



HAL
open science

Collaborating with an Artist in Digital Geometry

Éric Andrès, Gaëlle Largeteau-Skapin, Aurélie Mourier

► **To cite this version:**

Éric Andrès, Gaëlle Largeteau-Skapin, Aurélie Mourier. Collaborating with an Artist in Digital Geometry. 20th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI 2017), Sep 2017, Vienne, Austria. pp.19-30, 10.1007/978-3-319-66272-5_3 . hal-01651824

HAL Id: hal-01651824

<https://hal.science/hal-01651824>

Submitted on 29 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Collaborating with an Artist in Digital Geometry

Eric Andres, Gaelle Largeteau-Skapin, and Aurélie Mourier

Laboratory XLIM, Team ASALI, UMR CNRS 6712,
University of Poitiers, BP 30179,
86962 Futuroscope Chasseneuil Cedex, France
{eric.andres, gaelle.largeteau.skapin}@univ-poitiers.fr
contact@aureliemourier.net

Abstract. In this invited paper, we are going to present an ongoing collaboration with a local artist, Aurélie Mourier. The artist works with voxel shapes and this led our digital geometry team to develop new shape modeling tools and explore a particular class of unfolding problems.

1 Introduction and Context of the Artistic work

In this paper, we are going to present the results of an ongoing collaboration of our Digital Geometry Research team with a local graphical artist, Aurélie Mourier, that happens to be working with cubes. Let us present the motivations behind her work and then we'll present some examples of our research that were partly or completely driven by the artists demands and questions.

An artist typically tries to understand the world around him by proposing his own reproduction and/or by dissecting specific elements of it in order to propose an original point of view. The artist will embody his unique and personal point of view in an art work so that his experience of the world can be shared with the public. Art is not meant to be didactic but singular. The inspiration for his work may come from many different sources and points of view including scientific ones. There are many similarities in the approach used by artists and scientists in particular in the attempts to propose an abstract representation of the world. The artist will use his art form while the scientist will use mathematics as common shared language.

In A. Mourier's case, she focuses on shapes. Those shapes can be extracted from reality or invented. In order to study only the shape of things, she willingly discards parameters such as color, texture and size: a planet, a ball and a marble have all the same shape. It is interesting here to make a parallel with the ideas behind the invention of topology. In her case however, the geometric shape (although in a digital abstract form) still plays an important role. She proceeds by injecting the object she wants to reproduce into a 3D cubic grid similar to what happens when one *pixelizes* a shape in 2D. A resolution was chosen: the grid size of $25 \times 25 \times 25$. This sizes was chosen arbitrarily so that the shapes are just big enough to be recognizable and allow some expressibility. A shape is *well formed* for A. Mourier if it is in one piece, with each cube touching another one by face and touches at least two opposite sides of the grid. She orders the

shapes according to their volume (their number of voxels). She did not know, in the beginning of her artistic work, that her process was called digitization and her *one piece, face-connected* constraint translates in topology as a *single 6-connected component*.

Her inspiration came from the observation that when we look at the world through computer screens, our world view is formed of digital images with a finite fixed resolution. There exists only finite, although very big, number of possible images. If we could create a catalog of all the possible images, it would, for instance, include all the images of all the people in the world, dead, alive and yet to be born. Let us note here that our biological eye has also a finite number of cones and that our brain has a limit on the number of different colors it is able to distinguish. A. Mourier works on a shape repository but more importantly, on how a shape can be represented or coded and how such shapes can then lead to sculptures that reinstate a size and a material with physical properties. This idea of repository is inspired by the novel written in 1941 by J.-L. Borges, "La biblioteca de Babel" [6]. The story of Borges describes a library with all the possible books of 410 pages, each made of 40 lines of about 80 characters each in a alphabet composed of 22 regular letters, and the characters space, coma and point (check out <https://libraryofbabel.info/> for a virtual example of this library). Of course, just as for the Library of Babel, and even though well formed objects represent only a fraction of the $2^{25^3} \approx 3.9 \times 10^{4703}$ possible digital objets that can be represented in her grid, it gives a setting for shape exploration.

Before we met, A. Mourier used to model her voxel objects with generic modeling softwares. She then printed and cut the result by hand to obtain either the unfolded pattern of the shape or its slices. An example of a stereo microscope shape is presented Fig. 1. The number in the name of the shape corresponds to its number of voxels (797) and the order of the object among all the shapes having the same number of voxels. Once a shape has been defined, A. Mourier is interested in all the ways such a shape can be represented: as a set of voxels in 3D, as a set of 2D slices, unfolded as a net, etc.

2 Creating Digital Shapes for an artist

As an artist working on her shape depository, A. Mourier used to design her digital objects by hand. Coming from a completely different background, terms like *digitization*, *6-connectivity* or *digital geometry* were unknown to her, on the other hand she had manipulated such shapes and representation forms for a couple of years and often has a better intuition than we could possibly have on how to do certain operations. For her, it was interesting to put scientific words on notions she was manipulating intuitively. For us, it was interesting to discover new classes of problems with specific applicative constraints.

The starting point of our collaboration was through a student project that created a simple voxel modeling software that lets you freely create voxel objects. This tool can be found online at <http://www.aureliemourier.net/logiciel/25aucube.html>. However, with this

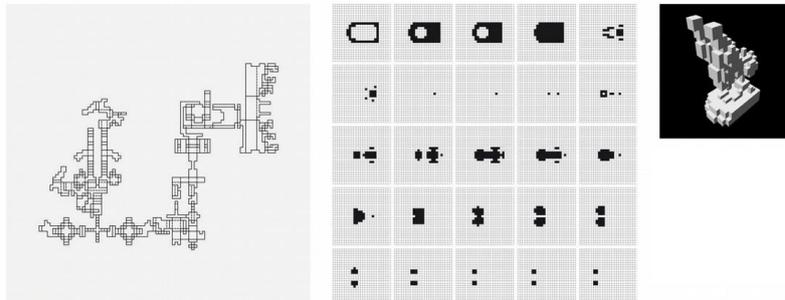


Fig. 1. A. Mourier stereo microscope shape 00797.001: the unfolded net, the slices in the z axis and the 3D numeric object.

tool, shapes still had to be constructed by hand. We proposed to design algorithms to help create specific classes of voxel objects. For this, several constraints had to be met so that our client, A. Mourier, could use these objects: one of the main constraints was to incorporate a manual design possibility so that the artist may express herself. As a first proposal, we developed an online software to generate digital surfaces of revolution based on a hand-drawn generatrix and hand-drawn curves of revolution. This work will be presented in section 2.1. We then proposed a method for creating tubes where the 3D curve and the section of the tube can be designed freely. This will be presented in section 2.2. More recently, we looked into the problem of unfolding the surface of a voxel object. This was motivated by the nets created, by the artist, by hand. This will be presented in subsection 2.3.

2.1 Digital Surface of Revolution with hand-drawn generatrix and curve of revolution

Working with a visual art artist, our goal was to propose flexible, intuitive to use tools for designing 3D voxel surfaces. For this we considered 2D (hand) drawings which are a natural shape representation form for most artists. A recent paper proposed a digitization method for surfaces of revolution [3] based on a very simple and straightforward method for digitizing almost any implicit n D surfaces [16]. It is based on a morphological type digitization method called *flake digitization*. The flake digitization allows to define digital surfaces with a controlled topology (control on tunnels of the digital surface). The paper [16]

can be seen as an extension to all tunnel connectivities and dimensions of the paper of S. Laine [14].

A surface of revolution is defined by two 2D curves: the curve of revolution and the generatrix. We propose three ways of defining a curve of revolution and three ways of defining a generatrix in order to allow maximum flexibility. A curve of revolution can be defined by an implicit curve (that separates space into positive and negative valued regions), by a closed hand-drawn curve or by a binary pixel image that serves as look-up matrix. The curve of revolution is not limited to the traditional circle and not limited to a unique connected component, which offers a great liberty in designing complex shapes. Contrary to the curve of revolution, the generatrix is not necessarily a closed curve. Three similar ways of defining generatrix curves are proposed: as an explicit function, as hand-drawn curves or as contour curves extracted from a binary image. This leads to nine different ways of defining digital surfaces of revolution. Two of those methods have already been published by Andres et al. respectively in [3] and [4].

Let us detail a bit the overall method. A curve of revolution can be basically defined in any way as long as we are able, slice by slice, to define regions with an interior and an exterior. The curve(s) of revolution are then defined as the boundaries between those regions. For the digitization, we determine if a digital point belongs to the digital surface of revolution by considering the vertices of the three dimensional structuring element (k -Flakes [16]) centered on this point and computing their position relatively to the curve of revolution. If some of the vertices are inside and some outside then the corresponding voxel is cut by the surface and therefore belongs to the Digital Surface of Revolution. When the curve of revolution is implicitly defined the vertex localisation is straightforward. When the curve of revolution is given as a hand-drawn curve, which is more natural for an artist, we record the sequence of Euclidean points, while the artist draws the curve of revolution. We ensure that this curve is a closed one (to define one or several interior(s) and exterior(s)) by adding the first point at the end of the list. This list of point is then treated as a closed polygon and the critical localisation information can be obtained using a *Point In Polygon* (PIP) algorithm [12].

The other curve, the generatrix g , typically plays the role of an homothetic factor for the curve of revolution. A slice $z = z_0$ of the surface of revolution is the curve of revolution scaled by a factor $g(z_0)$ for a generatrix that is defined by an explicit function $y = g(z)$. When the generatrix is drawn and represented by a list of points, there are two main issues: firstly, there can be more than one value $g(z)$ per z and secondly, the curve may not be closed. To handle the multiplicity of $g(z)$ values, the point sequence is divided into strictly monotonic (in z , increasing or decreasing) or horizontal subsequences. The end point of one subsequence is duplicated as the starting point of the next one. Each subsequence can be digitized completely independently because the flake digitization method we use is a morphological type digitization (i.e. consistant with the union operator). One last problem had to be addressed: the generatrix may be an open curve and therefore it may have extremities. This case needs

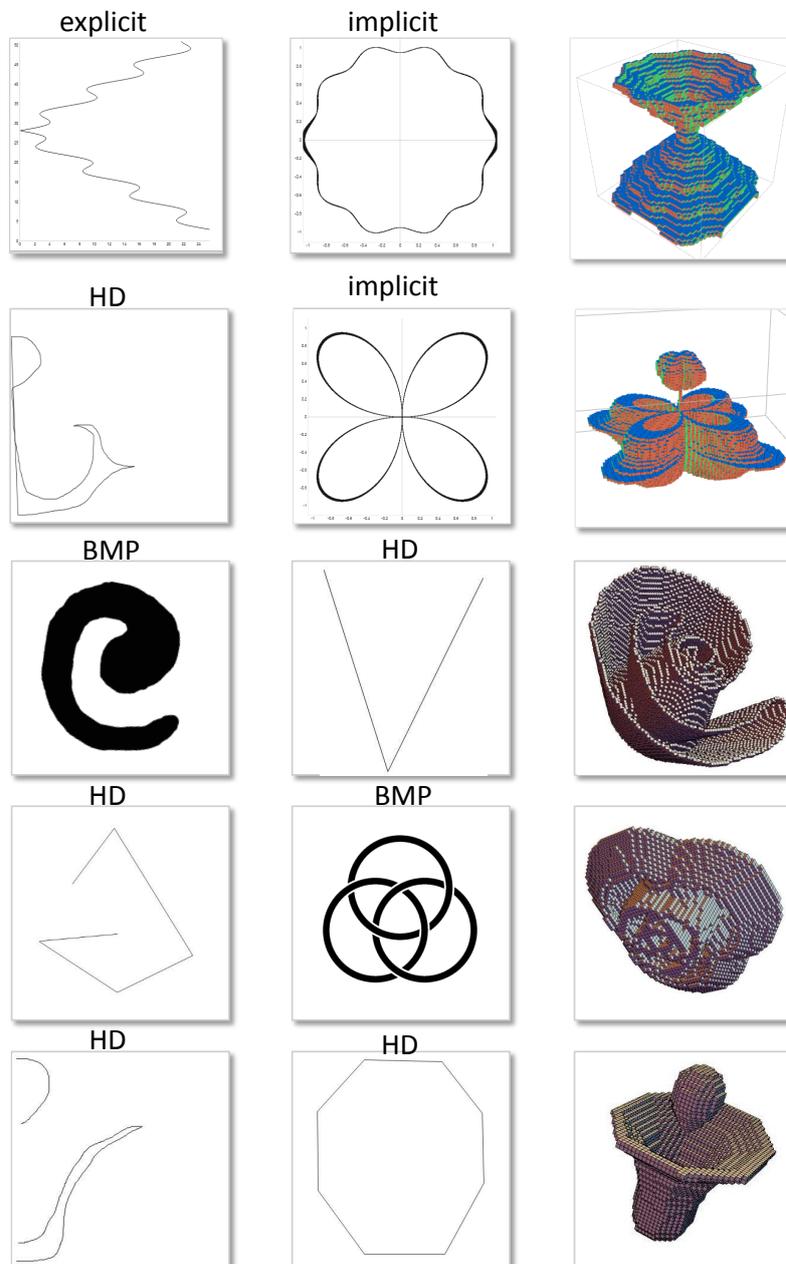


Fig. 2. Examples of 3D surfaces of revolution (HD: hand-drawn, BMP: bit map image).



Fig. 3. Spinning top sculptor

to be handled specifically because the digitization of the surface of revolution supposes that we are able, for all points, to compute its localisation relatively to the surface. At the generatrix extremities, some of the vertices of the voxels' flake are outside the domain of definition of the generatrix. There can be, in this case, a defined localisation for some vertices and not for others. In this case, we consider only the parts of the flake that are inside the domain and therefore we take, as substitute for the undefined vertices, the endpoints of a *cropped flake* line segments. See [4] for more details. See <http://imgur.com/a/eDFbY> for some examples of swept digital tubes and digital surfaces of revolution. The method is not limited to curves of revolution of dimension two. You can see an example at <http://imgur.com/a/eDFbY> of a four-dimensional torus defined by a 3D surface of revolution (a sphere) and a 2D generatrix (a circle).

The software that implements these methods is available at http://xlim-sic.labo.univ-poitiers.fr/demonstrateurs/DSoR_Generator/. This software has been used by A.Mourier to build chess pieces that have been 3D printed (See Figure 4 for an example), and a sculptor representing a spinning top (Figure 3).

2.2 Swept Tubular Surfaces

As a complement to surfaces of revolution, we designed a tool that allows to create *Swept Tubular surfaces*. Such tubes can also be defined as digital implicit surfaces [16]. These digital surfaces have applications well beyond our visual art interest: modeling of body parts [13], medicine [7], etc.

Formally, a swept tube is defined by a 3D curve, the *spine curve*, and a 2D closed curve, the *profile*. The *profile* (sometimes called *cross-section* in the literature) is swept out in a plane normal to the 3D *spine curve* (or *trajectory*). Using a mapping of the profile on the spine together with the implicit surface

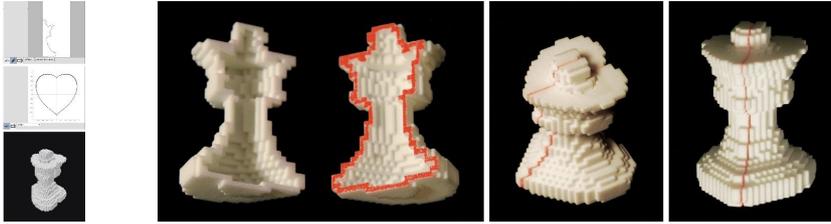


Fig. 4. The 3D printed chess white queen. This piece was built using an implicit curve of revolution and a hand-drawn generatrix.

digitization method [16], it is actually quite easy to build digitized swept tubes. In Figure 5, one can see three interlaced digital swept tubes generated by a parametric 3D spine curve and a bitmap image of three disks that served as profile (please check <http://imgur.com/a/eDFbY> for an animated version). For the mapping, we chose the Frenet-Serret formulas [10] which describes the motion of a particle along a 3D continuous curve (any other mapping intended to define a moving frame like the *Darboux frame* for instance could also be used). The Serret-Frenet frame is defined by the tangent, the normal and the binormal unit vectors in any point of the 3D curve.

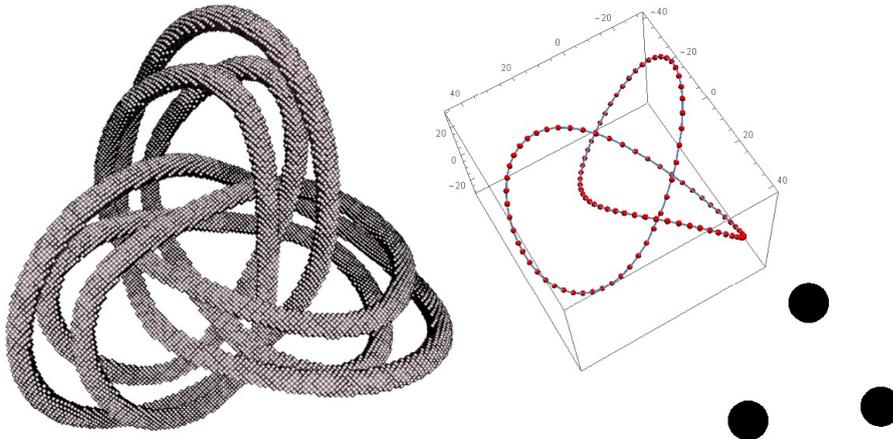


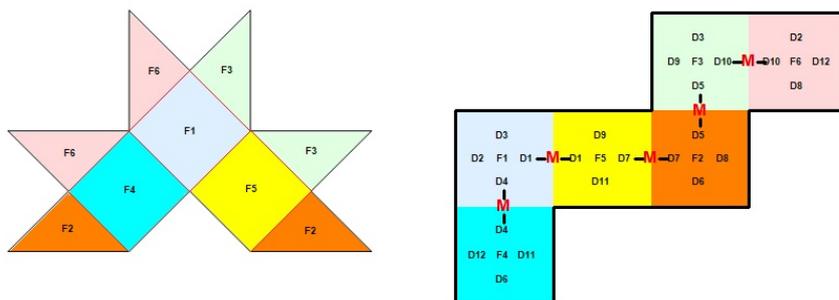
Fig. 5. An example of a swept tube defined by a parametric closed 3D spine curve (on the upper right side) and a bitmap image of three disks as profile (lower right side)

For the animations that we created or that the artist created, the idea was simply to rotate the profile (see <http://imgur.com/a/eDFbY> for some examples).

2.3 Unfolding voxel surfaces

A central question in graphical arts has always been the representation of 3D so-called *reality* in 2D. For a long time Hierarchical representation (where important personae tended to be painted in big and in the middle of a painting while others were painted in small and in the periphery) or perspective representation was the norm. Let us note that hierarchical representation, although less realistic, contained information with an added, otherwise abstract, social and political dimension. *Cubists*, at the beginning of the twentieth century, proposed a new form of representation that tried to incorporate multiple view points in an abstract recomposition. One of the motivations was to represent elements that would otherwise be hidden. The abstract nature of such recomposition make the paintings sometimes difficult to understand while at the same time they potentially represent a more complete representation of the reality than a more classical representation with only one point of view. That was the starting point of A. Mouriers' interest in nets and the problem of unfolding voxel surfaces. A net shows the complete 3D object in 2D with the whole surface visible although the net makes it difficult to imagine what the corresponding 3D object looks like. Representing a complete 3D *reality* comes with a price.

The unfolding problem is an old problem already discussed by A. Durer [11]. Since then, unfolding problems have been extensively studied with a wide range of applications ranging from industrial manufacturing, storage problems, to texture mapping, etc. The unfolding problem can be stated as follows: *can the surface of a 3D closed object be unfolded flat to a single component without overlap?* [5]. There are two main type of unfoldings: edge-following unfolding and general unfoldings. In edge-unfoldings, one can only make cuts along the edges of the polyhedra while for general unfoldings, one can cut through faces.



2.4 state of the art

It has been shown that a non convex polyhedra can not always be edge unfolded [5] and that a convex polyhedra has always a general unfolding [1]. To our best knowledge, it is not known if a convex polyhedra always has an edge unfolding or if a non convex polyhedra always has a general unfolding. There is however a subclass of unfolding problems that does not deal with an arbitrary polyhedra but with what is called orthogonal polyhedra. An Orthogonal polyhedron is a polyhedron whose faces meet only with a 0 angle (both faces are coplanar), a $\pi/2$ angle (both faces have a so-called *valley* fold) or a $3\pi/2$ angle (both faces have a so-called *mountain* fold). The terms valley and mountain fold comes originally from the origami community. In 2007, it has been shown that arbitrary genus 0 orthogonal polyhedra always have a general unfolding [9]. More recently, in 2016, an algorithm for genus 2 orthogonal polyhedra has been proposed [8] as well as a one layer general genus method [15]. A particular case of general unfoldings that are considered are so called grid-unfoldings where cuts across orthogonal faces are only allowed along the edges of a subgrid that may be arbitrarily small in some cases.

2.5 Our unfolding problem

Our work on these problems have been driven by the questions and needs as presented to us by A. Mourier. An *art-and-science* project funded by the regional direction of Art and Culture (DRAC Nouvelle Aquitaine) was proposed with the question of peeling (unfolding) genus 0 egg shaped objects (not necessarily convex). For this, we proposed, firstly, a new way of generating digital objects based on Focus points which is presented at DGCI 2017 [2]. Once we had a way of generating such objects, we were asked if it was possible to generate nets randomly. The problem, for A. Mourier, is the question of having different nets that represent the same final 3D object.

In our case, we consider a 6-connected voxel object whose surface is divided into square voxel faces. We are therefore looking into an edge unfolding problem of orthogonal polyhedra that bears some resemblance with the general grid-unfolding of orthogonal polyhedra class of problems. The key difference here is that the voxel faces can not be subdivided. The classical counterexamples, to prove that edge-unfolding for orthogonal polyhedra is not always possible, have nets in our case. The question of the existence of an edge-unfolding solution for all such voxel surface object remains open. And if such a solution always exists, what algorithm could be proposed in the general case ?

For the problem on hand, with the demand of the artist to not propose a deterministic solution, we developed an algorithm based on the following basis:

1. we start with an empty 2D grid in which the voxel faces will be set in order to define a net.
2. First we extract a list of the surface faces of the 6-connected voxel object.
3. for each face we numerotate the edges. An edge is of course shared by two and only two faces.

4. We start with a random face and put it in the center of our grid.
5. We compute all the faces that can be put down next as neighbors for the already settled faces. The key point is that you can put down a new face next to a face in the grid if the edge corresponds. Of course it may happen that a grid square next to a settled face can not be filled because the face that corresponds to this edge has already been put down elsewhere in the grid for another edge.
6. we choose randomly, among those faces, the next face that can be put down as long as not all the faces have been put down or that we are not blocked.
7. if we are not blocked or haven't finished we go to step 5.

This method of building a net does not always provide a solution. It can be blocked simply because all the (four) grid places where we could put down a face are already occupied by other faces.

There are however several ways to improve the convergence: If we are blocked and have several faces left, we can try to correct our net. For the faces that we could not put down, we can determine where we could have put them. If the location where we could have put down the face is an isolated face or a face in a cycle such that it is not the place where the cycle is linked to the other faces of the net, then we can remove the isolated or cycle face and put down our blocked face. We have now swapped one face for another. The idea is that, may be, this new face will find a free grid spot to put it down. The reason why we can only swap with isolated or non splitting cycle faces is that otherwise the face represents the root of a tree of faces that would be disconnected from the net. Since the aim is to create a net in one piece, that can not be allowed. In some cases, a face can not be placed at any spot where we would have isolated faces. We have then the choice, either to abandon and start over or to extract all the faces of the smallest sub-tree and start over with trying to place all those new faces.

A last method we have implemented to get out of a blocking situation, without starting all over, is to remove all the isolated and non splitting cycle faces from the blocked net and start over at step 6.

2.6 results

To our surprise, the algorithm we have developed works surprisingly well with what we thought would be topologically complicated voxel objects. Figure 7 shows the net for a voxel object formed of a 5^3 voxel cube with 4 traversing tunnels on each side. There are 270 faces and the algorithm takes only a couple of seconds to find a net for such an object with very often only a couple of attempts. The algorithm has also found a solution for the much more complex digital object shown below in the figure. We didn't include here the image of the net because it is simply too big to distinguish any details. It can take usually up to a couple of hours to find a solution for such an object. There are 1350 faces in this case. The other surprise was that the method works not very well with topologically simple objects. The algorithm struggles with simple objects such as

digital spheres of relatively small radii. It is however not completely surprising, all things considered. Our net generator is based on random choices which tends to produce relatively compact nets. The algorithm will typically not stretch the net in one direction to provide more space to put down faces.

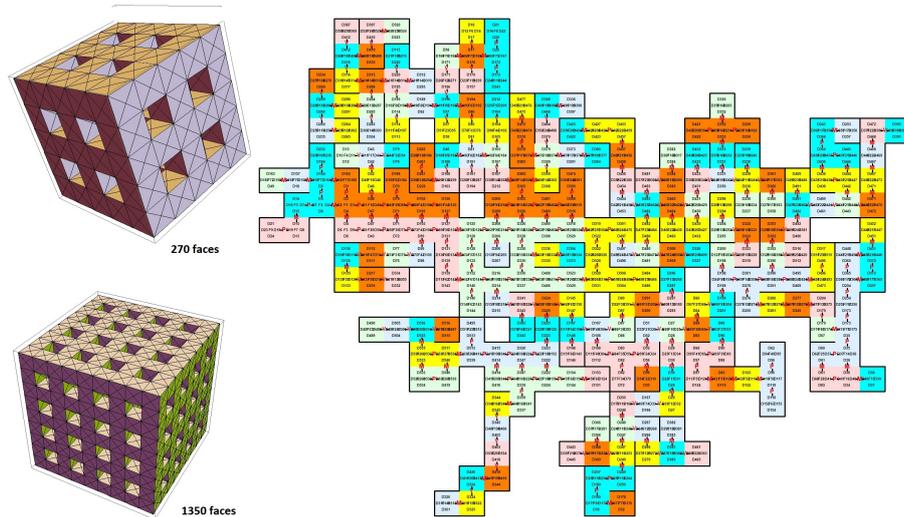


Fig. 7. Voxel cube with 4 holes by face and one of the resulting nets. Below Cube with 16 holes where a solution was found as well.

3 Conclusion and Perspectives

In this paper we have presented some of the scientific work that has resulted from the collaboration between a scientific team working in digital geometry and a visual art artist. As an artist, A. Mourier explores shapes in a voxel form. Several modeling tools for digital objects have been developed with a focus on giving the artist the possibility to express herself graphically. This has led to develop new methods for generating digital surfaces of revolution and digital tubular swept objects. A more recent collaboration has focused on the unfolding problem for voxel surface objects. A new algorithm for generating nets has been proposed. There are several open questions that have been raised in these different works: we have developed a method where the generatrix may be an open curve. what about open curves of revolution? The same question can be asked for digital swept tubes. Can they be defined with open profile curves? Lastly, we have considered a particular type of unfolding problem: orthogonal edge based grid unfoldings. Is there always a solution for this particular case? Right now we

have developed a random search algorithm to generate nets that are different each time. This method may need several attempts before it proposes a result. Is there a way to create deterministic algorithms for the general case of convex, non-convex voxel surface unfoldings?

References

1. Pankaj Agarwal, Boris Aronov, Joseph O'Rourke, and Catherine Schevon. Star unfolding of a polytope with applications. *SWAT 90*, pages 251–263, 1990.
2. E. Andres, R. Biswas, and P. Bhowmick. *Digital primitives defined by weighted focal set*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.
3. E. Andres and G. Largeteau-Skapin. Digital surfaces of revolution made simple. In *DGCI 2016, Nantes, France*, volume 9647 of *LNCS*, pages 244–255. Springer International Publishing, 2016.
4. Eric Andres, Lydie Richaume, and Gaelle Largeteau-Skapin. Digital surface of revolution with hand-drawn generatrix. *Journal of Mathematical Imaging and Vision*, Feb 2017.
5. Therese Biedl, Erik Demaine, Martin Demaine, Anna Lubiw, Mark Overmars, Joseph O'Rourke, Steve Robbins, and Sue Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proceedings of the 10th Canada conference on computing and geometry*, pages 70–71, 1998.
6. J.L. Borges. *La Biblioteca de Babel: prólogos*. Obras de Borges. Emecé Editores, 2000.
7. Alexander Bornik, Bernhard Reitering, and Reinhard Beichel. *Simplex-Mesh Based Surface Reconstruction and Representation of Tubular Structures*, pages 143–147. Springer Berlin Heidelberg, 2005.
8. Mirela Damian, Erik D. Demaine, Robin Y. Flatland, and Joseph O'Rourke. Unfolding genus-2 orthogonal polyhedra with linear refinement. *CoRR*, abs/1611.00106, 2016.
9. Mirela Damian, Robin Flatland, and Joseph O'Rourke. Epsilon-unfolding orthogonal polyhedra. *Graphs & Combinatorics*, 23, 2007.
10. Uday Chand De. *Differential Geometry of Curves and Surfaces in E3*. Anshan, 2007.
11. A. Dürer. *Unterweysung der Messung mit dem Zirkel un Richtscheyt in Linien Ebnen uhnd Gantzen Corporen*. 1525.
12. Michael Galetzka and Patrick O Glauner. A correct even-odd algorithm for the point-in-polygon (pip) problem for complex polygons. *arXiv preprint arXiv:1207.3502*, 2012.
13. Dae-Eun Hyun, Seung-Hyun Yoon, Myung-Soo Kim, and Bert Juttler. Modeling and deformation of arms and legs based on ellipsoidal sweeping. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, pages 204–212. IEEE, 2003.
14. S. Laine. A topological approach to voxelization. *Computer Graphics Forum*, 32(4):77–86, 2013.
15. Meng-Huan Liou, Sheung-Hung Poon, and Yu-Jie Wei. *On Edge-Unfolding One-Layer Lattice Polyhedra with Cubic Holes*, pages 251–262. 2014.
16. J.L. Toutant, E. Andres, G. Largeteau-Skapin, and R. Zrour. Implicit digital surfaces in arbitrary dimensions. In *18th DGCI, Siena (Italy)*, volume 8668 of *LNCS*, pages 332–343. Springer, 2014.