventional wisdom suggests that donuts can only form links that cannot be separated without cutting such as genus-1 tiles of Carlo Séquin's [12]. Egon Schulte showed the existence of such structures. We also want to point out that the method by Colin Adams can also lead positive genus tiles that can be separated without cutting if we ignore knotting part [1]. However, neither Schulte and Adams fabricated any example as far as we know.

In this work, we have developed a new method to design positive genus (homeomorphic to torus, double torus etc.) space-filling shapes that can be assembled and disassembled without cutting. We have also fabricated a few examples. Such space-filling shapes is *not only* theoretically interesting, *but also* opens up possibilities to provide strong interlocking behavior. These particular shapes initially emerged as by-products of our recent work on generalized Abeille tiles[2]. We observed that we can obtain any arbitrary genus space-filling tile by combining certain types of generalized Abeille tiles (See Figure2) . These combinations turns tree-type Voronoi sites that are used to create Abeille tiles into graphs that includes cycles. These cycles produces helps to produce positive genus space-filling shapes.

The existence of this set of positive genus space-filling shapes suggests that there also exist algorithms to compute them systematically. In this work, we briefly present our basic algorithm to design similar shapes. This algorithm to obtain positive genus space-filling shapes, in general, is a specific version of our approach to obtain a variety of space-filling shape types such as Delaunay lofts [14, 13], Generalized Abeille tiles [2] and Woven tiles [7]. Our approach stems from Delaunay's original intent for the use of Delaunay diagrams. He used symmetry operations on points and Voronoi diagrams to produce space-filling polyhedra, which he called Stereohedra [4].

(a) Five views of a genus-0 space-filling tile. These tiles are cast in aluminum using 3D printed molds.

(b) Two views of a genus-1 space-filling tile obtained by combining two genus-0 tiles.

(c) A genus-2 tile from the combination of three genus-0 tiles .

Figure 2: (Photographs of Fabricated Shapes obtained by our 3D Voronoi Algorithm) Another Example of space-filling arbitrary genus un-linked space-filling tiles. We can obtain any genus by placing genus-0 tiles on top of each other. Each new genus will still be space-filling.

To obtain these new class of shapes, we use higher dimensional Voronoi sites that are closed under symmetry operations. In Delaunay lofts, these higher dimensional Voronoi sites are curves in the form of $x = f_x(z)$ and $y = f_y(z)$ [14]. In Woven tiles, we use segments of periodic curves as Voronoi sites [6]. In Generalized Abeille tiles, we use 3D trees, such as Y-, X- or T-shaped 1-complexes (i.e. simplexes with maximum dimension 1, e.g. trees and curve networks) [2]. This work resulted from our observation that the unions of the tree-shaped 1-manifolds form general 3D graphs with cycles. Voronoi decomposition of 3D graphs with loops results in positive genus decomposition of 3D space and positive genus space-filling shapes. To guarantee that the resulting structures are positive genus and can be fall-apart without cutting, there is a need for some care in designing the Voronoi sites and symmetry operations.

Positive genus space-filling shapes that cannot fall apart without cutting can also be useful in some applications when space-filling chains are needed. However, for space-filling chains, the fabrication can only be done using 3D printing. If our goal is to manufacture each piece individually, symmetry operations must not create linked structures. This problem is related to knot theory and it is trickier than it sounds. To demonstrate the difficulty, assume we have three ellipses. Using these ellipses we can form a linked structure called the Borromean rings. This link will fall apart if we remove any one of the ellipses. In other words, any two ellipses are un-linked and adding the third one makes them linked.

To obtain positive genus structures, we also need to be sure that for each closed curve there must be sufficient empty space in addition to other Voronoi sites that are close to this empty region such that that it can be filled by these Voronoi sites (See for an example created by circles in Figure 1). For another example shown in Figure 2, consider we use two X's top of each other as Voronoi sites. They form a diamond-shaped closed curve in the middle and can potentially result in positive genus tiles. However, we need to orient the other Voronoi sites in such a way that they fill the inside of the diamond shape.

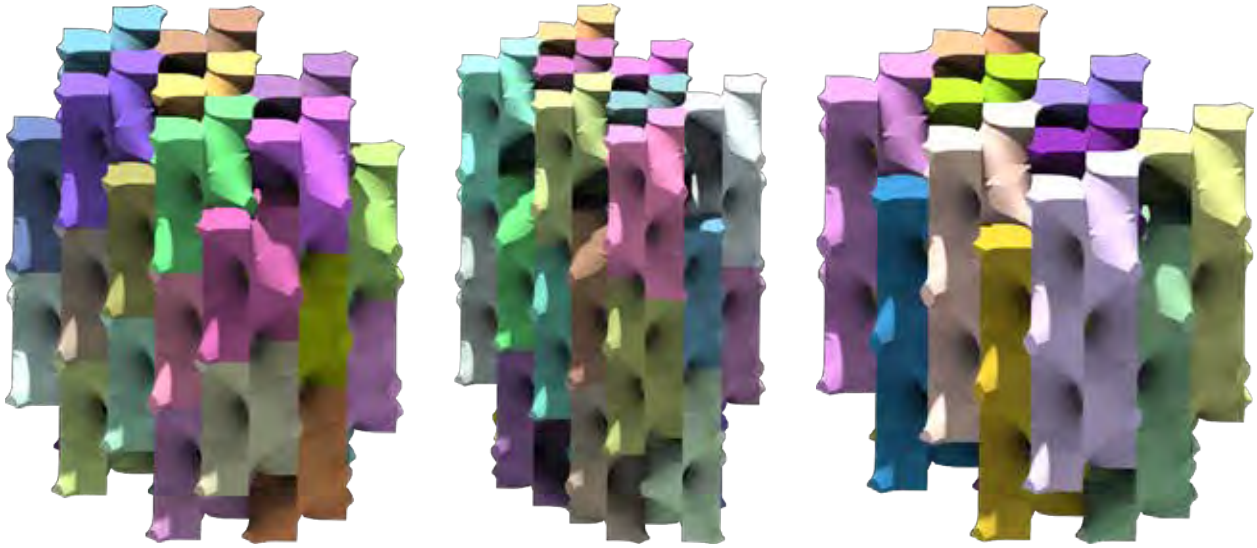
3 Conclusion and Future Work

In conclusion, positive-genus un-linked space-filling shapes provide a significant number of interesting research directions. Particularly in geometry, it will be useful to characterize and categorize the class of Voronoi sites that could produce different types of higher-genus tiles. Since positive genus tiles can provide strong inter-locking behavior, it can motivate studies to evaluate their physical behaviors. We are grateful to reviewers for their detailed review.

References

[1] C. C. Adams. Tilings of space by knotted tiles. *The Mathematical Intelligencer*, 17(2):41–51, 1995.

[2] E. Akleman, V. R. Krishnamurthy, C.-a. Fu, S. G. Subramanian, M. Eng, M. Ebert, C. Starrett, and H. Panchal. Generalized abeille tiles: Topologically interlocked space-filling shapes generated based on fabric symmetries. *Accepted - Computers & Graphics*, 2020.



(a) An assembly of genus-1 tiles (b) An assembly of genus-2 tiles (c) An assembly of genus-3 tiles

Figure 3: (Rendering of Virtual Shapes obtained by our 3D Voronoi Algorithm) Examples of assemblies of genus-1, genus2 and genus-3 space-filling tilling. Note that they are obtained by putting genus-0 vaults top of each other. In this way, we can obtain any genus as tall columns.

[3] T. Castle, M. E. Evans, S. T. Hyde, S. Ramsden, and V. Robins. Trading spaces: building three-dimensional nets from two-dimensional tilings. *Interface Focus*, 2(5):555–566, 2012.

[4] B. N. Delaunay and N. N. Sandakova. Theory of stereohedra. *Trudy Matematicheskogo Instituta imeni VA Steklova*, 64:28–51, 1961.

[5] B. Kolbe and V. Robins. Tile-transitive tilings of the euclidean and hyperbolic planes by ribbons. 2020.

[6] V. R. Krishnamurthy, E. Akleman, S. G. Subramanian, K. Boyd, C.-a. Fu, M. Ebert, C. Starrett, and N. Yadav. Bi-axial woven tiles: Interlocking space-filling shapes based on symmetries of bi-axial weaving patterns. *Accepted - Proceedings of Graphics Interface*, 2020.

[7] V. R. Krishnamurthy, E. Akleman, S. G. Subramanian, M. Ebert, J. Cui, C.-a. Fu, and C. Starrett. Geometrically interlocking space-filling tiling based on fabric weaves. *IEEE Transactions on Visualization & Computer Graphics*, (01):1–1, 2021.

[8] W. Kuperberg. Knotted lattice-like space fillers. *Discrete & Computational Geometry*, 13(3):561–567, 1995.

[9] A. L. Loeb. Space-filling polyhedra. In *Space Structures*, pages 127–132. Springer, 1991.

[10] P. Schmitt. Spacefilling knots. *arXiv preprint math/0107016*, 2001.

[11] E. Schulte. Space fillers of higher genus. *Journal of Combinatorial Theory, Series A*, 68(2):438–453, 1994.

[12] C. Séquin. Intricate isohedral tilings of 3d euclidean space. In *Bridges Conference, Leeuwarden, The Netherlands*, pages 139–148, 2008.

[13] S. G. Subramanian, M. Eng, V. Krishnamurthy, and E. Akleman. Delaunay lofts: a new class of space-filling shapes. In *ACM SIGGRAPH 2019 Posters*, pages 1–2. 2019.

[14] S. G. Subramanian, M. Eng, V. R. Krishnamurthy, and E. Akleman. Delaunay lofts: A biologically inspired approach for modeling space filling modular structures. *Computers & Graphics*, 82:73–83, 2019.

The Proceedings of the SMI 2020 Fabrication & Sculpting Event (FASE)

In Cooperation with ISAMA

9781716880117

FASE Paper Chairs:

Oleg Fryazinov
Carlo H. Séquin

Proceedings Editor:

Ergun Akleman

Front Page Sculpture:

Carlo H. Séquin

Back Page Sculpture:

Carlo H. Séquin

Background Images:

Firas Habib,
Sagit Asman,
Gershon Elber

Cover Design:

Ergun Akleman